On the Performance of SVD-based Algorithms for Collaborative Filtering

Manolis Vozalis, Angelos Markos and Konstantinos Margaritis
Department of Applied Informatics
University of Macedonia
Thessaloniki, Greece
Email: mans@uom.gr, amarkos@uom.gr, kmarg@uom.gr

Abstract—In this paper, we describe and compare three Collaborative Filtering (CF) algorithms aiming at the low-rank approximation of the user-item ratings matrix. The algorithm implementations are based on three standard techniques for fitting a factor model to the data: Standard Singular Value Decomposition (sSVD), Principal Component Analysis (PCA) and Correspondence Analysis (CA). CA and PCA can be described as SVDs of appropriately transformed matrices, which is a key concept in this study. For each algorithm we implement two similar CF versions. The first one involves a direct rating prediction scheme based on the reduced user-item ratings matrix, while the second incorporates an additional neighborhood formation step. Next, we examine the impact of the aforementioned approaches on the quality of the generated predictions through a series of experiments. The experimental results showed that the approaches including the neighborhood formation step in most cases appear to be less accurate than the direct ones. Finally, CA-CF outperformed the SVD-CF and PCA-CF in terms of accuracy for small numbers of retained dimensions, but SVD-CF displayed the overall highest accuracy.

Keywords—Collaborative Filtering; Singular Value Decomposition; Correspondence Analysis; Principal Component Analysis

I. INTRODUCTION

Collaborative Filtering (CF) is a popular approach employed by Recommender Systems (RSs), a term used to describe intelligent techniques that generate personalized recommendations. The premise of CF is that users who have agreed in the past tend to agree in the future. A common approach to collaborative prediction is to fit a factor model to the original rating data, and use it in order to make further predictions [1], [2]. The goal of a factor model is to uncover latent features that explain user preferences. That is achieved by approximating the observed user preferences in a low dimensionality space. In this paper, we will focus on CF methods based on Singular Value Decomposition (SVD), a powerful technique for approximating matrices of a given rank.

Several studies described and evaluated the performance of various matrix factorization techniques in the context of Collaborative Filtering, including both plain and similarity/neighborhood based approaches [3], [4], [5], [6]. In [5], an SVD approach is presented that incorporates demographic information and ratings to enhance the plain CF algorithm. Goldberg et al. utilized Principal Component Analysis (PCA) as a preprocessing step in a recommendation procedure, followed by a recursive rectangular clustering method [3]. Alternatively, PCA can be iteratively performed until convergence and users are clustered based on their scores in the reduced space [7]. In [4] a complex scheme is described for combining predictions generated by SVD extensions applied to the Netflix data set. In addition to a straightforward optimization approach for the general low-rank approximation problem, many authors proposed sophisticated algorithms to cope with the missing data problem [7], [4], [6].

In this paper, we present and evaluate three CF algorithms that can be used for the low-rank approximation of the user-item ratings matrix in terms of least-squares. The algorithms are adaptations of three baseline techniques for fitting a factor model to the data: Standard Singular Value Decomposition (sSVD), Principal Component Analysis (PCA) and Correspondence Analysis (CA). For each algorithm we implement two similar versions. The first one involves a direct rating prediction scheme based on the reduced user-item ratings matrix, while the second incorporates an additional neighborhood formation step.

The three CF versions described in the following sections share the SVD as an algorithmic engine for dimension reduction and prediction generation. PCA is the basis for linear least-squares approximation of a data matrix, while CA can be regarded as a particular case of non-linear, weighted PCA [8]. CA considerably expands the scope of a PCA-type analysis in its ability to handle a wide range of data. The main contribution of this paper lies in the application of CA in the context of CF. Also, to our knowledge, this is the first comparison of baseline low-rank approximation algorithms in terms of CF.

The paper is organized as follows: Section II is devoted to the brief presentation of the three SVD-based matrix factorization techniques. The two CF versions of each method are thoroughly described in Sections III and IV. The efficiency of each approach is demonstrated in Section V through a set of experiments on a publicly available data set. The paper concludes in Section VI.
II. A FAMILY OF SVD-BASED METHODS

In this section we present three SVD-based matrix factorization techniques which were selected for the implementation of the proposed approaches: Singular Value Decomposition, Principal Component Analysis and Correspondence Analysis. We will focus more on the latter method, since it is the first time, to our knowledge, that CA is utilized in the context of collaborative filtering algorithms.

A. Standard Singular Value Decomposition

Standard Singular Value Decomposition (sSVD) is a well known matrix factorization method which takes an $m \times n$ matrix $A$, with rank $r$, and decomposes it as follows [9]:

$$ A = USV^T $$

$U$ and $V$ are orthogonal matrices with dimensions $m \times m$ and $n \times n$ respectively. $S$, called the singular matrix, is an $m \times n$ diagonal matrix whose diagonal entries are non-negative real numbers.

The initial $r$ diagonal entries of $S$ $(s_1, s_2, \ldots, s_r)$ have the property that $s_1 > 0$ and $s_1 \geq s_2 \geq \ldots \geq s_r$. Accordingly, the first $r$ columns of $U$ are eigenvectors of $AA^T$ and represent the left singular vectors of $A$, spanning the column space. The first $r$ columns of $V$ are eigenvectors of $A^TA$ and represent the right singular vectors of $A$, spanning the row space. If we focus only on these $r$ nonzero singular values, the effective dimensions of the SVD matrices $U$, $S$ and $V$ will become $m \times r$, $r \times r$ and $r \times n$ respectively.

The sSVD provides a robust computational method for the low-rank approximation of the original matrix, $A$ in terms of the 2-norm and Frobenius norm [9] and can be a particularly useful technique in the context of Recommender Systems [10]. By retaining the $k \ll r$ largest singular values of $S$ and discarding the rest, we reduce the data dimensionality and expect to capture the underlying latent structure of the original data.

B. Principal Component Analysis

Principal Component Analysis, or PCA, is multivariate data analysis technique aiming at the low-rank least-squares approximation of a data matrix and has been extensively used in survey analysis, medical imaging, lossy data compression and feature extraction [11], [3], [7]. PCA reduces data dimensionality by optimally projecting highly correlated data along a smaller number of orthogonal dimensions. Its objective is to find a (linear) transformation of the original variables to a set of new uncorrelated variables (the principal components) such that a very high proportion of the variation of the old variables is captured by relatively few of the new ones. A comprehensive overview of the theory and applications of PCA can be found in [11].

In practice, classical PCA involves the calculation of the eigenvalue decomposition of the data covariance matrix. The eigenvalues of the covariance matrix indicate the amount of variance along the direction given by the corresponding eigenvector. The SVD offers an alternative viewpoint to some aspects of the PCA theory. The PCA solution can be found by first computing the SVD of the original mean centered data matrix $A$, i.e., $A = USV^T$ with $U^TU = V^TV = I$, where $U$ contains the variables’ loadings for the principal components and $S$ has the corresponding variances along the diagonal [11]. A reduction to $k$ dimensions is obtained by projecting the original data matrix on the subspace consisting of eigenvectors corresponding to the largest $k$ eigenvalues of the covariance matrix.

C. Correspondence Analysis

Correspondence Analysis (CA) is mainly considered as a non-linear multidimensional data analytic method, suitable for exploring the associations between two or more, non-metric, categorical variables without a priori hypotheses or assumptions. CA was originally used for the analysis and visualization of data arriving typically from fields of social sciences and biometrics [8]. However, the method has recently attracted the attention of the engineering community from a number of disciplines, including machine learning, image retrieval and data mining [12], [13], [14]. Additionally, the method has been described as a preprocessing step for pattern recognition [15]. In practice, CA can be performed to analyze almost any type of tabular data after suitable transformation or recoding [8]. The only input requirement of the method is a matrix with non-negative entries, with at least one non-zero entry to each row and each column.

CA can be described as a particular case of weighted PCA [16], [13]. Similar to PCA, the rows or columns of a data matrix are assumed to be points in a high-dimensional space in which distance is measured by a weighted Euclidean metric and the points themselves have differential weights, called “masses”. The method aims to highlight both visible and hidden relations in the data structure by mapping the original data onto lower-dimensional maps, so that the principal dimensions (usually two or three) capture the most variance possible. These dimensions can be considered as latent constructs or new composite quantitative variables, with metric properties, that summarize the original multidimensional information. In that sense, CA can be viewed as a method that quantifies qualitative data, with a simultaneous dimensionality reduction [16], [8].

In the CA context, the SVD provides a straightforward mechanism of approximating the so-called standardized residuals matrix with another matrix of lower rank by weighted least squares [8], [17]. All numerical results of the method are obtained directly from the SVD.

III. DIRECT RATING PREDICTION

In this section we will describe the CF versions of the aforementioned matrix factorization techniques, namely
SVD-CF, PCA-CF and CA-CF, in order to make direct prediction generation both scalable and effective.

A. SVD-CF

Following Sarwar et al. [18], we describe the sSVD algorithm in a CF context in order to capture latent relationships between users and items that allow us to compute the predicted likeliness of a certain item by a user.

Step 1. Data Representation

1) Define the original user-item matrix, \( R \), of size \( m \times n \), which includes the ratings of \( m \) users on \( n \) items. \( r_{ij} \) refers to the rating of user \( u_i \) on item \( i_j \).

2) Preprocess user-item matrix \( R \) in order to impute the missing data. The preprocessing is described as follows:

   a) Compute the average of each row, \( \bar{r}_i \), where \( i = 1, 2, ..., m \), and the average of each column, \( \bar{r}_j \), where \( j = 1, 2, ..., n \), from the user-item matrix, \( R \).

   b) Replace all missing values with the corresponding column average, \( \bar{r}_j \), which leads to a new filled-in matrix, \( R_f \).

   c) Subtract the corresponding row average, \( \bar{r}_i \), from \( R_f \), and obtain the row centered matrix \( A \) i.e. the row mean of \( A \) is 0.

Step 2. Low-rank approximation

Compute the SVD of \( A \) and keep only the first \( k \) eigenvalues. This is equivalent to the eigenvalue decomposition of the covariance matrix \( \frac{1}{m-1} \langle R_f \rangle \langle R_f \rangle^T \). The reduced or reconstructed matrix is denoted as \( A_k \).

Step 3. Prediction generation

The predicted rating for user \( u_i \) on item \( i_j \) is given by:

\[
pr_{ij} = \bar{r}_j + U_k \sqrt{S_k^T(i)} \sqrt{S_k V_k^T(j)},
\]

The second part of the equation gives the corresponding element of the reduced matrix \( A_k \). The prediction is generated by adding the mean of the appropriate column, \( \bar{r}_j \), to this element.

B. PCA-CF

PCA is implemented in a CF framework, in a way similar to the one described in the previous section.

Step 1. Data representation

1) Impute the missing values in the original user-item matrix, \( R \), with the corresponding row average, \( \bar{r}_i \), which leads to a new filled-in matrix, \( R_f \).

2) Subtract the corresponding column average, \( \bar{c}_j \), from \( R_f \), and obtain the column centered matrix \( A \) i.e. the column mean of \( A \) is 0.

Step 2. Low-rank approximation

Compute the SVD of \( A \) and keep only the first \( k \) eigenvalues. This is equivalent to the eigenvalue decomposition of the covariance matrix \( \frac{1}{m-1} \langle R_f \rangle \langle R_f \rangle^T \). The reduced or reconstructed matrix is denoted as \( A_k \).

Step 3. Prediction generation

The predicted rating for user \( u_i \) on item \( i_j \) is given by:

\[
pr_{ij} = \bar{c}_j + U_k \sqrt{S_k^T(i)} \sqrt{S_k V_k^T(j)},
\]

The second part of the equation gives the corresponding element of the reduced matrix \( A_k \). The prediction is generated by adding the mean of the appropriate column, \( \bar{c}_j \), to this element.

C. CA-CF

The theory of CA is based on the generalized SVD of the ratings matrix in a weighted least squares sense.

Step 1. Data representation

Impute the missing values in the original user-item matrix, \( R \), with the corresponding column average, \( \bar{c}_j \), which leads to a new filled-in matrix, \( R_f \).

Step 2. Low-rank approximation

Compute the SVD of \( A = D_q^{-1/2} \langle R_f - qw^T \rangle D_w^{-1/2} \), which is known as the standardized residuals matrix of \( R_f \) [8], where \( q \) and \( w \) are the vectors with the row and column marginal relative frequencies and \( D_q = diag(q) \), \( D_w = diag(w) \). The reduced or reconstructed matrix, keeping the first \( k \) dimensions, is denoted as \( A_k \).

Step 3. Prediction generation

The predicted rating for user \( u_i \) on item \( i_j \) is given by:

\[
pr_{ij} = q_i w_j + D_q^{1/2} U_k \sqrt{S_k^T(i)} D_w^{1/2} \sqrt{S_k V_k^T(j)},
\]

The equation gives the corresponding element of the reduced matrix \( A_k \).
step involves the calculation of a similarity measure between each user and his closest neighbors in order to form the user neighborhood. To find the proximity between two users, \( u_a \) and \( u_i \), we utilize the Pearson correlation coefficient [19], which is calculated as follows:

\[
cor_{ai} = \frac{\sum_{j=1}^{h} r_{aj}r_{ij}}{\sqrt{\sum_{j=1}^{h} r_{aj}^2 \sum_{j=1}^{h} r_{ij}^2}}
\]

Note that the summations over \( j \) are calculated over the \( l \) items for which both users \( u_a \) and \( u_i \) have expressed their opinions.

Prediction generation requires that a user neighborhood of size \( h \) is already formed for the active user, \( u_a \). Then, we compute the prediction rating \( p_{aj} \) for user \( u_a \) on item \( i_j \) for each one of the three SVD-based methods, using the following equations:

A. SVD-CF

\[
p_{aj} = \bar{r}_a + \sum_{i=1}^{h} r_{pij} \times \frac{cor_{ai}}{\sum_{i=1}^{h} |cor_{ai}|}
\]

B. PCA-CF

\[
p_{aj} = \bar{r}_j + \sum_{i=1}^{h} r_{pij} \times \frac{cor_{ai}}{\sum_{i=1}^{h} |cor_{ai}|}
\]

C. CA-CF

\[
p_{aj} = \sum_{i=1}^{h} r_{pij} \times cor_{ai} \sum_{i=1}^{h} |cor_{ai}|
\]

It is important to note that the user ratings, \( r_{pij} \), are taken from the reduced matrix \( A_k \). Finally, in the cases of SVD-CF and PCA-CF we have to add the original user and item average back, \( \bar{r}_a \) and \( \bar{r}_j \), respectively, since they were subtracted during the normalization steps of the preprocessing.

V. EXPERIMENTS

In this section we first provide a brief description of the various experiments we executed in order to evaluate and compare the proposed CF methods, and then present and comment on their results.

For this purpose, we utilized MovieLens, the data set publicly available from the GroupLens research group. The MovieLens data set [20], [21] consists of 100,000 ratings which were assigned by 943 users on 1682 movies. Ratings follow the 1(bad)-5(excellent) numerical scale. The sparsity of the data set is high, at a value of 93.7%. Starting from the initial data set, a distinct split of training (80%) and test (20%) data was utilized.

Mean Absolute Error (MAE) was the metric we employed to evaluate the accuracy of the methods [19]. MAE measures the deviation of predictions generated by the RSs from the true rating values, as they were specified by the user.

A. Direct Rating Prediction Results

The number of reduced dimensions, \( k \), was the sole parameter altered during the execution of our experiments on the three direct rating prediction methods. Figure 1 depicts the MAE values obtained for different \( k \) values, ranging between 1 and 25.

Based on Figure 1, the CA-based approach appears to be the most accurate among the three, for smaller values of \( k \). The results suggest that CA-CF is the best in terms of MAE for \( k = 2 \) to 5. However, SVD-CF displays the highest accuracy for larger values of \( k \), with the best overall MAE (0.7895) achieved at a \( k \) of 12. A series of paired t-tests confirmed the statistical significance of differences between SVD-CF and PCA-CF (\( t(8) = 3.60, p < 0.01 \)), as well as the SVD-CF and CA-CF (\( t(8) = 3.88, p < 0.01 \)).

B. Rating Prediction with Neighborhood Formation Results

The three rating prediction methods including neighborhood formation involved two parameters that had to be tuned: (a) neighborhood size and (b) number of retained dimensions. An initial experiment was executed in order to locate the optimal user neighborhood size for each method. Based on the results, neighborhoods of 2, 5 and 4 users were selected for SVD-CF, PCA-CF and CA-CF, respectively and were utilized for the subsequent experiments.

Once the neighborhood size was set, a second experiment was executed, aiming to evaluate the impact of the number of retained dimensions on the prediction accuracy. Figure 2 depicts the MAE results for values of \( k \) between 1 and 25. Based on Figure 2, SVD-CF outperforms the other methods for values of \( k \) lower than 15. For larger values of \( k \) the PCA-CF displays the lowest error values. Nevertheless, the differences in MAE between the two methods are not statistically significant. The best overall MAE (0.7908) is achieved by the SVD-CF approach for \( k=8 \). Follow-up t-tests indicate that the SVD-CF results in significantly lower MAE than the PCA-CF (\( t(160) = 1.99, p < 0.05 \)) and CA (\( t(160) = 2.65, p < 0.01 \)) approaches. Additionally, CA performs worse than the other methods, although it displays the most stable behavior. Comparatively, the SVD-CF approach shows the biggest variation in observed error values.

VI. CONCLUSIONS

This paper presented three CF methods based on standard techniques for fitting a factor model to the data in a least-squares sense. At the core of the three algorithms lies the SVD of an appropriately transformed user-item ratings matrix. For each method we implemented and evaluated a standard and a neighborhood formation scheme on a rating data set.

Results indicated that the direct prediction scheme outperforms the neighborhood formation approach, in general. Among the direct approaches, the one incorporating CA was
found to be more accurate for small numbers of retained dimensions, while the SVD-CF performed better for the remaining values of \( k \). This is an interesting remark considering the fact that CA has not been previously utilized in the CF framework. Many aspects of the CA method, such as its ability of jointly analyzing users and items utilizing a weighting scheme, still need to be further explored. Further research issues also include the application of the described algorithms on data sets with different characteristics, especially sparsity and size. Finally, a comparison of the presented SVD-based methods with some recently proposed SVD variations would also be of interest.

REFERENCES


Figure 2. Comparison of three SVD-based Rating Prediction approaches with Neighborhood Formation


