An Adapted Alternation Approach for Recommender Systems*

Carme Julià, Angel D. Sappa
Computer Vision Center,
Edifici O, Campus UAB,
08193 Bellaterra, SPAIN
cjulia, asappa@cvc.uab.es

Felipe Lumbreras, Joan Serrat and Antonio López
Computer Science Department,
Universitat Autònoma de Barcelona
Computer Vision Center
Edifici O, Campus UAB,
08193 Bellaterra, SPAIN
felipe, joans, antonio@cvc.uab.es

Abstract

This paper presents an adaptation of the Alternation technique to tackle the prediction task in recommender systems. These systems are widely considered in Electronic commerce to help customers to find products they will probably like or dislike. As the SVD-based approaches, the proposed adapted Alternation technique uses all the information stored in the system to find the predictions. The main advantage of this technique with respect to the SVD-based ones is that it can deal with missing data. Furthermore, it has a smaller computational cost. Experimental results with public data sets are provided in order to show the viability of the proposed adapted Alternation approach.

1. Introduction

It is well known that the amount of information available on the Web increases constantly. Due to that fact, sometimes it becomes difficult to search for interesting information, while discarding useless content. Therefore, there is a clear demand for methods that give information with respect to users's preferences. Recommender systems target this demand by helping users to find items they would probably like or dislike using only a few given preferences. That is, users give rates only to some items and the system is capable of predicting their punctuation to the rest of items (this is called *prediction task*). Furthermore, it can also recommend them the products they would probably like most (*recommendation task*). These two powerful tools are widely used on *e-commerce* Web sites. Since their introduction in 1990s, the recommender systems have been used to filter information on the Web and to provide recommendations about books, CDs, movies, news, electronics, financial services, travel, etc. One of the most popular recommender system is the one used at: http://www.amazon.com. The customer rates some books and the system suggests him other books using information from other customers. One different recommender system is used at: http://www.everyonesacritic.net, where users give their opinion about movies and the system makes recommendations from people who share similar tastes. In fact, in most cases, the main goal of recommender systems is to discover which products will like a customer at most, in order to increase sales and therefore, benefits. Of course this helps also the customer, who will find, in theory, something useful to him.

In recommender systems, data are stored into a large table of users (also denoted as customers) and items (or products). Actually, the information is stored into a matrix of data, whose rows and columns correspond to each user and item respectively, and whose entries correspond to the rates that customers give to items. In real problems, there is a wide set of both customers and items. Hence it is necessary to deal with large matrices. Furthermore, since each user only rates a subset of the large set of items, most entries of the data matrix are empty and the matrix tends to be very sparse.

One technique widely used in recommender systems is the *collaborative filtering* (e.g., [11], [13], [6]), which is usually based on finding neighbourhoods of *similar* customers that are obtained by computing the correlation between their opinions. The similarity function is different in each approach. Although this technique is useful in many domains, it has a high computational cost and has limited prediction when dealing with very sparse data, as pointed out in [12] and [4]. In fact, Billsus et al. [2] identify two important limitations in the collaborative filtering techniques.

*This work has been supported by the Government of Spain under projects TRA2007-62526/AUT and Consolider-Ingenio 2010 (CSD2007-00018). The second author has been supported by The Ramón y Cajal Program.

978-0-7695-3395-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ICEBE.2008.25

The first one is that the correlation between two users rates can only be computed on items that both users have rated. Since, in general, there are thousands of items to rate, the number of overlapped items is small in most cases and the similarity measure is based on the correlation of only a few items. The second problem is that with this similarity measure, two users can only be similar if there is overlap among the rated items. As mentioned above, when the number of items to rate is large, it is difficult to obtain overlap among the rates.

Billsus et al. [2] present collaborative filtering in a machine learning framework to tackle the aforementioned limitations. Their proposed approach is based on the Singular Value Decomposition (SVD) [7]. Other recommender systems use the SVD (e.g., [10], [1], [12]) to reduce the data representation and give predictions using linear regression. The main advantage of the SVD is that, not only the information of correlated customers are used, but also the obtained from users whose ratings are not correlated, or who have not even rated anything in common. The SVD allows to project user ratings and rated items into a lower dimensional space. Thus, some users become predictors for other users's preferences even without any overlap of rated items. Unfortunately, computing the SVD of a large matrix requires a high computational cost. Furthermore, all the data must be known. Therefore, in order to be able to apply SVD, the missing data must be filled in. Some approaches add zeros in the missing entries, while others fill them with the corresponding row or column average (e.g., [12]). Then, these previously filled in missing entries are actualized with the SVD.

A different approach, based also in the SVD, is proposed by Brand in [4]. Actually, it is an imputation method developed in [3] to predict the position of occluded features in computer vision problems. In [4], it is used in data mining tasks. Concretely, Brand presents a method for adding data to a *thin* SVD[1] data model, which is significantly quicker and economical than the full SVD. Instead of computing the SVD of a large matrix, he develops an exact rank-1 update which provides a linear-time construction of the whole SVD. The approach begins by sorting out the rows and columns of the data matrix so that a high density of data is accumulated in one corner. Then this initial full submatrix grows out of this corner by sequential updating with partial rows and columns. An imputation update that maximizes the probability of correct generalization is used to fill in the missing entries. One disadvantage of this technique is that, as pointed out in [8], the result depends on how data are sorted.

In addition to the aforementioned limitations, one problem of the collaborative filtering and most recommender systems in general, is that they do not model properly a human-to-human interaction, where the user and the adviser interact. Focusing on this problem, *conversational recommender system* have been recently proposed (e.g., [5], [9]). These systems provide dialogues supporting the customer in the selection process. Thus, by adding user feedbacks, the system can make a better idea of the type of product the user may be interested in. This kind of recommender systems is out of scope of this work.

The current paper presents an approach that, as the SVD-based approaches, uses the information from all the users, not only from the correlated ones. Notice that the *prediction* task in recommender systems can be seen as a way of filling in the missing entries in the data matrix. In particular, an adaptation of the *Alternation Technique* [8] is proposed as imputation method for recommender systems. Hence, missing rates are predicted and the dimension of the data is reduced. One of the advantages of the Alternation technique over the SVD is that, since it can deal with missing data, it is not necessary to fill in the missing rates with zeros or averages before applying it. On the other hand, its computational cost is not as high as in the case of the SVD. The proposed approach is focused on the *prediction* (not *recommendation*) and its performance is compared with one of the method proposed in [12].

This paper is structured as follows. First of all, the approach proposed in [12] is briefly introduced. Then, the proposed adaptation of the Alternation technique to the problem of prediction task in recommender systems is presented. Data sets and experimental results are provided. Finally, concluding remarks are summarized.

2. A SVD-based Approach: Sarwar et al.'s Proposal

Since the proposed adaptation of Alternation technique to recommender systems is later on compared to the Sarwar et al.'s [12] approach, this Section presents briefly their proposal.

For the shake of simplicity, it will be referred to as Sarwar's approach hereinafter. As mentioned above, this approach is based on the SVD. Hence, first of all, given a matrix of rates W that contains missing data, its missing entries must be filled in.

Sarwar et al. [12] propose to fill in the missing data with the corresponding column average. Then, the entries are normalized by subtracting the corresponding row average. Once the data matrix W has not missing entries, the SVD is computed, given the decomposition: $W = U\Sigma V^t$. The best rank-r approximation to W is obtained by keeping only the r largest singular values of S, giving Σ_r. Accordingly, the

[1] Given a matrix $W_{m\times n}$, being $n \ll m$, the *thin* SVD consists in computing only the n column vectors of U corresponding to the row vectors of V^t. That is: $W_{m\times n} = U_{m\times n}\Sigma_{n\times n}V^t_{n\times n}$. The remaining columns of U are not computed.

dimensions of U and V are also reduced and the matrix: $W_r = U_r \Sigma_r V_r^t$ is the closest rank-r matrix to W.

Finally, the predicted rate $W(i,j)$ of the customer i to the product j is obtained with the following expression:

$$W(i,j) = \overline{w}_j + (U_r \Sigma_r^{1/2})(i)(\Sigma_r^{1/2} V_r^t)(j) \qquad (1)$$

where $\overline{w}_j$ is the jth-column average.

One disadvantage of this approach is that the computational cost is very high. Due to that fact, with concrete data sets, the number of customers and products must be reduced. In this paper, this approach is implemented by using the aforementioned *thin* SVD, in order to reduce its computational cost.

3. Adaptation of the Alternation technique to recommender systems

This Section proposes a variant of the Alternation technique [8], adapted to the *prediction* task in the recommender systems.

Given a matrix of rates $W_{c \times p}$, the goal of the Alternation technique is to find the factors $A_{c \times r}$ and $B_{r \times p}$ such that minimizes the expression $\|W_{c \times p} - A_{c \times r} B_{r \times p}\|_F^2$, where r is the rank of the data matrix. The product AB is the best rank-r approximation of the matrix W in the sense of the Frobenius norm [7]. Although the aim is the same as the one of the SVD, the Alternation allows to deal with missing data and additionally, it has a far smaller computational cost. As in the case of the SVD, A and B are r dimensional representations of customers and products, respectively.

The Alternation technique is a two-step algorithm, which starts with one random factor (A_0 or B_0) and compute one factor at a time, until the product AB converge to W. The product of the factors gives a filled matrix: $W_{imputed} = AB$. Focusing on the recommender systems, it can be used the fact that the entries in W take values in a known interval: $[m, M]$, where m and M are the minimum and the maximum rate values, respectively. The idea is to enforce that the range of values of the entries in $W_{imputed}$ lies in this known interval. This is tackled later.

First of all, at each step of the Alternation algorithm, the rows of A (B respectively) are normalized. The proposed adapted Alternation is summarized in the following steps:

Algorithm: Given a data matrix $W_{c \times p}$, which contains the rates given by different customers:

1. Take a random matrix A_0, normalize its rows.
2. Compute B_k from ${A_{k-1}}^2$:
$$B_k = (A_{k-1}^t A_{k-1})^{-1} A_{k-1}^t W \qquad (2)$$
3. Normalize the rows of B_k.
4. Compute A_k from ${B_{k-1}}^2$:
$$A_k = W B_k^t (B_k B_k^t)^{-1} \qquad (3)$$
5. Normalize the rows of A_k.
6. Repeat the steps 2-5 until the product $W_{imputed} = A_k B_k$ converges to W.

Solution: $W_{imputed}$ contains the predicted rates values.

Notice that the classical Alternation algorithm consists in applying repeatedly the steps 2 and 4 from above until the convergence is achieved.

Hence, with the normalization steps added to the algorithm, each entry of the imputed matrix $W_{imputed}(i,j)$ is the scalar product between the ith–unitary row of A and the jth–unitary column of B. That is:

$$W_{imputed}(i,j) = \mathbf{a}_i \cdot \mathbf{b}_j = \cos(\alpha_{ij}) \qquad (4)$$

where $\mathbf{a}_i$ and $\mathbf{b}_j$ are the ith–unitary row of A and jth–unitary column of B, respectively and α_{ij} could be interpreted as the angle between them (if they are considered as vectors).

Therefore, if no restrictions are added, the entries of $W_{imputed}$ would take values in the interval $[-1, 1]$. However, as mentioned above, the aim is to achieve that the values in $W_{imputed}$ lie in the interval $[m, M]$. Hence, the values of the entries in $W_{imputed}$ should be transformed. One possibility is to transform them directly by using a linear transformation. The problem is that the $cosine$ function is not linear. Instead of the $cosine$, the angles are studied. This means that the $arccosine$ must be applied to obtain the angles. Therefore, an interval where the $cosine$ is invertible should be defined: the initial punctuation ratings must be transformed in order to take values in the interval $[0, 1]$ instead of $[-1, 1]$.

In addition, and in order to weight equally all the possible values, instead of considering values in $[m, M]$, the interval $[m - 0.5, M + 0.5]$ is taken (otherwise, the values m and M are not considered as the rest of values).

The following steps summarize the transformation that gives the initial values from the interval $[m, M]$ to $[0, 1]$:

1. $\widetilde{W} = \frac{(W-m)}{\Delta}$, where $\Delta = M - m$, $\widetilde{W}(i,j) \in [0, 1]$
2. $\widetilde{W} = \pi \widetilde{W}$, $\widetilde{W}(i,j) \in [0, \pi]$
3. $\widetilde{W} = \cos(\widetilde{W})$, $\widetilde{W}(i,j) \in [0, 1]$

[2] These products are computed only considering known entries in I.

The adapted Alternation technique is applied to the transformed matrix $\widetilde{W}$ and $\widetilde{W}_{imputed}$, which contains the prediction rates, is obtained. Finally, the above transformations must be undone:

$$W_{imputed} = \frac{\arccos(\widetilde{W}_{imputed})}{\pi}\Delta + m \qquad (5)$$

Hence, the values in $W_{imputed}$ lie in $[m, M]$, as the values in the initial missing data matrix W. Notice that the $arccosine$ can be applied because it is invertible in the interval $[0, \pi]$.

4. The r selection

The rank used to obtain the best predicted rates depends on the approach used in the prediction task.

Unfortunately, the rank of the matrix is not known a priori and it can not be directly computed, when working with missing data. Hence, the problem of estimating the rank of a missing data matrix W should be faced out.

Sarwar et al. [12] point out that it is important to choose the optimal r in order to obtain good predicted values. They search for a r-value large enough to capture all the important information in the matrix, while small enough to avoid overfitting errors. However, results obtained with their method do not vary so much considering different rank values.

In the incremental SVD presented by Brand [4], it is proposed to use the rank value as a measure of complexity of the model. The objective is to maximize the probability of correct generalization, while minimizing the complexity of the model. Furthermore, Brand points out that users ratings data have poor repeatability from day to day. Therefore, a good low-rank approximation of the data has higher probability of generalization than a medium-rank model that perfectly reconstructs the data. His experiments show that the incremental SVD, with rank 4 or 5, predicts the missing rates better than matrices with higher rank. Brand points out that the higher the singular values are, the more constrained the imputation is by previous inputs, and therefore, the better the estimated SVD. With only a few rated items, the SVD has small singular values. In general, in those cases, a smaller rank will give better predicted rates.

If W has rank r and the Alternation technique is used to approximate it by a rank $\tilde{r}$ matrix, being $\tilde{r} > r$, noise is added to the data during the process, in order to achieve a higher rank matrix. Consequently, the missing entries are wrongly filled in. On the contrary, if the matrix is approximated by a rank-$\tilde{r}$ matrix, with $\tilde{r} < r$, information is lost during the process. The missing entries are again wrongly filled in. Hence, the goodness of the predicted values obtained with the Alternation depends on the used rank value. In Section 6, it will be shown that using the proposed adaptation of the Alternation for recommender systems, the best predicted rates are obtained, in general, for $r = 4$ or $r = 5$, as in [4]. In extreme cases, with large amount of missing data, $r = 3$ or $r = 2$ are enough.

Having in mind the importance of a proper rank selection, our experimental results and comparisons have been performed considering a range of different rank values. The error considering each rank value is computed. Finally, the smallest rank for which a minimum error is obtained is selected. Actually, a similar procedure is carried out in the aforementioned approaches. The case $r = 1$ is not considered, for any approach, since it makes no sense to project the data onto a 1-dimensional subspace.

5. Data sets

Data sets used in the experiments are introduced in this Section. Concretely, three different public data sets are considered.

The first data set is the one provided by the MovieLens recommender system (http://www.movielens.umn.edu), which is a Web-based research recommender system. One of the data sets they give consists of 100,000 ratings (discrete values from 1 up to 5) from 943 users and 1,682 movies. A user-movie matrix W, formed by 943 rows and 1,682 columns, is constructed. Each entry $W(i, j)$ represents the rating (from 1 up to 5) of the ith-user on the jth-movie. This data set is also used in [12] and [4]. Since the goal is to study the goodness of the obtained predicted values, some entries are randomly removed and used to study the corresponding recovered values. These entries form the test data set. The rest of the entries used to recover the data are the train data set. Concretely, five different train and test data sets, split into 80,000 train and 20,000 test cases, are also given at: http://www.movielens.umn.edu. The initial data matrix has 93.69% of missing data, while using any train data set, a matrix of 94.95% of missing data is obtained.

The second data set is the one obtained from BookCrossing, a service where book lovers exchange books all around the world and share their experiences with others (http://www.bookcrossing.com). Ziegler et al. [13] collect data from 278,858 members of BookCrossing, referring to 271,379 different books. A total of 1,157,112 rates are provided. These rates take implicit (0) and explicit (from 1 up to 10) discrete values. The data set used in [13] is available at: http://www.informatik.uni-freiburg.de/∼cziegler. If all available data are used, the obtained data matrix is extremely sparse (concretely, it has a percentage of missing data of about 99.9968%). In our experiments, a smaller matrix with a higher density of known data, is considered. In particular, it is required that each user rates a minimum of books. At the same time, only the books that have been rated by a minimum of users are considered. Different minimum values will be considered in the experiments, as presented in Section 6.

Finally, the last data set used in the experiments is obtained from Jester, an online joke recommender system: http://eigentaste.berkeley.edu. The complete data set is publicly available at: http://www.ieor.berkeley.edu/∼goldberg/jester-data/ . Concretely, 4.1 million continuous rating (from -10 up to 10) of 100 jokes from 73,421 users are provided. In this case, different users (rows) are selected randomly, given a matrix with smaller dimensions. The data set is presented in [6], where Goldberg et al. present a collaborative filtering algorithm based on the principal component analysis (PCA) to obtain the predictions in the Jester recommender system. In particular, the authors propose to project the data into the eigenplane with the PCA. Then, the projected data are clustered by using a recursive rectangular clustering. When a new user ask for recommendations, first, the rates the user gives are projected onto the eigen plane. Then, the representative cluster of the user is found. Finally, recommendations are computed from rates collected in the cluster.

6. Experimental results

As mentioned above, this work is focused on the *prediction* task in recommender systems. The performance of our approach is compared with the method presented in [12]. For the comparison, the Mean Absolute Error (*MAE*) is used as a measure of goodness of the recovered values. This is the measure of goodness used in most proposed approaches for recommender systems and it is defined as follows:

$$MAE = \frac{1}{N}\sum_{i,j} |P_{ij} - W_{ij}| \qquad (6)$$

where i, j correspond to the indexes of the artificially removed entries in W (test data set), N is the number of these removed entries and P_{ij} is the obtained predicted value for the entry W_{ij}.

Due to the different nature of the data in each data set, different experiments are carried out with each one and the obtained results are presented separately in every data set. For instance, the percentage of available data and the size o the matrices are different in each data set. Another characteristic that should be taken into account is that the rates can take discrete or continuous values.

6.1. MovieLens Data Set

In this data set, rates take integer values from 1 up to 5 and the percentage of missing data is about 94.05%. With such amount of missing data, experiments considering different percentages of missing data would not give significant conclusions. Five different train/test sets are used in the experiments and the mean of all the train/test sets are given.

Different r-dimension values (equivalently, r-rank values) are tested (from 2 up to 20) and the one for which the *MAE* is minimum is chosen.

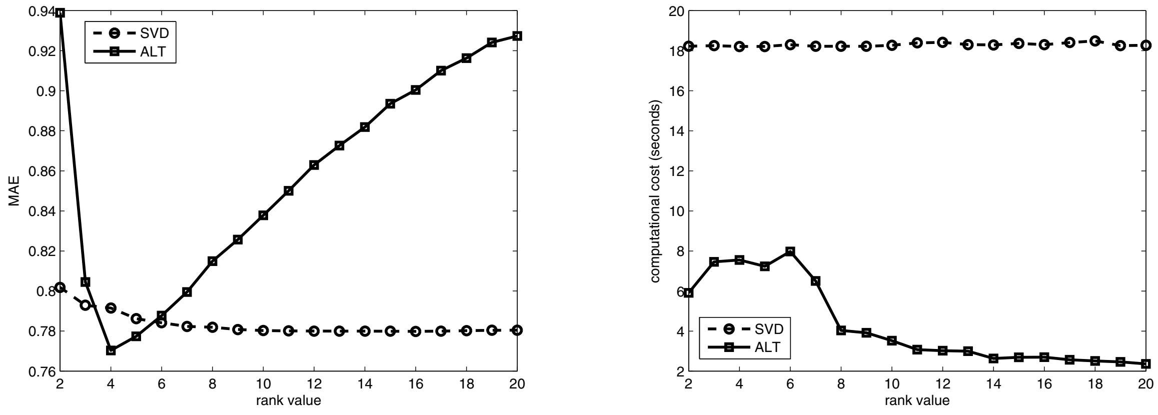
The mean of the obtained results in the five test/train data sets, for each rank value, is plotted in Fig. 1. The minimum error (*MAE*) obtained with the proposed approach (denoted as ALT in the plots) is smaller than the one obtained with the approach proposed in [12] (denoted as Sarwar's approach and as SVD in the plots), as it can be seen in Fig. 1 (a). Concretely, with the adapted Alternation approach, the smallest *MAE* value is obtained in the rank-4 case and its value is 0.7703. With Sarwar's approach, the minimum error (*MAE* = 0.7772) is obtained in the rank-16 case. The results presented in [12] are a little different: the obtained *MAE* is about 0.7400 for the rank-14 case. This difference is possibly due to different test/train splits. Notice that, in both approaches, a *MAE* with this value means an error of about ± 1 in the prediction task. As pointed out in [4], this is very accurate, since difference may reach ± 2 values if the user is asked in different days.

From our results, it can be concluded that the data space can be reduced to a 4-dimensional subspace, as it is also concluded in [4], where this data set is also studied. Unfortunately, open source of that algorithm is not available for a fair comparison. Sarwar et al. [12], found that a 14-dimensional subspace is needed in order to capture all the variance in the data. In fact, in their approach, the *MAE* does not change much considering different dimensions, as it is shown in Fig. 1 (a).

The computational cost for both approaches is depicted in Fig. 1 (b). It can be seen that our method is clearly faster, for all the tested rank values.

6.2. BookCrossing Data Set

The matrix obtained with this data set is very sparse. Concretely, it has a percentage of missing data of about 99.9968%. In order to obtain a matrix with more density of data, the users that have rated less than 20 books are discarded, while at the same time, only the books that have been rated by at least 200 users are considered. Since books and users are discarded at the same time, the above conditions do not means that every row and column have more than 20 and 200 known entries, respectively. In fact, with this two conditions, the number of considered users (rows) and books (columns) are 17,197 and 193, respectively and the percentage of missing data is about 98.5094%. Again, 5 different test/train data sets are generated. Concretely,



(a) (b)

Figure 1. MovieLens Data Set: (a) MAE considering different rank values; (b) computational cost in seconds.

the test data set contains 0.4906% of known data, while the train data set only 1% of known data.

Fig. 2 (a) shows the obtained MAE values, considering different rank values when a matrix with a 98.5094% is considered. It can be seen that the minimum error is obtained with the Alternation, for $r = 2$. Concretely, $MAE = 3.5811$. With Sarwar's approach, the minimum error is also achieved for $r = 2$ and $MAE = 3.7535$. With only a 1% of known data, a 2-rank matrix predicts better the missing rates than a matrix with a higher rank.

Notice that the error in this case is larger than in the experiments with the MovieLens data set. However, since the rates lie in different ranges, another measurement should be defined. Goldberg et al. [6] propose to use the *Normalized Mean Absolute Error* ($NMAE$) in order to compare errors obtained from different data sets. The $NMAE$ is defined as:

$$NMAE = \frac{MAE}{M - m} \tag{7}$$

where M and m are the maximum and minimum value in the range of rates, respectively. Using this new measure of error, the results obtained with the proposed approach and both data sets can be compared: in the case of the MovieLens data set, $MAE = 0.7704$, which gives $NMAE = 0.1926$, while in the case of BookCrossing data set, the obtained MAE is 3.5811, which gives $NMAE = 0.3581$. Effectively, the error with this second data set is higher than with the MovieLens' one. Recall that with this second data set, the percentage of missing data is higher.

In order to test different data matrices, a similar experiment is carried out, requiring a smaller number of rates per book. Concretely, books from the original matrix that have been rated by less than 150 users are discarded. Hence, in this experiment, the number of considered users and books are 21,026 and 354, respectively and the percentage of missing data is about 99.0465%. Hence, the obtained matrix has larger dimensions and more missing data. Concretely, test data set contains 0.1535% of data, while the train data set contains only 0.8% of data.

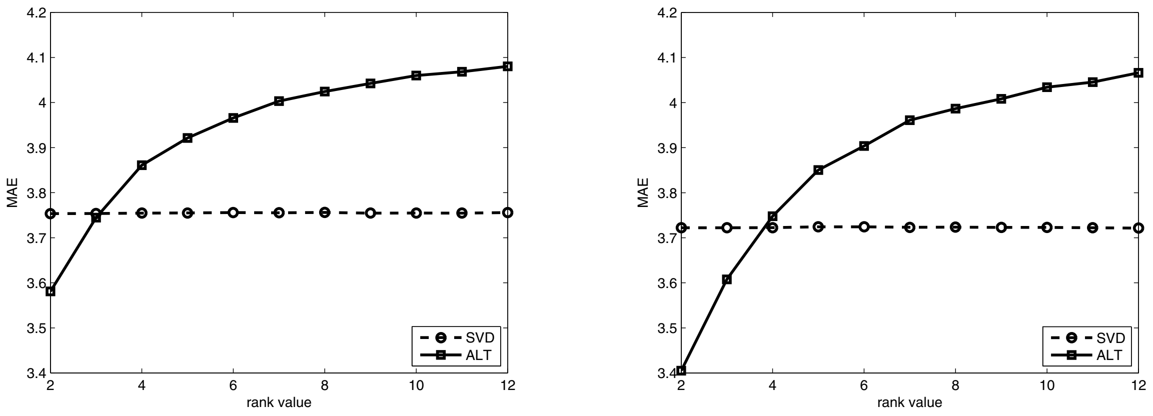
The obtained MAE values in this case are plotted in Fig. 2 (b). Notice that the minimum MAE is obtained with the Alternation, for $r = 2$. Its value is 3.4051. With the Sarwar's approach, the $MAE = 3.7215$ and it is obtained for $r = 12$. However, similar results are obtained for any rank value.

6.3. Jester Data Set

The rates in this data set take continuous values from -10 up to 10. The obtained matrix contains the rates given by 73,421 users to 100 jokes and the percentage of known data is about 48%.

Different percentages of missing data are generated by randomly removing data (concretely, 60%, 70%, 80% and 90%). The removed entries form the test data set, while the rest of the data form the train data set. Again, 5 different train/test data sets are considered at each experiment. Only 18,000 users from the total of 73,421 are randomly selected for the experiments, as in [6].

Fig. 3 shows the error (*MAE*) value obtained considering different percentages of missing data and different rank values (from 2 up to 20). It can be seen that in the case of Sarwar's approach, the obtained *MAE* is quite similar for any rank value. Concretely, the minimum *MAE* is achieved with



(a) (b)

Figure 2. BookCrossing Data Set: obtained *MAE* considering different rank values; (a) the data matrix has 98.5094$\%$ of missing data; (b) the data matrix has 99.0465$\%$ of missing data.

$r = 14$ and $r = 12$, with 60% and 70% of missing data. With percentages of missing data of about 80% and 90%, the minimum MAE is obtained with $r = 3$ and $r = 2$, respectively. In the case of the Alternation, the MAE value depends on the rank value, as it can be appreciated in Fig. 3. The minimum MAE is obtained for $r = 5$, while the percentage of missing data is below 90%, in which case, the minimum MAE is obtained with $r = 4$ (see Fig. 3 (d)).

The obtained *MAE* values are similar to the ones presented in [6], which studies the same data set. Unfortunately, no comparison can be performed with [6], since the authors do not provide an accurate information about the percentage of missing data they consider, nor the used rows (they select 18,000 rows randomly).

Although the obtained MAE seems to be higher than the obtained with the MovieLens data set, the ratings lie in different ranges. If the $NMAE$ proposed by Goldberg et al. [6] is used, it can be seen that the results are similar with both data sets. In the case of the MovieLens data set, the values obtained with the proposed approach are as follows: $MAE = 0.7704$, which gives $NMAE = 0.1926$, while in the case of Jester data set, the MAE obtained with 90% of missing data is 3.8959, which gives $NMAE = 0.1948$. Therefore, the goodness of the predicted rates is quite similar in both cases.

7. Conclusions

The Alternation technique is adapted to tackle the *prediction* task in recommender systems. Concretely, a variant of this technique, which uses the fact that rates take values in a known interval, is presented. As in the SVD-based approaches, not only correlated customers are used in the *prediction* task, but also non correlated ones.

The proposed adapted Alternation is compared to the approach presented in [12]. Three different public data sets, obtained from three different recommender systems, are studied. Experimental results show that the proposed approach performs better than the SVD used in [12], both regarding the error value and also the computational cost. It can be concluded that the data space can be reduced to a low-dimensional subspace, in most cases, with the proposed approach.

It should be highlighted the good results obtained with the proposed adapted Alternation approach, even with percentages of missing data of more than 90%.

References

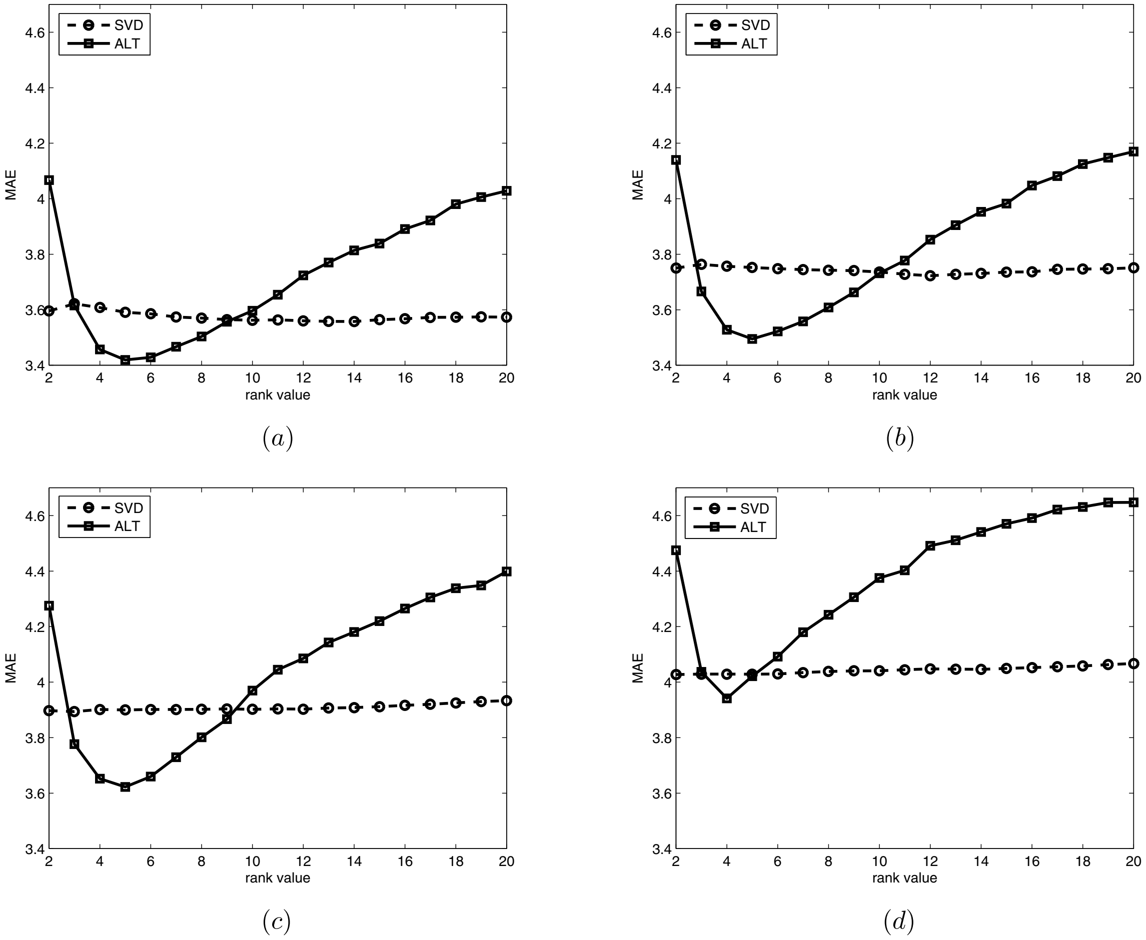
[1] M. Berry, S. Dumais, and G. O'Brien. Using linear algebra for intelligent information retrieval. In *Society for industrial and applied mathematics (SIAM)*, volume 37, pages 573–595, 1995.

[2] D. Billsus and M. Pazzani. Learning collaborative information filters. In *Proc. 15th International Conf. on Machine Learning*, pages 46–54, 1998.

[3] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *ECCV*, pages 707–720, 2002.

[4] M. Brand. Fast online SVD revisions for lightweight recommender systems. In *SIAM International Conference on Data Mining*, 2003.

[5] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker. An integrated environment for the development of knowledge-based recommender applications. *International Journal of Electronic Commerce*, 11:11–34, Winter 2006-2007.



(a) (b) (c) (d)

Figure 3. Jester Data Set: obtained *MAE* values for different rank values and different percentages of missing data: (a) 60%; (b) 70%; (c) 80%; (d) 90%.

[6] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. In *Information Retrieval*, volume 4, pages 133–151, 2001.

[7] G. Golub and C. Van Loan, editors. *Matrix Computations*. The Johns Hopkins Univ. Press, 1989.

[8] R. Hartley and F. Schaffalitzky. Powerfactorization: 3D reconstruction with missing or uncertain data. In *Australian-Japan advanced workshop on Computer Vision*, 2003.

[9] R. McGinty and B. Smyth. Adaptive selection: an analysis of critiquing and preference-based feedback in conversational recommender systems. *International Journal of Electronic Commerce*, 11:35–57, Winter 2006-2007.

[10] M. Pryor. The effects of Singular Value Decomposition on collaborative filtering. *Computer Science Technical Report*, 1998.

[11] P. Resnick, N. Iacovou, M. Suchak, M. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *Computer Supported Cooperative Work (CSCW)*, 1994.

[12] M. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system - a case study. In *E-Commerce Workshop*, pages 309–324, 2000.

[13] C. Ziegler, S. McNee, J. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *International World Wide Web Conference (WWW '05)*, 2005.