Available online at www.sciencedirect.com
ScienceDirect
Information Sciences 177 (2007) 3017–3037
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Using SVD and demographic data for the enhancement of generalized Collaborative Filtering

M.G. Vozalis *, K.G. Margaritis

Parallel Distributed Processing Laboratory, Department of Applied Informatics, University of Macedonia, Egnatia 156, P.O. Box 1591, 54006 Thessaloniki, Greece

Received 10 January 2005; received in revised form 26 February 2007; accepted 28 February 2007

Abstract

In this paper we examine how Singular Value Decomposition (SVD) along with demographic information can enhance plain Collaborative Filtering (CF) algorithms. After a brief introduction to SVD, where some of its previous applications in Recommender Systems are revisited, we proceed with a full description of our proposed method which utilizes SVD and demographic data at various points of the filtering procedure in order to improve the quality of the generated predictions. We test the efficiency of the resulting approach on two commonly used CF approaches (User-based and Item-based CF). The experimental part of this work involves a number of variations of the proposed approach. The results show that the combined utilization of SVD with demographic data is promising, since it does not only tackle some of the recorded problems of Recommender Systems, but also assists in increasing the accuracy of systems employing it.

Keywords: Recommender Systems; Collaborative Filtering; Personalization; Singular Value Decomposition (SVD); Demographic information

1. Introduction

Recommender Systems were introduced as a computer-based intelligent technique to deal with the problem of information and product overload. Their purpose is to provide efficient personalized solutions in e-business domains, that would benefit both the customer and the retailer.

Two basic entities are featured in all Recommender Systems: the *user* and the *item*. A user, who utilizes the Recommender System, provides his opinion about past items. The goal of the Recommender System is to generate suggestions about new items for that particular user. This process is based on the input provided, which is usually expressed in the form of ratings from that user, and the filtering algorithm, which is applied on that input.

* Corresponding author. Tel.: +30 2310 891841; fax: +30 2310 891842.
E-mail addresses: mans@uom.gr (M.G. Vozalis), kmarg@uom.gr (K.G. Margaritis).
URLs: eos.uom.gr/~mans (M.G. Vozalis), eos.uom.gr/~kmarg (K.G. Margaritis).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2007.02.036

A number of fundamental problems may reduce the quality of the predictions generated by Recommender Systems. Among others we have to mention *sparsity*, which makes the location of successful neighbors more difficult, *scalability*, which refers to a performance degradation following a possible increase in the amount of data involved, and *synonymy*, which is caused by the fact that similar products may have different names and cannot be easily linked.

Many solutions have been suggested, intending to solve those problems [11,5,13]. We will focus on the case of Singular Value Decomposition which comes from the area of Linear Algebra and was successfully employed in the domain of Information Retrieval. Only recently, it was proposed by various Recommender Systems researchers as a method possibly capable of alleviating the aforementioned problems. The algorithm we present in this paper can be viewed as a generalized Collaborative Filtering approach which utilizes SVD, as an augmenting technique, and demographic data, as a source of additional information, in order to enhance the Recommender System's efficiency and improve the accuracy of the generated predictions.

The subsequent sections are structured as follows: Section 2 introduces the concept of Singular Value Decomposition. Section 3 presents some Recommender Systems which have employed SVD in order to improve their performance. Section 4 provides a brief analysis of our experimental methodology. Section 5 gives a step-by-step description of the proposed filtering method and then applies it on User- and Item-based Collaborative Filtering. The experimental part, included in the same section, tests a number of variations of the main algorithm, contrasting them with each other. Finally, Section 6 summarizes the essence of this work, providing directions for future research.

2. Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) [1,26] is a matrix factorization technique which takes an $m \times n$ matrix A, with rank r, and decomposes it as follows:

$$\mathrm{SVD}(A) = U \times S \times V^{\mathrm{T}}$$

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 $s_i > 0$ and $s_1 \geqslant s_2 \geqslant \cdots \geqslant s_r$. 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 $A^{\mathrm{T}}A$ 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.

SVD can provide the best low-rank approximation of the original matrix, A, which is an attribute particularly useful in the case of Recommender Systems. By retaining the first $k \ll r$ singular values of S and discarding the rest, which, based on the fact that the entries in S are sorted, can be translated as keeping the k *largest* singular values, we reduce the dimensionality of the data representation and hope to capture the important "latent" relations existing but not evident in the original representation of matrix A. The resulting diagonal matrix is termed S^k. Matrices U and V should be also reduced accordingly. U_k is produced by removing $r - k$ columns from matrix U. V_k is produced by removing $r - k$ rows from matrix V. Matrix A_k is defined as

$$A_k = U_k \times S_k \times V_k^{\mathrm{T}}$$

A_k represents the closest linear approximation of the original matrix A with reduced rank k. Once this transformation is completed, users and items can be thought off as points in the k-dimensional space.

3. Using SVD in Recommender Systems

SVD, as part of *Latent Semantic Indexing* (*LSI*), was used widely in the area of **Information Retrieval** [6] in order to solve the problems of synonymy and polysemy. Furthermore, techniques like *SVD-updating* or *folding-in* were proposed to alleviate the problem of *updating*, which refers to the process of adding new terms and/or documents to existing matrices [2].

Those ideas were adopted by researchers in the area of Recommender Systems. Initially, Billsus and Pazzani [3] utilized SVD in order to formulate Collaborative Filtering as a classification problem. In their work they reduce the dimensions of a data matrix via SVD before they feed it into an Artificial Neural Network (ANN). This ANN, or, as the authors claim, any alternative Machine Learning algorithm, can then be trained in order to generate predictions.

GroupLens have also applied SVD in at least three distinct cases regarding Recommender Systems: (i) in an approach which reduces the dimensionality of the user–item space and forms predictions in the reduced space, never building explicit neighborhoods during that procedure [16], (ii) in an approach that generates a user-neighborhood in the SVD reduced space and then applies normal User-based Collaborative Filtering on it [14], and (iii) in an approach that aims at increasing the scalability by applying folding-in for the incremental computation of the user–item model [17].

Goldberg et al. use Principal Component Analysis [7], a technique very similar to SVD, in order to optimally project highly correlated data along a smaller number of orthogonal dimensions. Their Eigentaste algorithm clusters the projected data, before proceeding with an online computation of the recommendations. Finally, Selamat and Omatu propose a method for classification of news web pages, which applies PCA to select the most relevant web page features before feeding them to an Artificial Neural Network for training and testing [19].

4. Experimental methodology

4.1. MovieLens: A GroupLens data set

For the execution of our subsequent experiments we utilized the data publicly available from the GroupLens movie Recommender System. The MovieLens data set, used by several researchers [18,21,8], consists of 100.000 ratings which were assigned by 943 users on 1682 movies. Users should have stated their opinions for at least 20 movies in order to be included. Ratings follow the 1(bad)–5(excellent) numerical scale. Starting from the initial data set, five distinct splits of training and test data were generated (*u1.base*, *u2.base*, *u3.base*, *u4.base*, *u5.base* and *u1.test*, *u2.test*, *u3.test*, *u4.test*, *u5.test*). For each data split, 80% of the original set was included in the training and 20% of it was included in the test data. The test sets in all cases were disjoint.

The complete data set includes in random order 100.000 vectors of the following form:

```
user id|item id|rating|time stamp
```

Obviously, users are enumerated from 1 to 943, items from 1 to 1682, while `ratings` take values between 1 and 5. The time stamps are unix seconds since 1/1/1970 UTC. An actual sample from the GroupLens data set can be found in [22].

Except for ratings awarded by users on items, the MovieLens data set includes information regarding specifically the users and the items. Such information proved to be crucial for the subsequent algorithmic methods. Regarding users, the included data consists of a sequential list, with 943 vectors of the following form:

```
user id|age|gender|occupation|zip code
```

The `user ids` are the ones also used in the main data file. The `gender` can be either ‘M’, for male, or ‘F’, for female. The `occupation` takes a value from a list of 21 distinct possibilities. An actual sample from the demographic information about the users, which are included in the MovieLens data set, can be found in [22].

Finally, regarding the items, which in the case of the MovieLens data set correspond to movies, there is another sequential list, with 1682 vectors of the following form:

```
movie id|movie title|release date|video release date|IMDb URL|unknown|Action|
Adventure|Animation|Children's|Comedy|Crime|Documentary|Drama|Fantasy|Film-Noir|
Horror|Musical|Mystery|Romance|Sci-Fi|Thriller|War|Western
```

The `movie ids` are the ones used in the main data set. The `movie title` is a string with the title of the movie. The `release dates` are of the form *dd-mmm-yyyy*, e.g. 14-Jan-1967. The `IMDb URL` is a web link leading to the Internet Movie Database page of the corresponding movie. The last 19 fields are the film genres. Items can belong to more than one genres at the same time. An actual sample from the item-related information which are part of the MovieLens data set, can be found in [22].

Both user and item specific information, incorporated in the MovieLens data set, fall under the "demographic data" category. They will be used extensively, as part of the algorithmic approaches which will be proposed in the following sections.

4.2. The evaluation metric

Several techniques have been used to evaluate Recommender Systems. Those techniques have been divided by Herlocker et al. [8] into three categories. The first category includes *Predictive Accuracy Metrics*, such as Mean Absolute Error, mean squared error and normalized mean absolute error. These metrics measure how close the recommender's predictions are to the true user ratings. The second category, *Classification Accuracy Metrics*, includes methods such as Receiver Operating Characteristic curves (ROC curves) and the F1 metric, and measures how often a Recommender System can decide correctly whether an item is beneficial for the user and therefore should be suggested to him. Those metrics require a binary classification of items into useful and not useful. The last category of metrics, called *Rank Accuracy Metrics*, measure the proximity of a predicted ordering of items, as generated by a Recommender System, to the actual user ordering of the same items.

The choice among those metrics should be based on the selected user tasks and the nature of the data sets. P-Tango [5] calculates its efficiency via the deviation of generated predictions from the user-specified ratings. As a result the selected evaluation metric is *inaccuracy*, an alternative name for Mean Absolute Error. On the other hand, Breese et al. [4] want to evaluate a ranked list of recommended items and therefore they calculate the expected utility of that list to the user and use that amount as their metric. We wanted our proposed algorithms to derive a predicted score for already rated items rather than generate a top-N recommendation list. Based on that specific task we proceeded in the selection of the initial evaluation metric for our experiments. That metric was Mean Absolute Error (MAE) [20].

$$\mathrm{MAE} = \frac{\sum_{i=1}^{k} pr_i - r_i}{k} \tag{1}$$

It is a statistical accuracy metric which measures the deviation of predictions, pr_i, generated by the Recommender System, from the true rating values, r_i, as they were specified by the user, for k $\{user, item\}$ pairs. MAE is measured only for those items, for which a user has expressed his opinion.

5. Applying SVD on generalized Collaborative Filtering

5.1. Introduction

The following sections include a detailed introduction to our filtering approach, which tries to enhance conventional Collaborative Filtering techniques, by combining Singular Value Decomposition with the additional information about users or items, derived from demographic data. We start with a general description of our algorithm, presenting its step-by-step execution, which, seemingly, is suitable for application in collaboration with existing CF methods. We proceed by specifically applying it on the two most common CF representatives. *UdemSvd* is the outcome of its combination with User-based CF, and *IdemSvd* is the outcome of its combination with Item-based CF.

Experimental results that will follow, show that while IdemSvd appears to be an effective hybrid filtering solution, UdemSvd does not provide significant added value, when compared to both IdemSvd and the base algorithms. As a result, we choose to provide a rather limited analysis of the UdemSvd algorithm and its experimental results, while focusing more on giving a complete presentation of IdemSvd.

The interesting part of this work is that the proposed algorithm can be thought of as a generalized CF approach. Alternatively, both User-based CF and Item-based CF, naturally selected as the base algorithms for the subsequent experiments which test the efficiency of the resulting UdemSvd and IdemSvd implementations, can be viewed as two specific cases of the proposed algorithm, generated when the parameters involved in the filtering procedure are tuned accordingly.

5.2. Description of our algorithm

We will now present the general steps of how SVD and demographic data can be incorporated in multiple points of Collaborative Filtering in order to enhance it.

- *Step 1a*: Construct **demographic vectors** for the m users and n items that participate in the recommendation process. The information required for those vectors can be usually found in the utilized Collaborative Filtering data sets, like MovieLens and EachMovie. Once constructed, the demographic vectors are collected in array D.
- *Step 1b*: At this point we have to select one of two possible scenarios:
  - *Case 1*: Leave the array of demographic vectors, D, intact, or,
  - *Case 2*: Perform SVD on the array of demographic vectors.
- *Step 2*: Proceed with **Data Representation** which concludes with the construction of the initial user–item matrix, R, of size $m \times n$.
- *Step 3*: Resume with the **Neighborhood Formation**. Once again, we have to select one of the following possible ways:
  - *Case 1*: Neighborhood Formation *without* SVD. We apply the Similarity metric of choice on the ratings of the original user–item matrix, R, and generate the ratings-based correlations for pairs of users or items.
  - *Case 2*: Neighborhood Formation *with* SVD. We perform SVD on the user–item matrix, R, before proceeding with the Neighborhood Formation.
- *Step 4*: Calculate the **Demographic Correlation** between the active user or item, ui_a, and each of the members of its neighborhood, ui_i, by computing their corresponding vector similarities [12]:

$$dem\_cor_{ai} = vect\_sim(\overrightarrow{ui_a}, \overrightarrow{ui_i}) = \frac{\overrightarrow{ui_a} \cdot \overrightarrow{ui_i}}{||\overrightarrow{ui_a}||_2 * ||\overrightarrow{ui_i}||_2} \tag{2}$$

  There exist quite a few possible vector similarity metrics, discussed in the Information Retrieval literature [10], which we could have used instead. We selected the cosine measure, first because it appears to be one of the successful metrics [9], but also for compatibility reasons with past experiments we have executed, where it was also used for the comparison of vectors. Nevertheless, it would be interesting to test and compare the performance of a system utilizing alternative vector similarity metrics, and we intend to do so as part of our future work.Depending on the choices made in Steps 1b and 3, there exist four possible cases, which are summarized in Table 1. They differ in whether the selected pairs of users/items were taken from the *reduced* or the *original* user–item matrix, and in whether *reduced* or *original* demographic vectors were utilized for the calculations of the demographic correlations.
- *Step 5*: Calculate the **Enhanced Correlation**, enh_cor_{ai}, for every pair of the form $\{ui_a, ui_i\}$, where ui_a is the active user/item and ui_i is a member of its neighborhood.

$$enh\_cor_{ai} = \alpha * rat\_cor_{ai} + \beta * dem\_cor_{ai} + \gamma * (rat\_cor_{ai} * dem\_cor_{ai}) \tag{3}$$

Table 1
Different levels of SVD application in Demographic Correlation calculations

Case	SVD on demographic array D	SVD on user–item matrix R
1	Yes	Yes
2	Yes	No
3	No	Yes
4	No	No

rat_cor_{ai} and dem_cor_{ai} represent the ratings-based and the demographic correlation between active user/item ui_a and neighborhood member ui_i, while α, β and γ are flags that define the participation of each of the three components.The enhanced correlation equation was introduced as an attempt to combine ratings-based with demographic correlations between users. The usage of different flag values, as evidenced in the experiments that will follow, allows for testing how varying participation of the aforementioned components can affect the system's result. Being the first time such a blending of correlations was examined, our experiments were limited to few combinations of flag values. In future work, further combinations of the flag values can be tested. Moreover, alternative ways of mixing ratings-based with demographic correlations can be proposed.

- *Step 6*: Proceed with the final step of the recommendation procedure, which is **Prediction Generation**. Our approach differs slightly from the one adopted by classic Collaborative Filtering algorithms. The prediction generation formula utilized instead, replaces the ratings-based correlation, rat_cor_{ai}, between the active user/item, ui_i, and any of the members of its neighborhood, ui_i, by their enhanced correlation, enh_cor_{ai}. It is obvious that the enhanced correlation, as computed in the previous step, possibly incorporates the effects of SVD on the demographic vectors, via the demographic correlations, and/or on the user–item matrix, via the ratings-based correlations.

5.3. Implementations involving User-based Collaborative Filtering

Having outlined the general form of the algorithm, we can proceed with its first specific application. This approach, which we call *UdemSvd*, utilizes User-based Filtering as its base, benefiting from SVD and user demographic information during its execution.

For reasons explained in Section 5.1 we will provide only a brief presentation of UdemSvd. As a result, we decided not to include any additional information about the user demographic data incorporated in the MovieLens data set, the selected encoding of this data, or the generated user demographic vectors. Such information can be found in [22].

The sections that follow start with a brief description of four distinct implementations of UdemSvd, each of them incorporating a different level of SVD and demographic data involvement in the filtering procedure. An experimental section comes next. The efficiency of the implementations is tested and contrasted, not only against each other, but, also, against the selected base algorithm, User-based CF.

5.3.1. Description of the algorithms

Table 2 summarizes the successive execution steps of the four UdemSvd implementations, namely *U-Demog*, *UdemSvd-Dsvd* (U-Dsvd), *UdemSvd-Rsvd* (U-Rsvd) and *UdemSvd-2svd* (U-2svd):

- The **first step** describes the construction of the demographic matrix, and, specifically, states whether this matrix incorporates the application of SVD (*U-Dsvd*, *U-2svd*), or not (*U-Demog*, *U-Rsvd*), depending on the implementation.
- The **second step** is related to the user–item data representation. We distinguish between implementations where the original user–item matrix remains intact (*U-Demog*, *U-Dsvd*), and implementations where SVD is applied (*U-Rsvd*, *U-2svd*). In the latter cases, l defines the size of the dimensionality reduction.
- The **third step** refers to Neighborhood Formation, with each table entry including the corresponding *Similarity metric equation*. This equation differs, depending on the decisions made at the previous step. In implementations where **no** SVD was applied on the user–item matrix (*U-Demog*, *U-Dsvd*), we are utilizing the same similarity metric equation as in plain User-based CF. In implementations where SVD **was** applied on the user–item matrix (*U-Rsvd*, *U-2svd*), we are utilizing a slightly different similarity metric equation, with the *meta-ratings* (mr_{ij}), taken from the reduced user–item matrix, plugged in.
- The **fourth step** includes the calculations of the demographic correlations. We are computing the vector similarities (Eq. (2)) between the corresponding user demographic vectors. Those vectors are taken either from the original demographic matrix (as in *U-Demog* and *U-Rsvd*), or from the reduced demographic matrix (as in *U-Dsvd* and *U-2svd*).

Table 2
Brief description of UDemSvd implementations

	Demographic vectors	Data representation	Neighborhood Formation	Demographic correlation	Enhanced correlation	Prediction generation
U-Demog	Original demographic matrix of size $m \times 27$	Original user–item matrix of size $m \times n$	$sim_{ak} = corr_{ak} = \frac{\sum_{j=1}^{l}(r_{aj}-\bar{r_a})(r_{kj}-\bar{r_k})}{\sqrt{\sum_{j=1}^{l}(r_{aj}-\bar{r_a})^2\sum_{j=1}^{l}(r_{kj}-\bar{r_k})^2}}$	Eq. (2) on original user demographic vectors	Eq. (3)	$udemog\_pr_{aj} = \bar{r_a} + \frac{\sum_{i=1}^{h}(r_{ij}-\bar{r_i})*enh\_cor_{ai}}{\sum_{i=1}^{h}\lvert enh\_cor_{ai}\rvert}$
U-Dsvd	Reduced demog matrix of size $m \times k$, $k < 27$	Original user–item matrix of size $m \times n$	$sim_{ak} = corr_{ak} = \frac{\sum_{j=1}^{l}(r_{aj}-\bar{r_a})(r_{kj}-\bar{r_k})}{\sqrt{\sum_{j=1}^{l}(r_{aj}-\bar{r_a})^2\sum_{j=1}^{l}(r_{kj}-\bar{r_k})^2}}$	Eq. (2) on reduced user demographic vectors	Eq. (3)	$udem\_dsvd\_pr_{aj} = \bar{r_a} + \frac{\sum_{i=1}^{h}(r_{ij}-\bar{r_i})*enh\_cor_{ai}}{\sum_{i=1}^{h}\lvert enh\_cor_{ai}\rvert}$
U-Rsvd	Original demographic matrix of size $m \times 27$	User–item matrix reduced by SVD, $R_{\text{red}} = U_l \cdot S_l \cdot V_l^{\text{T}}$	$sim_{ak} = corr_{ak} = \frac{\sum_{j=1}^{l} mr_{aj}mr_{kj}}{\sqrt{\sum_{j=1}^{l} mr_{aj}^2\sum_{j=1}^{l} mr_{kj}^2}}$	Eq. (2) on original user demographic vectors	Eq. (3)	$udem\_rsvd\_pr_{aj} = \bar{r_a} + \frac{\sum_{i=1}^{h}(rr_{ij}-\bar{r_i})*enh\_cor_{ai}}{\sum_{i=1}^{h}\lvert enh\_cor_{ai}\rvert}$
U-2svd	Reduced demog matrix of size $m \times k$, $k < 27$	User–item matrix reduced by SVD, $R_{\text{red}} = U_l \cdot S_l \cdot V_l^{\text{T}}$	$sim_{ak} = corr_{ak} = \frac{\sum_{j=1}^{l} mr_{aj}mr_{kj}}{\sqrt{\sum_{j=1}^{l} mr_{aj}^2\sum_{j=1}^{l} mr_{kj}^2}}$	Eq. (2) on reduced user demographic vectors	Eq. (3)	$udem\_2svd\_pr_{aj} = \bar{r_a} + \frac{\sum_{i=1}^{h}(rr_{ij}-\bar{r_i})*enh\_cor_{ai}}{\sum_{i=1}^{h}\lvert enh\_cor_{ai}\rvert}$

- The **fifth step** is similar for all implementations. We are simply applying the Enhanced Correlation equation (Eq. (3)) on the ratings-based correlations (taken from step 3) and on the demographic correlations (taken from step 4).
- The **last step** concludes the filtering process with Prediction generation. The difference among the proposed prediction generation formulas is that in implementations where SVD was applied on the user–item matrix (*U-Rsvd*, *U-2svd*), we should be using the reduced ratings, rr_{ij}. In the other two implementations (*U-Demog*, *U-Dsvd*), no SVD was applied and, thus, we are utilizing the user ratings, r_{ij}, from the original user–item matrix.

At this point, a summarized description of the four algorithmic implementations of UdemSvd has been provided. Overall, the desirable enhancement of U-Demog, through the application of SVD, did not work out well in 2 out of the 3 proposed methods. For that reason, an analysis of the experiments executed, results collected and conclusions drawn, is not provided here. Instead, you can refer to [24].

5.4. Implementations involving Item-based Collaborative Filtering

The second specific application of our general algorithmic approach, which we call *IdemSvd*, picks Item-based Filtering as its starting point, while taking advantage of SVD and item demographic information during its execution.

For the construction of item demographic vectors, we are taking into account the MovieLens data set, which, as mentioned in Section 4.1, includes items that correspond to films rated by users. Specifically, the MovieLens data set distinguishes 18 distinct film genres, ranging from Children's to Horror. This leads to the construction of an item demographic vector with 19 features—in case we utilize an additional slot, called "unknown", referring to films that cannot be categorized under any of the existing genres. It is important to point out that a film can belong to more than one genres at the same time. For example, it can be a Comedy, and a Musical. In that case, the slots of the item demographic vector, which correspond to each of these categories, should take a value of 1 (True), with the rest staying fixed at 0 (False). More information about the selected encoding of the item demographic data, or the generated item demographic vectors can be found in [22].

Except for the demographic vectors matrix, the implementations to be discussed in the following sections involve a possible application of SVD on the original user–item matrix, aiming to benefit from the resulting factorization of the user model. The incorporation of SVD in UbCF algorithms, as implemented in Sarwar et al. [14], but also in the methods, which were proposed in Section 5.3, led to a reduced dimensional representation, where the users were represented using pseudo-items, instead of "actual" items. Bearing in mind the results from the user clustering techniques, discussed in Sarwar [16]—according to which, user clustering leads to faster but less accurate predictions—we preferred to apply user factorization, via SVD, instead of user clustering, for this variation of our generalized filtering approach, which chooses IbCF as its basis. Nevertheless, as future work we plan to consider an alternative implementation of the same generalized algorithm with the involvement of user clustering techniques instead.

The sections that follow begin with a brief description of four distinct implementations of IdemSvd. Each implementation incorporates a different level of SVD and demographic data involvement in the filtering procedure. A detailed experimental section comes next. The efficiency of the implementations is tested and contrasted, not only against each other, but, also, against the selected base algorithm, Item-based CF.

5.4.1. Description of the algorithms

Table 3 summarizes the successive execution steps of the four IDemSvd implementations, namely *I-Demog*, *IdemSvd-Dsvd* (I-Dsvd), *IdemSvd-Rsvd* (I-Rsvd) and *IdemSvd-2svd* (I-2svd):

- The **first step** describes the construction of the demographic matrix, and, specifically, states whether this matrix incorporates the application of SVD (*I-Dsvd*, *I-2svd*), or not (*I-Demog*, *I-Rsvd*), depending on the implementation.

Table 3
Brief description of IDemSvd implementations

	Demographic vectors	Data representation	Neighborhood Formation	Demographic correlation	Enhanced correlation	Prediction generation
I-Demog	Original demographic matrix of size $n \times 19$	Original user–item matrix of size $m \times n$	$sim_{jf} = adjcorr_{jf} = \frac{\sum_{i=1}^{l}(r_{ij}-\bar{r}_i)(r_{if}-\bar{r}_i)}{\sqrt{\sum_{i=1}^{l}(r_{ij}-\bar{r}_i)^2\sum_{i=1}^{l}(r_{if}-\bar{r}_i)^2}}$	Eq. (2) on original item demographic vectors	Eq. (3)	$idemog\_pr_{aj} = \frac{\sum_{k=1}^{l} r_{ak} * enh\_cor_{jk}}{\sum_{k=1}^{l}\lvert enh\_cor_{jk}\rvert}$
I-Dsvd	Reduced demog matrix of size $n \times k$, $k < 19$	Original user–item matrix of size $m \times n$	$sim_{jf} = adjcorr_{jf} = \frac{\sum_{i=1}^{l}(r_{ij}-\bar{r}_i)(r_{if}-\bar{r}_i)}{\sqrt{\sum_{i=1}^{l}(r_{ij}-\bar{r}_i)^2\sum_{i=1}^{l}(r_{if}-\bar{r}_i)^2}}$	Eq. (2) on reduced item demographic vectors	Eq. (3)	$idem\_dsvd\_pr_{aj} = \frac{\sum_{k=1}^{h} r_{ak} * enh\_cor_{jk}}{\sum_{k=1}^{h}\lvert enh\_cor_{jk}\rvert}$
I-Rsvd	Original demographic matrix of size $n \times 19$	User–item matrix reduced by SVD, $R_{\text{red}} = U_l \cdot S_l \cdot V_l^{\text{T}}$	$sim_{jf} = adjcorr_{jf} = \frac{\sum_{i=1}^{l} mr_{ij} \cdot mr_{if}}{\sqrt{\sum_{i=1}^{l} mr_{ij}^2 \sum_{i=1}^{l} mr_{if}^2}}$	Eq. (2) on original item demographic vectors	Eq. (3)	$idem\_rsvd\_pr_{aj} = \frac{\sum_{k=1}^{h} enh\_cor_{jk} * (rr_{ak} + \bar{r}_a)}{\sum_{k=1}^{l}\lvert enh\_cor_{jk}\rvert}$
I-2svd	Reduced demog matrix of size $n \times k$, $k < 19$	User–item matrix reduced by SVD, $R_{\text{red}} = U_l \cdot S_l \cdot V_l^{\text{T}}$	$sim_{jf} = adjcorr_{jf} = \frac{\sum_{i=1}^{l} mr_{ij} \cdot mr_{if}}{\sqrt{\sum_{i=1}^{l} mr_{ij}^2 \sum_{i=1}^{l} mr_{if}^2}}$	Eq. (2) on reduced item demographic vectors	Eq. (3)	$idem\_2svd\_pr_{aj} = \frac{\sum_{k=1}^{h} enh\_cor_{jk} * (rr_{ak} + \bar{r}_a)}{\sum_{k=1}^{l}\lvert enh\_cor_{jk}\rvert}$

- The **second step** is related to the user–item data representation. We distinguish between implementations where the original user–item matrix remains intact (*I-Demog*, *I-Dsvd*), and implementations where SVD is applied (*I-Rsvd*, *I-2svd*). In the latter cases, l defines the size of the dimensionality reduction.
- The **third step** refers to Neighborhood Formation, with each table entry including the corresponding *Similarity metric equation*. This equation differs, depending on the decisions made at the previous step. In implementations where **no** SVD was applied on the user–item matrix (*I-Demog*, *I-Dsvd*), we are utilizing the same similarity metric equation as in plain Item-based CF. In implementations where SVD **was** applied on the user–item matrix (*I-Rsvd*, *I-2svd*), we are utilizing a slightly different similarity metric equation, with the *meta-ratings* (mr_{ij}), taken from the reduced user–item matrix, plugged in.
- The **fourth step** includes the calculations of the demographic correlations. We are computing the vector similarities (Eq. (2)) between the corresponding item demographic vectors. Those vectors are taken either from the original demographic matrix (as in *I-Demog* and *I-Rsvd*), or from the reduced demographic matrix (as in *I-Dsvd* and *I-2svd*).
- The **fifth step** is similar for all implementations. We are simply applying the Enhanced Correlation equation (Eq. (3)) on the ratings-based correlations (taken from step 3) and on the demographic correlations (taken from step 4).
- The **last step** concludes the filtering process with Prediction generation. The difference among the proposed prediction generation formulas is that in implementations where SVD was applied on the user–item matrix (*I-Rsvd*, *I-2svd*), we should be using the reduced ratings, rr_{ak}. In the other two implementations (*I-Demog*, *I-Dsvd*), no SVD was applied and, thus, we are utilizing the item ratings, r_{ak}, from the original user–item matrix.

At this point, a summarized description of the four algorithmic implementations of IdemSvd has been provided. We will proceed with a detailed discussion of the experiments executed for each of these implementations. The series of the experiments will conclude with an overall comparison, where the proposed algorithms' efficiency will be evaluated.

5.4.2. Experiments with I-Demog: Demographically enhanced Item-based Filtering without SVD

I-Demog [25,23] can be described as an attempt to enhance the performance of Item-based Filtering, by taking advantage of demographic information regarding items. At the same time, for reasons that will be mentioned shortly, I-Demog can also be considered a generalization of plain Item-based CF.

For the experiments that follow, a number of distinct implementations of I-Demog have been tested. These implementations, distinguished by the values of the enhanced correlation flags, along with the resulting enhanced correlation equations, are gathered in Table 4. As we can see, plain Item-based Filtering can be viewed as a sub-case of I-Demog, resulting from it after the assignment of the appropriate values to the flags: $\alpha = 1$, $\beta = 0$ and $\gamma = 0$.

Fig. 1 compares the Mean Absolute Errors (MAE) collected from our I-Demog implementations (*i-demog1&2&3&4*) and Item-based Collaborative Filtering (*Item-based*), for neighborhoods with varying sizes. Based on this figure, I-Demog3 and I-Demog4 display the best overall accuracy, clearly outperforming not only the rest of the demographically enhanced I-Demog implementations but, most importantly, plain Item-based Filtering. We pick I-Demog4, which sets $\alpha = 1$, $\beta = 1$ and $\gamma = 0$ in its enhanced correlation equation (Eq. (3)), as the single best I-Demog implementation. Therefore, this particular combination of enhanced

Table 4
Brief description of I-Demog implementations

	Flags	Enhanced correlation
Item-based	$\alpha = 1, \beta = 0, \gamma = 0$	$enh\_cor = adj\_cor$
I-Demog1	$\alpha = 0, \beta = 0, \gamma = 1$	$enh\_cor = adj\_cor * dem\_cor$
I-Demog2	$\alpha = 1, \beta = 0, \gamma = 1$	$enh\_cor = adj\_cor + adj\_cor * dem\_cor$
I-Demog3	$\alpha = 1, \beta = 1, \gamma = 1$	$enh\_cor = adj\_cor + dem\_cor + adj\_cor * dem\_cor$
I-Demog4	$\alpha = 1, \beta = 1, \gamma = 0$	$enh\_cor = adj\_cor + dem\_cor$

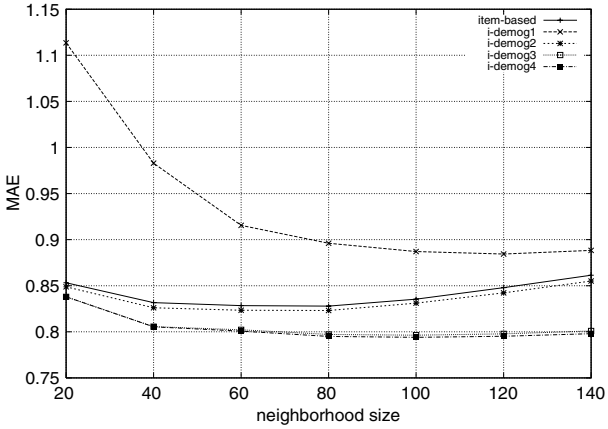

Fig. 1. Comparing different implementations of I-Demog with plain Item-based Filtering.

correlation flags will be utilized in the subsequent experiments, which intend to test whether I-Demog can be enhanced by the application of SVD at various points of the filtering procedure.

5.4.3. Experiments with IdemSvd-Dsvd: Applying SVD only on the demographic matrix

IdemSvd-Dsvd takes I-Demog and tests its behavior once SVD is applied on the matrix of the demographic vectors. The following experiments intend to, first, identify the optimal settings regarding the size of the neighborhood and the value of k, and then to compare our approach, optimally set, with the best predictions of plain Item-based Filtering.

5.4.3.1. Experiment 1: Identifying the optimal neighborhood size. It has been shown [15] that the size of the item neighborhood plays an important role in the recommendation procedure. The purpose of our first experiment was to locate the optimal neighborhood size to be utilized in subsequent experiments. To achieve that, we kept the value of k fixed to 6, where k corresponds to the number of dimensions retained for the demographic vectors, while setting the enhanced correlation flags, from Eq. (3), to the values which yielded the most accurate predictions, according to reported experiments [22]: $\alpha = 1$, $\beta = 1$, $\gamma = 0$. At the same time, we varied the size of the neighborhood, *neigh-size* $= \{20\text{–}140\}$. Fig. 2 depicts the generated Mean Absolute Errors, averaged over all five data splits.

From this figure we can observe that after an initially rapid and afterwards smoother improvement, the accuracy reaches its optimal value for a neighborhood of 80 items. For sizes bigger than that, the accuracy

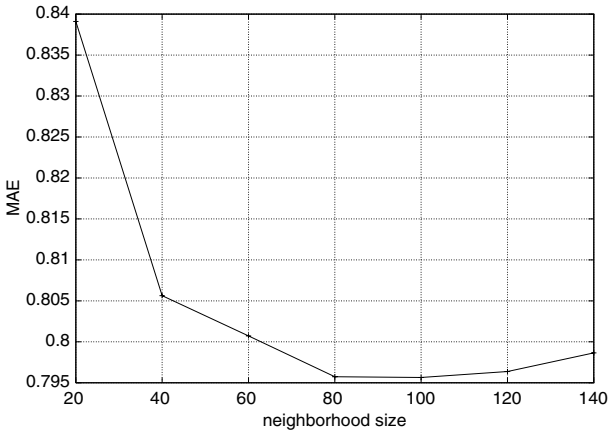

Fig. 2. IdemSvd-Dsvd: Identifying the best neighborhood size.

remains, in average, mostly unchanged or slightly worse. As a result, the rest of our experiments were executed with a neighborhood set to include 80 items.

5.4.3.2. Experiment 2: Identifying the optimal value of k. The second experiment's purpose was to identify the best value of k, which corresponds to the number of *pseudo*-features to be retained by the item demographic vectors after the application of SVD. As a result, we set the neighborhood size to the optimal values found by the previous experiment, assigned the appropriate values to the enhanced correlation flags ($\alpha = 1, \beta = 1, \gamma = 0$), and varied only the values of k, $k = \{1, 2, 4, 6, 8, 10, 12\}$. The generated Mean Absolute Errors, averaged over the five data splits, are displayed in Fig. 3.

Without forgetting that the original item demographic vectors include 18 distinct features, expressing the genres of the corresponding movies, the behavior of the line in Fig. 3 can be considered as rather surprising: it shows that, on average, only two demographic features can describe the item in mind adequately. This number is significantly lower than the 18 original features. Furthermore, there was a single data split which yielded its best accuracy for $k = 1$.

Based on these observations, we can assume that the demographic features, included in the utilized data set, provide information about the items which appear to be quite similar or, even, overlapping. Consequently, and by taking into account the resulting improvement in prediction accuracy, their merging should be recommended.

5.4.3.3. Experiment 3: Comparing IdemSvd-Dsvd with plain Item-based Filtering. Having identified the optimal parameter settings for k and the size of the item neighborhood, we can utilize those values in an initial comparison where the predictions by *IdemSvd-Dsvd* will be contrasted against those generated by plain Item-based Filtering. For the latter method, optimal parameter settings were also considered.

From Fig. 4, which includes the Mean Absolute Errors for *IdemSvd-Dsvd* and plain Item-based Filtering as observed for each of the 5 data splits, we can conclude that demographically enhanced Item-based Filtering, having its demographic vectors reduced by SVD, does indeed provide a considerable accuracy improvement over plain Item-based Filtering. It is in our intentions to find out which part of this improvement is attributed merely to the participation of the demographic vectors, and which part is owed to the dimensionality reduction applied on them.

5.4.3.4. Validating the optimal settings for k and neighborhood size. In order to statistically validate our decisions regarding the optimal settings for the selected parameters, i.e. k and *neighborhood size* ($n - s$), we executed *a 2-way Analysis of Variance* (anova), using these two parameters as factors. According to the first result we obtained, *there is no factor interaction*, and therefore we can proceed with an additive 2-way model, in order to check the effects of the factors, separately. Anova showed that there actually exist statistically significant effects, owed to the chosen factors.

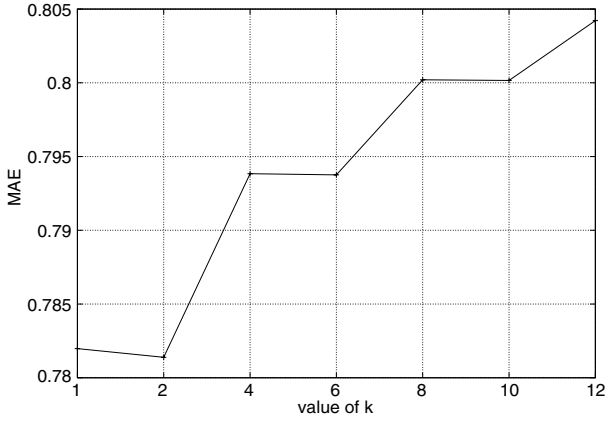

Fig. 3. IdemSvd-Dsvd: Identifying the best value of k.

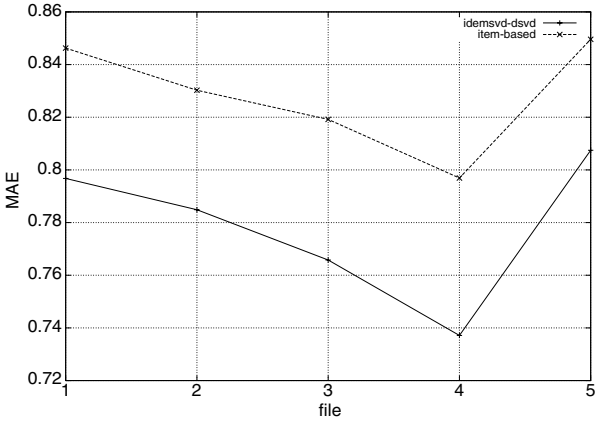

Fig. 4. Comparing IdemSvd-Dsvd with plain Item-based Filtering.

Specifically, there were significant effects for factor k, as evidenced by the corresponding P value ($P = 0.030$). *Tukey test* for pairwise comparisons formed three homogeneous subsets for the levels of factor k, namely $\{1,2\}$, $\{4,6,8,10\}$, and $\{12\}$, with the first subset of $\{1,2\}$ yielding a statistically smaller MAE, when compared to the rest. From the first subset, we can pick $k = 1$, retaining just a single *pseudo*-demographic feature.

At the same time, there were also significant effects for factor $n - s$. The corresponding P value was equal to 0. *Tukey test* for pairwise comparisons formed two homogeneous subsets for the levels of factor $n - s$, namely $\{10,20\}$ and $\{40,60,80,100,120,140\}$. The second subset yielded statistically smaller MAE. That is an interesting result, since it shows that we can pick any neighborhood size from this subset, instead of the clearly optimal size ($n - s = 80$), without a statistically significant loss in accuracy. As a result, we can select a neighborhood of 40 items, a setting that limits the executed calculations, compared to those required for a neighborhood of 80 items, and at the same time leads to equivalent accuracy results.

Table 5 highlights the results from the executed 2-way Analysis of Variance, including the tested factors, k and $n - s$, corresponding F and P values, etc.

Conclusively, an alternative optimal value combination for the two experimental parameters would be $k = 1$ and $n - s = 40$.

5.4.4. Experiments with IdemSvd-Rsvd: Applying SVD only on the user–item matrix

IdemSvd-Rsvd attempts to improve on I-Demog by applying SVD on the user–item matrix. The following experiments intend to, first, identify the optimal settings regarding the size of the neighborhood and the value of k, and then to compare our approach, optimally set, with the best predictions of plain Item-based Filtering.

5.4.4.1. Experiment 1: Identifying the optimal neighborhood size. The purpose of this experiment was to identify the best item neighborhood size before contrasting it with plain Item-based Filtering. To achieve that, we had to keep the value of k, corresponding to the low rank user–item matrix, fixed to 6, while setting the enhanced correlation flags according to the values reached in Section 5.4.2. At the same time, we varied the size of the

Table 5
2-Way Analysis of Variance for IdemSvd-Dsvd

Source	DF	SS	MS	F	P
k	6	0.007302	0.001217	2.37	0.030
$n - s$	7	0.138389	0.019770	38.57	0.000
Error	266	0.136330	0.000513		
Total	279	0.282021			

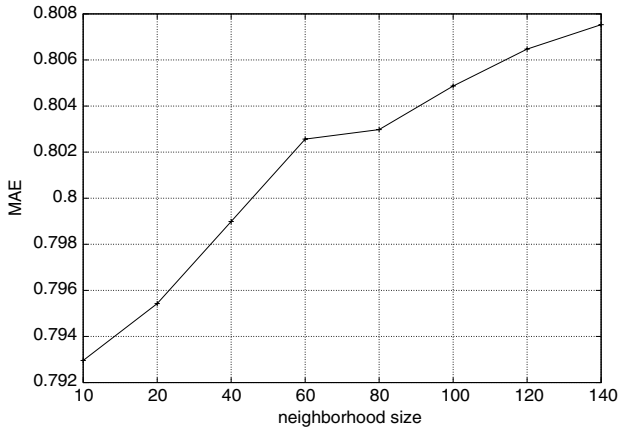

Fig. 5. IdemSvd-Rsvd: Identifying the best neighborhood size.

item neighborhood, *neigh-size* = {10–140}. Fig. 5 displays the generated Mean Absolute Errors (MAEs), averaged over all five data splits.

The MAE line in Fig. 5 does not follow the behavior commonly detected in similar Collaborative Filtering experiments, and illustrated in Fig. 2, according to which the accuracy improves as the neighborhood size increases, and remains stable or shows a slight decline after surpassing a certain threshold. In the case of IdemSvd-Rsvd, the accuracy starts with a low error value for a neighborhood including 10 items. Then it gets steadily worse for neighborhoods whose size continuously increases, until reaching the maximum size of 140 items. Therefore, the rest of our experiments in this section were executed by defining a neighborhood of 10 items.

5.4.4.2. Experiment 2: Identifying the optimal value of k. Our second experimental step involved trying different values of k, aiming to identify the one that would lead to the best accuracy. The GroupLens data set includes 943 distinct users, and k corresponds to the *pseudo*-users retained after applying SVD in order to reduce the dimensions of the original user–item matrix.

To initiate this experiment we set the neighborhood to its optimal size, according to the previous experiment, and assigned the appropriate values to the enhanced correlation flags. We only varied the values of k, $k = \{1, 2, 4, 6, 8, 10, 12, 14, 16\}$. The generated Mean Absolute Errors (MAEs), averaged over the five data splits, are displayed in Fig. 6.

For a better understanding of the line in Fig. 6 we can view it in correlation with the same experiment in IdemSvd-Dsvd (Fig. 3. The demographic matrix, D_i, includes only 18 features, and as a result, in

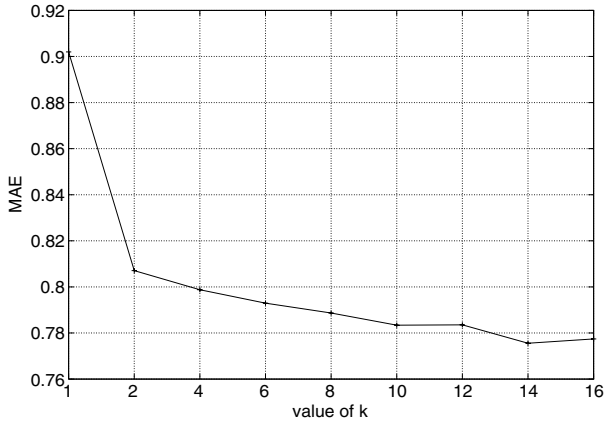

Fig. 6. IdemSvd-Rsvd: Identifying the best value of *l*.

IdemSvd-Dsvd we were able to reach an optimal accuracy value by retaining only two pseudo-features. In the case of IdemSvd-Rsvd, the matrix we are trying to reduce includes a considerably larger number of users ($943 \gg 18$). Thus, it is natural to require a comparatively bigger number of pseudo-users in order to capture a behavior which approximates optimally the behavior of the original matrix. Specifically, as shown from the figure, a value of k equal to 14 provides the best system accuracy.

5.4.4.3. Experiment 3: Comparing IdemSvd-Rsvd with plain Item-based Filtering. We have now identified the optimal parameter settings for k and the size of the item neighborhood. We can apply those values on *IdemSvd-Rsvd* and compare the generated predictions with those produced by plain Item-based Filtering, when also optimally tuned.

The obvious conclusion from Fig. 7, which includes the Mean Absolute Errors for *IdemSvd-Rsvd* and plain Item-based Filtering as observed for each of the five data splits, is that demographically enhanced Item-based Filtering, with its original user–item matrix reduced by SVD, does indeed provide a considerable accuracy improvement over plain Item-based Filtering. With further experiments we intend to specify how these results fare against other approaches which feature demographically enhanced Item-based Filtering, but allow SVD to affect different aspects of their filtering process.

5.4.4.4. Validating the optimal settings for k and neighborhood size. In continuation to what was discussed for *IdemSvd-Dsvd*, we wanted to statistically validate our decisions regarding the optimal parameter settings for the second proposed approach, *IdemSvd-Rsvd*. For that reason, we executed *a 2-way Analysis of Variance* (anova), using the two selected parameters, i.e. k and *neighborhood size* ($n - s$), as factors. The first experimental result showed *there is no factor interaction*. Therefore, we proceeded with an additive 2-way model, in order to check the effects of the factors, separately. This time, anova supported the hypothesis that there actually exist statistically significant effects, owed to the chosen factors.

Initially, there were significant effects for factor k, with the corresponding P value being equal to 0,032. *Tukey test* for pairwise comparisons distinguished a subset for the levels of factor k, including values $\{8, 10, 12, 14, 16\}$, as the one yielding statistically smaller MAE, when compared to the remaining levels of factor k. According to this experimental result, we are able to select any value for parameter k, among those available in this particular subset, without statistically significant loss in accuracy. Therefore, we can pick $k = 8$, which states that keeping eight *pseudo*-users, after the dimensionality reduction by SVD, is equally effective to selecting $k = 14$—a value that represents the lowest MAE, in absolute terms.

There were also significant effects for factor $n - s$. The corresponding P value was equal to 0.078. *Tukey test* for pairwise comparisons formed two homogeneous subsets for the levels of factor $n - s$, namely $\{10, 20, 40, 60\}$ and $\{80, 100, 120, 140\}$, with the former subset yielding statistically smaller MAE. Therefore, the recommended decision would be to select the smallest neighborhood, of 10 members, from this specific subset.

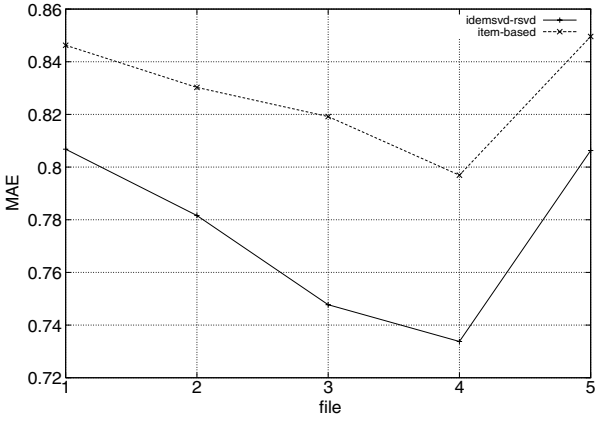

Fig. 7. Comparing IdemSvd-Rsvd with plain Item-based Filtering.

Table 6
2-Way Analysis of Variance for IdemSvd-Rsvd

Source	DF	SS	MS	F	P
k	6	0.008435	0.001406	2.34	0.032
$n-s$	7	0.007792	0.001113	1.85	0.078
Error	266	0.159964	0.000601		
Total	279	0.176190			

Table 6 highlights the results from the executed 2-way Analysis of Variance, including the tested factors, k and $n-s$, corresponding F and P values, etc.

Conclusively, an alternative optimal value combination for the two experimental parameters would be $k = 8$ and $n - s = 10$.

5.4.5. Experiments with IdemSvd-2svd: Applying SVD on both demographic and user–item matrix

IdemSvd-2svd can be characterized as demographically enhanced Item-based Filtering, which has both its demographic and user–item matrix reduced by SVD. The corresponding experiments include an additional step, when contrasted to those on IdemSvd-Dsvd and IdemSvd-Rsvd, since the parameters we have to tune before we compare this approach against plain Item-based Filtering, include (i) the size of the neighborhood, (ii) the value of k, which represents the dimensions retained after applying SVD on the user–item matrix, and (iii) the value of l, which represents the dimensions retained after applying SVD on the demographic vectors.

5.4.5.1. Experiment 1: Identifying the optimal neighborhood size. In the past two sections we tested the effect of a neighborhood size change for two implementations of IdemSvd (*IdemSvd-Dsvd* and *IdemSvd-Rsvd*), which are similar in that they apply SVD on a single point of the filtering procedure, but differ in that point of application. The experimental results were completely opposite. We could now run the same experiment on a filtering approach which applies SVD on two points of the procedure and observe its behavior. To achieve that, we set the values of k and l to 14 and 2, respectively, and only varied the size of the item neighborhood (*neigh-size* = {10–200}). Fig. 8 displays the generated Mean Absolute Errors, averaged over the five data splits.

The behavior of IdemSvd-2svd, as defined by the line in Fig. 8, differs from what reported for plain Item-based Filtering, and also by the corresponding experiment on IdemSvd-Dsvd (Fig. 2). The changes in the size of the neighborhood affect the system's accuracy in a way directly comparable to IdemSvd-Rsvd, which applies SVD only on the user–item matrix. Specifically, the lowest error values were observed for the smallest neighborhood size tested in our experiments (*neigh-size* = 10). As the size of the neighborhood was getting bigger, until reaching its maximum size (*neigh-size* = 200), the recorded error kept increasing. Therefore, the neighborhood selected for the subsequent experiments includes 10 items.

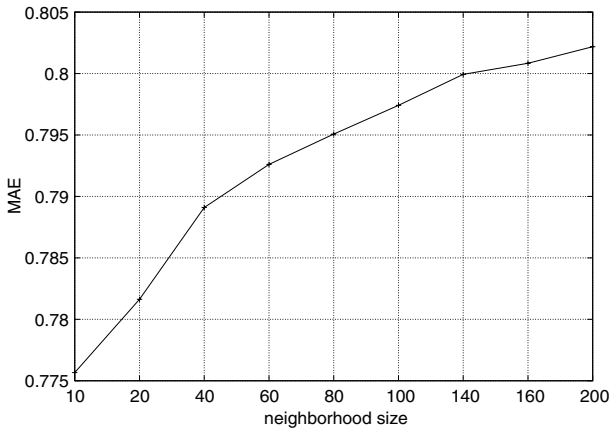

Fig. 8. IdemSvd-2svd: Identifying the best neighborhood size.

5.4.5.2. Experiment 2: Identifying the optimal value of k. Our second experiment with IdemSvd-2svd involved testing the algorithm for different values of k, which expressed distinct low rank representations of the original user–item matrix. According to the results of the previous experiment, we defined an optimal neighborhood of 10 items. We also assigned to l a random value of 2. We only varied the values of k, $k = \{1, 2, 4, 6, 8, 10, 12, 14, 16, 18\}$. The generated Mean Absolute Errors (MAEs), averaged over the five data splits, are displayed in Fig. 9.

In a result far from surprising, and similarly to the observations included in the previous experiment, the behavior of IdemSvd-2svd, when varying the value of k, appears to be very close to the one reported by the same experiment in IdemSvd-Rsvd (Fig. 6). In both cases we are defining the number of *pseudo*-users to retain, out of the 943 existing in the initial user–item matrix. Furthermore, the behavior of the line, which depicts a decrease of MAE as the value of k increases, makes sense: it only seems natural that a bigger k would allow for a better approximation of the original user–item space. According to these experimental results any value in the range of $k = \{14–18\}$ can be assigned to k without considerable, if any, accuracy loss.

5.4.5.3. Experiment 3: Identifying the optimal value of l. With our third IdemSvd-2svd experiment, we wanted to identify the optimal number of *pseudo*-features to retain from the original, 18-featured demographic vectors, expressed by l. We utilized the optimal values for k and the size of the neighborhood, as obtained from the past two experiments, and only varied the values of l, $l = \{1, 2, 4, 6, 8, 10, 12, 14, 16\}$. Fig. 10 depicts the generated Mean Absolute Errors, averaged over all five data splits.

We can easily conclude that there is no specific pattern followed by the MAE line: the error moves up and down repeatedly, reaching its lowest value for $l = 8$. This behavior is different from the one in Fig. 3, which depicted the error of the same experiment in IdemSvd-Dsvd. We should also note that the effect of l on the system accuracy is rather trivial. Any performance variation caused by it, is limited to the third decimal place of the MAE values, allowing us to describe it as a fine-tuning of accuracy results which were achieved by the previous experiments.

5.4.5.4. Experiment 4: Comparing IdemSvd-2svd with plain Item-based Filtering. At this point we have identified the optimal parameter settings for k, l, and the size of the item neighborhood. We will apply those values on *IdemSvd-2svd* in an attempt to compare its optimal predictions against those generated by plain Item-based Filtering. For the latter method, optimal parameter settings were also utilized.

Fig. 11 records the Mean Absolute Errors of *IdemSvd-2svd* and plain Item-based Filtering for each of the five data splits. Based on these MAE lines we can claim that demographically enhanced Item-based Filtering, with SVD applied to both user–item and demographic matrices, does indeed provide a considerable accuracy improvement over plain Item-based Filtering.

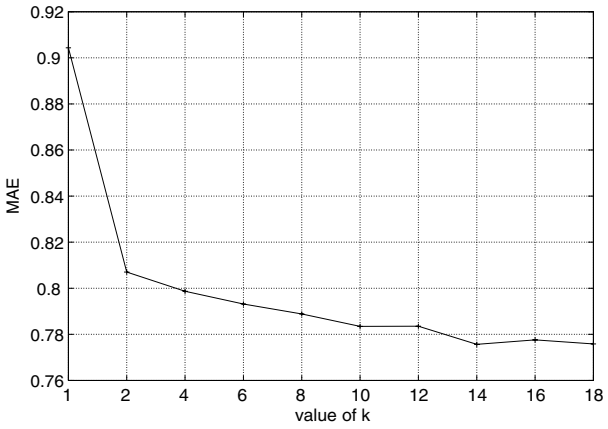

Fig. 9. IdemSvd-2svd: Identifying the best value of k.

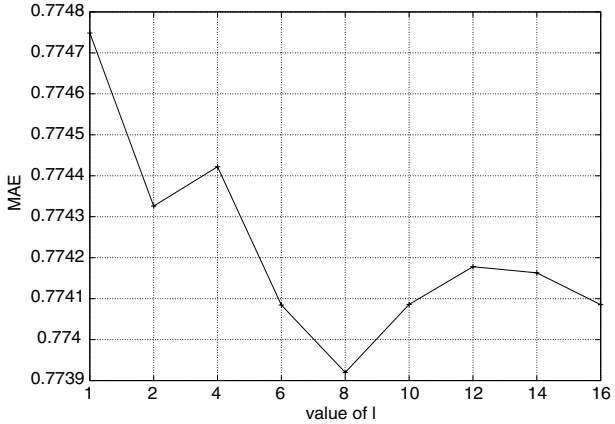

Fig. 10. IdemSvd-2svd: Identifying the best value of *l*.

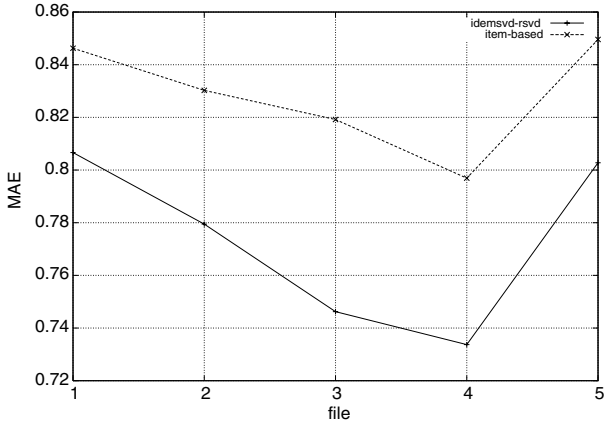

Fig. 11. Comparing IdemSvd-2svd with plain Item-based Filtering.

5.4.6. Overall comparison of IdemSvd implementations

The aim of this final section was to collect the most accurate results generated by the four *IdemSvd* implementations (*I-Demog*, *IdemSvd-Dsvd*, *IdemSvd-Rsvd* and *IdemSvd-2svd*) and utilize them in an overall comparison. According to the methodology followed in preceding sections, we selected plain Item-based CF as our base algorithm.

Fig. 12 depicts the lowest Mean Absolute Errors which were reported in each of these cases, for the five splits of the data set. The averages of these MAEs, which represent a single best mean accuracy value for each participating approach, are included in Table 7. Table 8 collects the corresponding execution costs. We note that m is the number of users, n is the number of items, and k is the number of *pseudo*-users retained after the application of SVD, where $k \ll n$.

In the cases of *Item-based* and *i-demog*, there is no application of SVD. At the same time, and contrary to what happens in the User-based approaches, the item–item correlation matrix is a lot less volatile. This means that we can compute the item correlations in less frequent intervals without affecting the overall performance of the system, which allows us to assign them to the off-line component. This move places an mn^2 cost under the *off-line* column. The *on-line* component can be now dedicated solely to prediction generation, which in the worse case induces an n^2 cost. Still, this cost is, on average, reduced to nl, where l corresponds to the size of the item neighborhood, which is usually $l \ll n$.

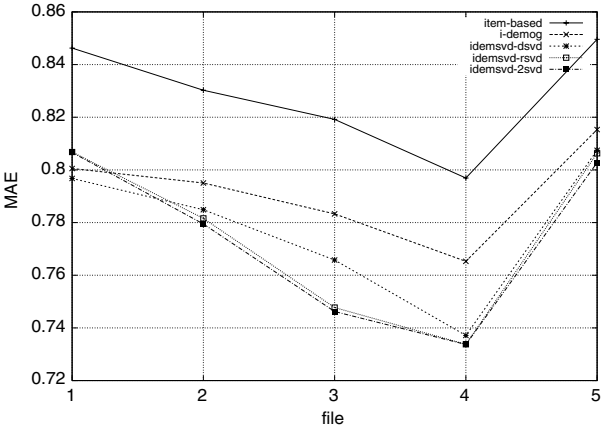

Fig. 12. Comparing all IdemSvd implementations.

Table 7
Best IdemSvd MAEs averaged over five data splits

	Item-based	i-demog	idemsvd-dsvd	idemsvd-rsvd	idemsvd-2svd
MAE	0.82843656	0.79189590	0.77840260	0.77523445	0.77376098

Table 8
Off and on-line costs for all five methods

	Off-line	On-line
Item-based	mn^2	n^2
i-demog	mn^2	n^2
idemsvd-dsvd	$[m^3] + [n^2m]$	n^2
idemsvd-rsvd	$[m^2n + m^3] + [n^2k]$	n^2
idemsvd-2svd	$[2(m^2n + m^3)] + [n^2k] + [n^2k]$	n^2

Regarding the three remaining methods (*idemsvd-dsvd*, *idemsvd-rsvd* and *idemsvd-2svd*), they all involve the application of SVD. The additional cost is equal to $m^2n + m^3$ for IdemSvd-Rsvd and equal to $2(m^2n + m^3)$ for IdemSvd-2svd, since SVD is applied twice. The situation is a bit more complicated for IdemSvd-Dsvd: we apply SVD on an $m \times 18$ matrix, of m items over 18 item demographic features. This leads to a cost of $m^2n + m^3 = 18m^2 + m^3 = m^3$.

For the cases of IdemSvd-Rsvd and IdemSvd-2svd, there are only k *pseudo*-users involved in the item similarities calculations after the application of SVD, which means that the corresponding costs would be reduced to n^2k. Still, and contrary to what discussed in User-based approaches, this reduction stays at the off-line component, and cannot be fully realized. The same cost remains at n^2m for IdemSvd-Dsvd, where no SVD is applied on the user–item matrix.

In the case of the GroupLens data set, which we utilized for our experiments, $m = 943$ and $n = 1682$. Therefore, we can assume that m and n are of the same order and adjust the execution costs of *IdemSvd-Rsvd* and *IdemSvd-Dsvd* to approximately n^3, and of *IdemSvd-2svd* to $2n^3$. The on-line costs could be adapted accordingly.

Bearing in mind the accuracy results from Fig. 12 and Table 7, along with the execution costs from Table 8, we can reach the following conclusions:

- *Plain Item-based Collaborative Filtering* has the lowest off-line execution cost, along with I-Demog. Still, it cannot be recommended since its accuracy ranks at the last place among those tested.

- *I-Demog*'s off-line cost is equal to that of plain Item-based CF, but its accuracy is significantly improved. Thus, we should prefer it in cases where we cannot handle the burden placed on the off-line component by the execution of SVD.
- Comparing the two filtering approaches which apply SVD *once* during the filtering procedure, we should select *IdemSvd-Dsvd* if we care for the lowest off-line cost, and *IdemSvd-Rsvd* if we prefer the best accuracy.
- *IdemSvd-2svd* may be the method with the lowest overall error, but, at the same time, it incurs the biggest off-line costs. Thus, among all approaches which apply SVD, we should prefer IdemSvd-2svd only when those off-line costs won't matter when compared to the improvement in the system's performance, or when we are able to calculate the off-line component at the least frequent time intervals.
- Conclusively, and contrary to the results reported after executing the same experiments in User-based CF, we can claim that SVD can successfully enhance the I-Demog algorithm. All three approaches we tested lead to a performance improvement over plain Item-based CF or I-Demog. Thus, we can recommend any of them under different circumstances, depending on the priorities we have set regarding the frequency of execution of the off-line component, the time which can be designated for the off-line component, and the prediction accuracy which can be achieved.

6. Conclusions and future work

The focus of this work was basically on the use of Singular Value Decomposition as a technique which can possibly enhance the performance of existing filtering algorithms. In the past, SVD was mainly combined with User-based Collaborative Filtering, proving to be an effective solution for recorded problems of Recommender Systems such as scalability and sparsity. With this work we extended the application of SVD to a new CF algorithm (Item-based CF). We also tested its effectiveness when combined with data other than the common user ratings on items, by utilizing it in collaboration with demographic information.

Our proposed method can be described as a filtering algorithm which utilizes SVD in order to reduce the dimensions of the original user–item and/or demographic matrix, while, at the same time drawing supplementary information possibly available in related demographic data. A claim stating that our approach can be considered a generalized filtering solution, was supported by identifying the appropriate assignments of the parameter values, which allow User- and Item-based CF to be viewed as distinct cases of it.

The subsequent experimental work proved that SVD can be blended smoothly with filtering methods which benefit from available demographic data. While the variations involving User-based CF did not produce particularly valuable findings, all Item-based filtering related approaches generated predictions yielding significantly lower error values. Thus, we can claim that the application of SVD on Recommender Systems has the potential to not only successfully tackle problems such as scalability and sparsity, but also to lead to measurable accuracy improvement.

Keeping these promising results as our starting point, it is in our intentions to experiment with Principal Component Analysis, as a viable alternative to SVD. We view it as a second method which can possibly enhance the filtering procedure, by assisting in dimensionality reduction.

In conclusion, we have to state that we view this work as an introduction towards a method that attempts to combine SVD with demographic data, in order to enhance plain Collaborative Filtering. As a result, additional tuning of the proposed approach, through experiments which may include alternative implementations regarding the encoding of the demographic data or the combination of ratings-based and demographic correlations, could not be included in the current paper, nevertheless being a welcome extension to it. Furthermore, past experiments have shown that the accuracy of Collaborative Filtering may depend on the selected data set. As a result, an important future task will involve testing the presented algorithms with different data sets, for their further evaluation. Finally, the statistical validation of optimal parameter settings, which was successfully applied to algorithms *IdemSvd-Dsvd* and *IdemSvd-Rsvd*, with the help of Anova and Tukey test, may also be extended to *IdemSvd-2svd*. We have to note, though, that this specific algorithm will further complicate the aforementioned procedure, since it incorporates an additional parameter, meaning that a *3-way Anova* will be required.

References

[1] Y. Azar, A. Fiat, A. Karlin, F. McSherry, J. Saia, Spectral analysis of data, in: Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing, Hersonissos, Greece, 2001, pp. 619–626.
[2] M.W. Berry, S.T. Dumais, G.W. O'Brien, Using linear algebra for intelligent information retrieval, SIAM Review 37 (1995) 573–595.
[3] D. Billsus, M.J. Pazzani, Learning collaborative information filters, in: 15th International Conference on Machine Learning, Madison, WI, 1998, pp. 46–53.
[4] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, 1998, pp. 43–52.
[5] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, M. Sartin, Combining content-based and collaborative filters in an online newspaper, in: ACM SIGIR Workshop on Recommender Systems-Implementation and Evaluation, Berkeley, CA, 1999, pp. 15–21.
[6] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science 41 (6) (1990) 391–407.
[7] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: A constant time collaborative filtering algorithm, Information Retrieval Journal 4 (2001) 133–151.
[8] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems 22 (2004) 5–53.
[9] W.P. Jones, G.W. Furnas, Pictures of relevance: a geometrical analysis of similarity measures, Journal of the American Society for Information Science 38 (1987) 420–442.
[10] Y. Jung, H. Park, D. zhu Du, An effective term-weighting scheme for information retrieval, Tech. rep., University of Minnesota, 2000.
[11] P. Melville, R.J. Mooney, R. Nagarajan, Content-boosted collaborative filtering, in: Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002), Edmonton, Canada, 2001, pp. 187–192.
[12] F. Meziane, Y. Rezgui, A document management methodology based on similarity contents, Information Sciences 158 (2004) 15–36.
[13] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Incremental singular value decomposition algorithms for highly scalable recommender systems, in: Fifth International Conference on Computer and Information Technology (ICCIT 2002), 2002, pp. 399–404.
[14] B.M. Sarwar, Sparsity, scalability, and distribution in recommender systems, Ph.D. thesis, University of Minnesota, 2001.
[15] B.M. Sarwar, G. Karypis, J.A. Konstan, J.T. Riedl, Analysis of recommendation algorithms for e-commerce, in: 2nd ACM Conference on Electronic Commerce, 2000, pp. 158–167.
[16] B.M. Sarwar, G. Karypis, J.A. Konstan, J.T. Riedl, Application of dimensionality reduction in recommender systems – a case study, in: ACM WebKDD 2000 Web Mining for E-Commerce Workshop, 2000, pp. 82–90.
[17] B.M. Sarwar, G. Karypis, J.A. Konstan, J.T. Riedl, Item-based collaborative filtering recommendation algorithms, in: 10th International World Wide Web Conference (WWW10), Hong Kong, 2001, pp. 285–295.
[18] A.I. Schein, A. Popescul, L.H. Ungar, D.M. Pennock, Methods and metrics for cold-start recommendations, in: ACM SIGIR-2002, Tampere, Finland, 2002, pp. 253–260.
[19] A. Selamat, S. Omatu, Web page feature selection and classification using neural networks, Information Sciences 158 (2004) 69–88.
[20] U. Shardanand, P. Maes, Social information filtering: Algorithms for automating 'word of mouth', in: Proceedings of Computer Human Interaction, 1995, pp. 210–217.
[21] S. Ujjin, P.J. Bentley, Particle swarm optimization recommender system, in: Proceedings of the IEEE Swarm Intelligence Symposium 2003, Indianapolis, 2003, pp. 124–131.
[22] M. Vozalis, K.G. Margaritis, On the enhancement of collaborative filtering by demographic data, Web Intelligence and Agent Systems, An International Journal 4 (2) (2006) 117–138.
[23] M. Vozalis, K.G. Margaritis, Collaborative filtering enhanced by demographic correlation, in: Proceedings of the AIAI Symposium on Professional Practice in AI, Part of the 18th World Computer Congress, Toulouse, France, 2004, pp. 393–402.
[24] M. Vozalis, K.G. Margaritis, Enhancing collaborative filtering with demographic data: The case of item-based filtering, in: Proceedings of the Fourth IEEE International Conference on Intelligent Systems Design and Applications (ISDA 2004), Budapest, Hungary, 2004, pp. 361–366.
[25] M. Vozalis, K.G. Margaritis, How svd and demographic data can be used to enhance generalized collaborative filtering, Tech. rep., University of Macedonia, Greece, 2004.
[26] M.E. Wall, A. Rechtsteiner, L.M. Rocha, Singular Value Decomposition and Principal Component Analysis, Kluwer, Norwell, MA, 2003, pp. 91–109.