Predicting Missing Ratings in Recommender Systems: Adapted Factorization Approach

Carme Julià, Angel D. Sappa, Felipe Lumbreras, Joan Serrat, and Antonio López

ABSTRACT: The paper presents a factorization-based approach to make predictions in recommender systems. These systems are widely used in electronic commerce to help customers find products according to their preferences. Taking into account the customer's ratings of some products available in the system, the recommender system tries to predict the ratings the customer would give to other products in the system. The proposed factorization-based approach uses all the information provided to compute the predicted ratings, in the same way as approaches based on Singular Value Decomposition (SVD). The main advantage of this technique versus SVD-based approaches is that it can deal with missing data. It also has a smaller computational cost. Experimental results with public data sets are provided to show that the proposed adapted factorization approach gives better predicted ratings than a widely used SVD-based approach.

KEY WORDS AND PHRASES: Factorization technique, recommender systems, singular value decomposition.

Since the amount of information available on the World Wide Web increases constantly, sometimes it becomes difficult to focus on interesting information and discard redundant content. For this reason, there is a high demand for methods that select interesting information with respect to users' preferences. Recommender systems target this demand by helping users to find items, using previous knowledge about the user's preferences. Users give ratings only to some of the items and therefore the system is able to predict their preferences on the rest of items (this is known as *prediction* task). The system can also recommend products according to the user's preferences (*recommendation* task). These two powerful tools are widely used on e-commerce sites. Since their introduction in the 1990s, recommender systems have been used to filter information on the Web and to provide recommendations for books, CDs, movies, news, electronics, financial services, travel, and other products. One of the most popular recommender systems is the one at www.amazon.com. The customer rates some books and the system suggests other books, considering information from other customers. A different recommender system is used at www.everyonesacritic.net, where users give their opinion about movies and the system makes recommendations for people who share similar tastes. Another example (www.gnomoradio.org) consists of a music recommender system, where the user rates the music, and the system builds a

This work was supported by the government of Spain under projects TRA2007-62526/AUT and DPI2007-66556-C03-03, and research program Consolider-Ingenio 2010: MIPRCV (CSD2007-00018). The authors thank Nathan Faggian for providing an incremental SVD code.

International Journal of Electronic Commerce / Winter 2009–10, Vol. 14, No. 2, pp. 89–108.

1086-4415/2009 $9.50 + 0.00.
DOI 10.2753/JEC1086-4415140203

listening profile based on the user's ratings. In addition, it recommends music from other users with similar profiles. Thus, in most cases, the main goal of a recommender system is to discover the customer's preferred products in order to increase sales. This also helps customer, because they will only receive information filtered according to their individual taste.

Recommender systems store data in a large table of users (also denoted as customers) and items (or products). Hence, 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 ratings customers give to items. In real problems, the number of customers and items is huge, so it is necessary to deal with large data matrices. Since each user only rates a subset of the items, most entries in the matrix of ratings are empty, which means that the matrix tends to be very sparse.

Related Work

The technique of *collaborative filtering* is widely used in recommender systems (e.g., [8, 14, 19]). It is usually based on finding neighborhoods of *similar* customers, whose similarity is obtained by computing the correlation between their opinions. The similarity function is different in each approach. Although this technique is useful in many different domains, it has a high computational cost and its prediction is limited when dealing with very sparse data, as pointed out by Brand and by Sarwar et al. [3, 15]. In fact, Billsus and Pazzani identify two important limitations in the collaborative filtering techniques [2]. The first one is that the correlation between two user ratings can only be computed on items that both users have rated. Since there are generally thousands of feasible items to rate, the number of overlapped items is quite 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 ratings.

Billsus and Pazzani present collaborative filtering in a machine-learning framework to solve the aforementioned limitation [2]. Their approach is based on Singular Value Decomposition (SVD) [9] (details on SVD can be found in the Appendix). Other recommender systems use SVD to reduce data representation and give predicted ratings using linear regression (e.g., [1, 13, 15]). Sarwar et al. show the limitations of the classical collaborative filtering algorithms and propose to use SVD to deal with them [15]. Their experimental results show better performance of SVD with respect to collaborative filtering techniques when working with sparse matrices of ratings [15]. The main advantage of SVD is that it uses, not only information from correlated customers, but also information obtained from users whose ratings are not correlated. SVD makes it possible to project user ratings and rated items into a lower dimensional space. Thus, some users become predictors for other users' preferences even without any overlap of rated items. Unfortunately, computing the SVD of a large matrix requires a high computational cost, and also all the data must be known. Therefore, in order to be able to apply SVD, missing ratings must be filled in somehow. Some approaches add zeros in the missing entries, while others fill them with the corresponding row or column average (e.g., [15]). Then, these previously filled in missing entries are updated with the SVD. In a more recent paper, Sarwar et al. present an incremental SVD that aims at reducing the computational cost of the SVD [16]. The idea is to precompute an SVD decomposition by using the method in [15], considering a reduced number of users and items, which forms the model (known also as the basis). Then, this precomputed decomposition is used to perform predicted ratings for new users. The size of the basis of the precomputed SVD must be determined in order to obtain good predicted ratings—it should be small enough to produce a fast model and large enough to produce good prediction quality. Although this technique requires less time and storage space than the SVD, it can result in loss of quality due to the fact that the computed incremental SVD model is not orthogonal, as pointed out in [16].

A similar incremental SVD is proposed by Brand [3]. Actually, it was introduced by Brand to predict the position of occluded features in computer vision problems [4]. Specifically, Brand presents a method for adding data to a *thin* SVD data model (see Appendix for details), which is significantly faster than full SVD. Instead of computing the SVD of a large matrix, an exact *rank*-1 update, which provides a linear-time construction of the whole SVD, is computed. The main advantage with respect to [16] is that the obtained model is orthogonal, which provides better results. Moreover, Brand proposes to use some prior knowledge in order to obtain better predicted ratings. This approach first rearranges the rows and columns of the matrix of ratings so that a high density of data is accumulated in one corner of the matrix. This initial submatrix grows out of the 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. The main drawback of this technique is that the result depends on how the data are sorted (see [10]). Furthermore, the goodness of the predicted ratings depends on the size of the original full matrix, which tends to be very small in most cases, due to the sparse nature of the matrices of ratings.

In addition to the aforementioned limitations, a common problem of collaborative filtering techniques, and of most recommender systems, is that they do not properly model human-to-human interaction. In order to deal with this, *conversational recommender systems* have been recently proposed (e.g., [6, 12]). These systems provide dialogues supporting the customer in the selection process. By adding the user's feedback, the system can improve the prediction of the product the user may be interested in. This type of recommender system is beyond the scope of the present work.

Objective

The current paper presents an approach for dealing with the prediction task in recommender systems. It is an extension of the previously introduced method [11]. The proposed approach uses the fact that the problem can be reduced to a low-dimensional one. Thus, the goal consists in the approximation of the data matrix by a low-rank matrix. An adaptation of a factorization technique originally introduced for the Structure from Motion problem (SFM) [17] is used to find the low-rank matrix approximation. In particular, the *Alternation* technique [18], which has been widely studied in the computer vision framework (e.g., [5, 7, 10]), is used. Given a matrix of ratings W, the Alternation technique aims at finding the best A and B factors whose product results in the best low-rank matrix approximation of W. At the same time, missing entries in W are filled in with the product of the recovered factors AB. Hence, the prediction task in recommender systems can be seen as a way of filling in the missing entries in the matrix of ratings. The proposed approach uses information from all the users, not only from the correlated ones, in the same way as the SVD-based approaches. One of the advantages of the Alternation technique over the SVD is that it can deal with missing data, and thus there is no need to either fill in the missing ratings with zeros or averages before applying it or to begin with an initially full submatrix. In addition, its computational cost is not as high as the SVD approach. The performance of the proposed adapted factorization approach is compared with the SVD-based method proposed in [15].

An SVD-Based Approach

Drawbacks of Collaborative Filtering Techniques

As mentioned in the preceding section, Sarwar et al. report some of the limitations of the collaborative filtering techniques commonly used in recommender systems [15]. These techniques are based on finding neighborhoods of *similar* customers by computing a correlation coefficient. Sarwar et al. point out that in some cases, the correlation coefficient is defined between customers that have only rated few products in common [15]. This produces a limitation referred to as *sparsity* that is engendered by the fact that recommender systems generally work with large sets of products. Sarwar et al. also mention the *synonymy* limitation, which occurs because correlation-based systems would see no match between different product names that refer to similar objects. Finally, they comment on the *scalability* limitation: that is, the computational cost of the nearest-neighbor algorithms grows with the number of customers and also the number of products, which are both very large in most recommender systems. In order to solve these limitations, Sarwar et al. propose an SVD-based approach for recommender systems [15] that is briefly introduced in the next section.

Sarwar et al.'s Proposal

For the sake of simplicity, this approach will hereinafter be referred to as Sarwar's approach. Let W be a matrix of ratings where every W_{ij} corresponds to the rating given by the ith-user to the jth-product. In the case where W

contains missing ratings, these missing entries must first be filled in somehow (otherwise the SVD could not be applied).

Sarwar et al. propose to fill in the missing ratings with the corresponding column average (namely, product or item average), since this gives better results than using the row average (customer or user average) [15]. Then the entries are normalized by subtracting the corresponding row average. Once the data matrix W has no missing ratings, 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 $\Sigma(\Sigma_r)$. Accordingly, the 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. More details about the SVD can be found in the Appendix.

Finally, the predicted rating W_{ij} of the ith-customer to the jth-product is obtained with the following expression:

$$W_{ij} = \bar{w}_i + \left(U_r \Sigma_r^{1/2}\right)_i \left(\Sigma_r^{1/2} V_r^t\right)_j , \tag{1}$$

where $\bar{w}_i$ is the ith-row average of W.

A disadvantage of this approach is that the computational cost is very high. Due to this fact, with concrete data sets the number of customers and products must be reduced. In the current paper, Sarwar's approach is implemented by using the aforementioned *thin* SVD (see Appendix for details), in order to reduce its computational cost [15].

Summary

Experimental results show that Sarwar's approach, which is based on SVD, performs better than collaborative filtering techniques when the training data set is very sparse [15]. Furthermore, the SVD-based approach provides better on-line performance than collaborative filtering techniques [15]. On the other hand, collaborative filtering techniques perform slightly than the SVD-based approach when there are enough training data (more than 50%) [15].

Proposed Approach

This section proposes an adapted factorization technique to predict missing rates in recommender systems. Concretely, the Alternation technique is used to find the best low-rank matrix approximation of the matrix of ratings W [18].

Given a matrix of ratings $W_{c\times p}$, where c and p are the numbers of users and products, respectively, the goal of the Alternation technique is to find the factors $A_{c\times r}$ and $B_{r\times p}$ 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. Hereinafter, the size of the matrices is not specified, for the sake of simplicity. In the case of missing ratings in W, the expression to minimize is:

$$\left\| W_{ij} - (AB)_{ij} \right\|_F^2 , \tag{2}$$

where i, j corresponds to the index pairs where W is defined.

The product AB is the best rank-r approximation of the matrix W in the sense of the Frobenius norm [9]. Although the aim is the same as with SVD, the Alternation makes it possible to deal with missing data and, additionally, has a far smaller computational cost. As with SVD, A and B are r dimensional representations of customers and products, respectively.

The Alternation technique is a two-step algorithm that starts with one random factor (A_0 or B_0) and computes one factor at a time, until the product of the factors AB converges to W. A filled matrix $W_{imputed}$ is obtained with the product of these factors: $W_{imputed} = AB$. The algorithm is summarized below:

Alternation algorithm: Given a matrix of ratings $W_{c\times p}$, take a random $c \times r$ matrix A_0 (analogously with a random $r \times p$ matrix B_0) and repeat steps 1–2 until the product A_kB_k converges to W:

Compute B_k from $A_k - 1$:

$$B_k = \left(A_{k-1}^t A_{k-1}\right)^{-1} \left(A_{k-1}^t W\right) \tag{3}$$

Compute A_k from B_k:

$$A_k = WB_k^t \left(B_k B_k^t\right)^{-1} \tag{4}$$

Solution: $W_{imputed} = A_kB_k$ is the best rank-r approximation to W.

One of the main advantages of this two-step algorithm is that the updates of A given B (analogously B given A) can be done by solving a least-squares problem for each row of A independently (analogously each column of B). Therefore, missing entries in W correspond to omitted equations (note that products (3) and (4) are computed only considering known entries in W). The Alternation algorithm considering rows of A and columns of B independently is presented below.

Alternation algorithm, row-column formulation: Given a matrix of ratings $W_{c\times p}$, take a random $c \times r$ matrix A_0 (analogously with a random $r \times p$ matrix B_0) and repeat steps 1–2 until the product A_kB_k converges to W:

Given A_{k-1}, compute the columns of B_k independently:

$$b^j = \left(A_{k-1}^t A_{k-1}\right)^{-1} \left(A_{k-1}^t w^j\right), \tag{5}$$

where b^j is the jth-column of B_k, and w^j is the jth-column of W. Note that only the known entries in w^j and the corresponding rows in A_{k-1} are used.

Given B_k, compute the rows of A_k independently:

$$a^i = w^i B_k^t \left(B_k B_k^t\right)^{-1}, \tag{6}$$

where a^i is the ith-row of A_k, and w^i is the ith-row of W. Only the known entries in w^i and the corresponding columns in B_k are used.

Solution: $W_{imputed} = A_k B_k$ is the best rank-r approximation to W.

The Alternation technique converges to a global minimum when W is full or has a low percentage of missing ratings. With a large amount of missing ratings in W, the algorithm may fail to converge. Concretely, the Alternation technique gives a good rank-r approximation to W, while the number of known entries at each row/column is at least r. Initially missing ratings are wrongly estimated otherwise. This is due to the fact that at each step of the Alternation algorithm, the number of unknown entries (each row of A or column of B) is r. The number of equations to compute these unknowns is the number of known elements in the corresponding row of W when A is computed (or the corresponding column when B is computed). If the number of equations is smaller than the number of unknowns, a pseudo-inverse can be computed, but, in general, the result will not be the global minimum of the least-squares problem. Actually, in order to obtain good results, the number of equations should be higher than the number of unknowns.

In the particular case of recommender systems, matrices tend to have a high percentage of missing ratings, due to the large number of users and items. In preliminary attempts, the classical Alternation technique was applied to predict missing entries in the matrix of ratings. The problem was that, without using any prior knowledge, some predicted ratings take very large values. The proposed adapted factorization technique uses the fact that the ratings given by the users lie in an initially known range of values: [m, M], where m and M are the minimum and maximum ratings, respectively. The idea of the adapted factorization technique consists in enforcing that the obtained predicted ratings (entries in $W_{imputed}$) lie in this known interval. Hence, large predicted ratings are avoided.

At each step of the Alternation algorithm, the rows of A (columns of B, respectively) are first normalized by using the norm of each row of A (or column of B), respectively. Thus, the resulting rows and columns can be thought of as unitary vectors. Note that with the normalization steps added to the Alternation algorithm, each entry of the imputed matrix $(W_{imputed})_{ij}$ can be interpreted as the scalar product between the ith-unitary row of A and the jth-unitary column of B. That is:

$$\left(W_{imputed}\right)_{ij} = a_i b_j = \cos\alpha_{ij}, \tag{7}$$

where a_i and b_j are the ith-unitary row of A and the jth-unitary column of B, respectively, and α_{ij} could be interpreted as the angle between them (if they were considered as vectors).

Therefore, if no restrictions were added, the entries of $W_{imputed}$ would take values in the interval [–1, 1]. However, the aim is to achieve that the predicted ratings in $W_{imputed}$ lie in the $[m, M]$ interval, as the initially known ratings in W. Hence, the initially known ratings of W should be transformed in order to be interpreted as the scalar product of two vectors. The following transformation is applied to the matrix W:

$$\hat{W} = \cos\left(\pi\left(\frac{W-m}{M-m}\right)\right). \tag{8}$$

Since the cosine function is not linear, the angles will be recovered from the corresponding arccosine function. Therefore, an interval where the cosine is invertible has been defined: note that, by using the transformation (8), the cosine is applied to values in the $[0, \pi]$ interval, where it is an invertible function.

The Alternation technique is applied to the transformed matrix $\hat{W}$, given a filled matrix of predictions: $\hat{W}_{imputed}$. Finally, the following transform should be applied:

$$W_{imputed} = \frac{\arccos\left(W_{imputed}\right)}{\pi}\left(M-m\right)+m. \tag{9}$$

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

The proposed adaptation of the Alternation focused on the prediction task in recommender systems is summarized in the following steps (differences with respect to the classical Alternation technique correspond to steps 1, 3, 5, 7, and 9, highlighted with italic type):

Adapted factorization algorithm: Given a matrix of ratings $W_{c\times p}$:

Apply the transformations summarized in (8), which give $\hat{W}$

Take a random $c \times r$ matrix A_0

Normalize the rows of A_0

Compute B_k from A_{k-1}:

$$B_k = \left(A_{k-1}^t A_{k-1}\right)^{-1}\left(A_{k-1}^t \hat{W}\right) \tag{10}$$

Normalize the columns of B_k

Compute A_k from B_k:

$$A_k = \hat{W} B_k^t \left(B_k B_k^t\right)^{-1} \tag{11}$$

Normalize the rows of A_k

Repeat steps 4–7 until the product $\hat{W}_{imputed} = A_k B_k$ converges to $\hat{W}$

Apply the transformations summarized in Equation (9)

Solution: The matrix $W_{imputed}$ contains the predicted ratings.

Rank Selection

As mentioned above, the prediction task in recommender systems can be reduced to find a low-rank matrix approximation of the data matrix. The particular dimension—that is, the rank value that gives the best predicted ratings—is not known *a priori* and cannot be directly computed when working with missing data.

Sarwar et al. point out that the goodness of the obtained predicted ratings depends on the selected rank value (r) of W [15]. In particular, they search for a rank value large enough to capture all the important information in the matrix, but small enough to avoid overfitting errors. Their experiments show that the overall performance of the SVD-based prediction algorithm significantly changes for a wide range of values of r [15]. The current paper presents results obtained by considering a reduced range of rank values. It will be seen further on that the predicted ratings obtained with [15] are similar in the studied range of ranks.

Brand proposes to use the rank value as a measure of the complexity of the model in the incremental SVD [3]. The objective is to maximize the probability of a correct generalization, while minimizing the complexity of the model. Furthermore, Brand claims that user ratings 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 $r = 4$ or $r = 5$, predicts the missing ratings 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 ratings, the SVD has small singular values. In general, in those cases, a smaller rank will give better predicted ratings.

It should be remarked that results obtained with the Alternation technique highly depend on the selected rank value. That is, if W has a rank r and the Alternation technique is used to approximate it by a rank r' matrix, being $r' > r$, 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 r' matrix, with $r' < r$, information is lost during the process. The missing entries are again wrongly filled in. Hence, the goodness of the predicted ratings, obtained with the Alternation, depends on the used rank value. Further on, it will be shown that using the proposed adaptation of the Alternation for recommender systems, the best predicted ratings are obtained, in general, for $r = 4$ or $r = 5$, as in [3]. In extreme cases, with a large amount of missing ratings, $r = 3$ or $r = 2$ is enough.

The present experimental results and comparisons were performed considering a range of different rank values. First, the error considering each rank value was computed. Then the rank with a minimum error was selected. Actually, a similar procedure was carried out in [3] and [15]. The case $r = 1$ is not considered, since it makes no sense to project the data onto a 1-dimensional subspace.

Data Sets

The data sets used in the experiments will now be introduced. Concretely, three different public data sets are considered.

The first data set is the one provided by the MovieLens recommender system (www.movielens.umn.edu), a Web-based research recommender system. One of its data sets consists of 100,000 ratings (discrete values from 1 to 5) by 943 users of 1,682 movies. A user-movie matrix W, formed by 943 rows and 1,682 columns, is constructed (see the obtained matrix in Figure 1, just as an example). Each entry W_{ij} represents the rating (from 1 to 5) of the ith-user on the jth-movie. This data set is also used in [3] and [15]. Since the goal is to study the goodness of the obtained predicted ratings, some entries are randomly removed and used to study their recovered values. These entries will hereinafter be referred to as the test data set. The rest of the entries used to recover the data will be referred to hereinafter as the training data set. Concretely, five different training and test data sets, split into 80,000 training and 20,000 test cases, are also given at www.movielens.umn.edu. The initial matrix of ratings has 93.69 percent of missing data, whereas using any training data set, a matrix of 94.95 percent of missing data is obtained.

The second data set was obtained from BookCrossing, a service where book lovers all around the world exchange books and share their experiences with others (www.bookcrossing.com). Ziegler et al. collect data from 278,858 members of BookCrossing, referring to 271,379 different books [19]. A total of 1,157,112 ratings are provided. These ratings take implicit (0) and explicit (from 1 up to 10) discrete values. The data set used in [19] is available at www.informatik.uni-freiburg.de/~cziegler. If all available data were used, the obtained matrix of ratings would be extremely sparse (concretely, it would have a percentage of missing ratings of about 99.9968%). The present experiments consider a smaller matrix with a higher density of known ratings. In particular, it is required that each user rates a minimum of books. At the same time, only books rated by a minimum of users are considered. Different minimum values will be considered in the experiments, as presented below.

Finally, the last data set used in the experiments was obtained from Jester, an on-line joke recommender system: http://eigentaste.berkeley.edu. The complete data set is publicly available at www.ieor.berkeley.edu/~goldberg/jester-data. Concretely, 4.1 million continuous ratings (from –10 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 by Goldberg et al. with a collaborative filtering algorithm based on principal component analysis (PCA) to obtain the predicted ratings in the Jester recommender system [8]. In particular, the authors propose to project the data into the eigenplane with the PCA. Then the projected data are clustered by using recursive rectangular clustering. When a new user asks for recom-

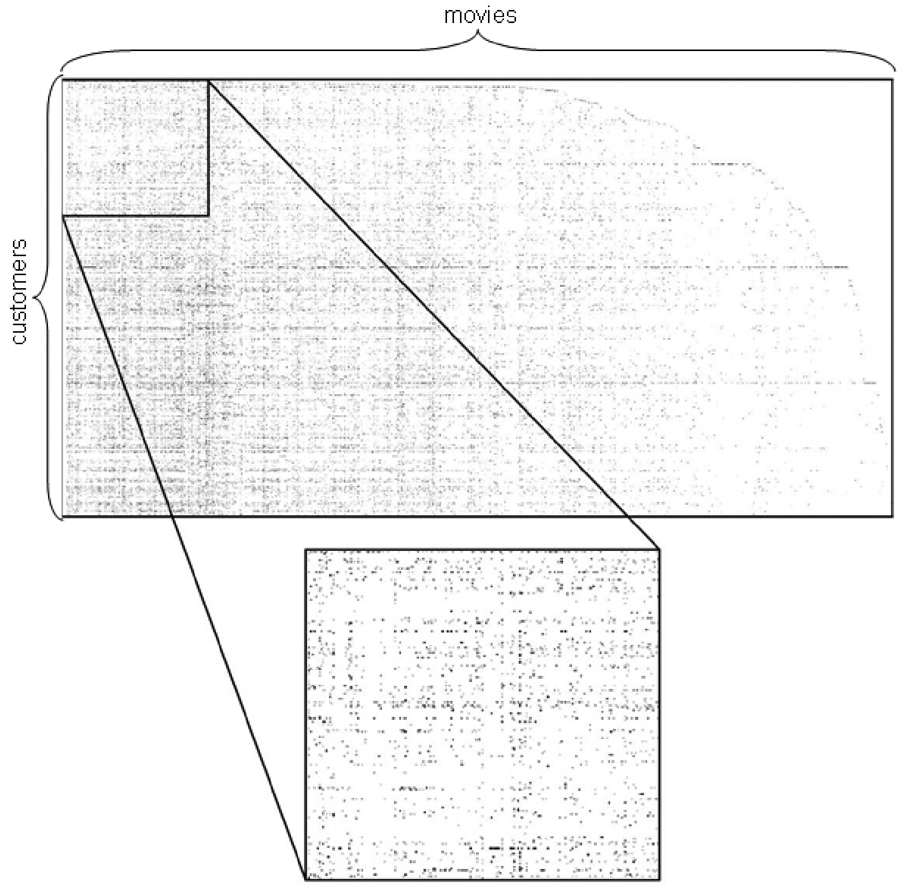

Figure 1. Matrix of ratings from the MovieLens data set, $W_{943 \times 1682}$, white entries correspond to unknown ratings; each row and column correspond to a customer and a movie respectively. The percentage of missing ratings is about 94.05%.<<IS IT POSSIBLE TO SUPPLY A CLEANER FIGURE?>>

mendations, the ratings the user gives are projected onto the eigenplane. Then the representative cluster of the user is found. Finally, the recommendations are computed from the ratings collected in the cluster.

Experimental Results

The predicted ratings obtained with the proposed adapted factorization approach will now be compared with the ones given by the SVD-based method presented in [15]. The latter method was selected based on the fact that it overcomes the performance of standard collaborative filtering techniques when dealing with very large and sparse matrices (see [15] for more details).

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 of the approaches proposed in the literature for the prediction task in recommender systems, and it is defined as follows:

$$MAE = \frac{1}{N}\sum_{i,j}\left|P_{ij} - W_{ij}\right|, \tag{12}$$

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 rating for the entry W_{ij}. The lower the MAE, the more accurate are the predicted ratings.

Experimental results from different data sets are presented separately due to their different natures. For instance, the percentage of available data and the size of the matrices are different in each data set. Another characteristic that should be taken into account is that the ratings can take discrete or continuous values.

It should be highlighted that the approach proposed in [15] is about 10 times more expensive than the proposed adapted factorization approach. It would be even more expensive if the classical SVD were used instead of the thin SVD.

MovieLens Data Set

In this data set, ratings take integer values from 1 to 5 and the percentage of missing data is about 94.05 percent. With so large an amount of missing data, experiments considering different percentages of missing data would not give significant conclusions. Five different training/test sets, provided at www.movielens.umn.edu, are used in the experiments, and the mean of all the training/test sets is given. Different 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 training/test data sets for each rank value is plotted in Figure 2. The minimum error (*MAE*) obtained with the proposed adapted factorization approach (denoted as ALT in the plots) is smaller than the one obtained with Sarwar's approach [15] (denoted SVD in the plots), as can be seen in Figure 2 (left). Concretely, with the adapted factorization 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. Results presented in [15] are a little bit different—the obtained *MAE* is about 0.7400 for the rank-14 case. This difference is possibly due to different training/test splits. Note that these *MAE* values mean that errors of about ±1 are obtained in the prediction task. As pointed out by Brand, this is very accurate, since the difference may reach ±2 values if the user is asked on different days [3].

Unfortunately, Brand's source code, which also works with the MovieLens data set, is not available [3]. Therefore it was necessary to rely on the reported results. The *MAE* obtained with [3] is 0.7914, which is slightly higher than the one obtained with the proposed adapted factorization technique. Again, this difference may be due to the training/test set used. Brand claims that the

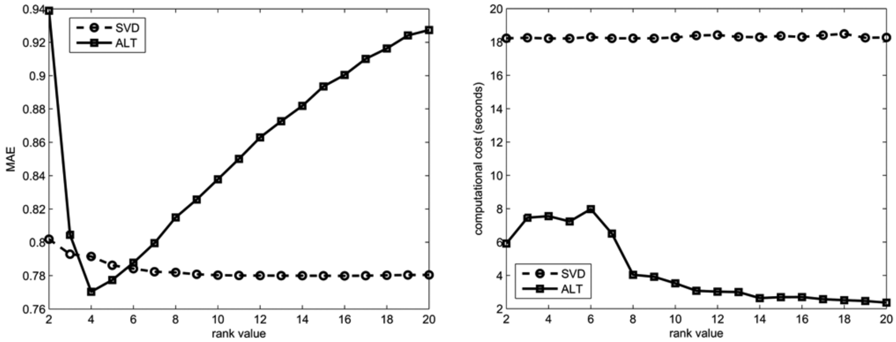

Figure 2. (left) MAE considering different rank values; (right) computational cost in seconds.

approach gives predicted ratings as least as good as those with [15], but for a smaller rank value.

From the present results, one may conclude that the data space can be reduced to a 4-dimensional subspace, as was also concluded by Brand, who also studied this data set [3]. Sarwar et al. found that a 14-dimensional subspace is needed in order to capture all the variance in the data [15]. In fact, in their approach, the *MAE* does not change much considering different dimensions, as is shown in Figure 2 (left). The computational cost for both approaches is depicted in Figure 2 (right). As can be seen, the proposed method is clearly faster for all tested rank values.

BookCrossing Data Set

The matrix of ratings obtained with the BookCrossing data set is even sparser than in the previous experiment. Concretely, it has a percentage of missing data of about 99.9968 percent. In order to obtain a matrix with more density of data, users who rated fewer than 20 books are discarded, and only books 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 has more than 20 and 200 known entries, respectively. In fact, with these two conditions, the number of considered users (rows) and books (columns) is 17,197 and 193, respectively, and the percentage of missing data is about 98.5094 percent. Again, five different training/test data sets are generated. Concretely, the test data set contains 0.4906 percent of known data, while the training data set contains only 1 percent of known data.

Figure 3 (left) shows the obtained *MAE* values, considering different rank values when a matrix with a 98.5094 percent of missing data is considered. As can be seen, the minimum error is obtained with the proposed adapted factorization approach, for $r = 2$. Concretely, $MAE = 3.5811$. The minimum error is also achieved for $r = 2$ with Sarwar's approach, but in this case $MAE = 3.7535$.

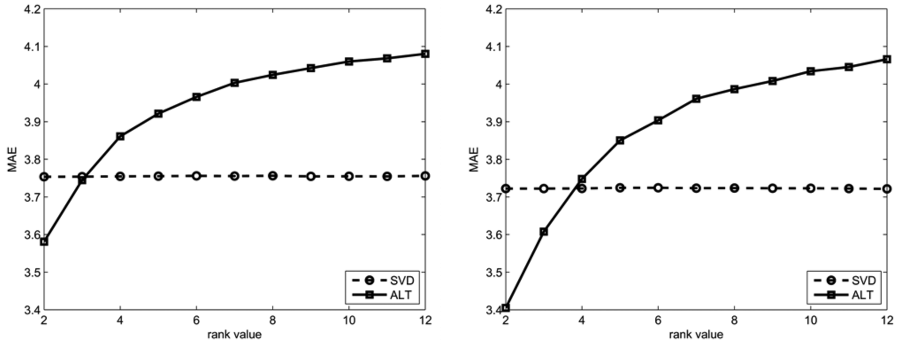

Figure 3. Obtained MAE considering different rank values; (left) A matrix of ratings with 98.5094% of missing data; (right) A matrix of ratings with 99.0465% of missing data.

A matrix of rank 2 predicts the missing ratings better than a matrix with a higher rank with only a 1 percent of known data.

Note that the error is larger in this case than in the experiments with the MovieLens data set. However, since the ratings lie in different ranges, another measurement should be defined. Goldberg et al. propose the use of the normalized mean absolute error (*NMAE*) to compare errors obtained from different data sets (7). The *NMAE* is defined as:

$$NMAE = \frac{MAE}{M - n}, \tag{13}$$

where *M* and *m* are the maximum and minimum values in the range of ratings, respectively. The results obtained with the proposed adapted factorization approach and both data sets can be compared using this new measure of error: in the case of the MovieLens data set, *MAE* = 0.7703, which gives an *NMAE* = 0.1926, while in the case of the BookCrossing data set, the obtained *MAE* is 3.5811, which gives an *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.

Different matrices of ratings are tested in a similar experiment requiring a smaller number of ratings per book. Books from the original matrix that were rated by fewer than 150 users are discarded. Hence, in this experiment, the number of considered users and books is 21,026 and 354, respectively, and the percentage of missing data is about 99.0465 percent. Thus, the obtained matrix has larger dimensions and more missing data. Concretely, the test data set contains 0.1535 percent of data, while the training data set contains only 0.8 percent of data. The obtained *MAE* values in this case are plotted in Figure 3 (right). Note that the minimum *MAE* is obtained with the proposed adapted factorization approach, for r = 2; its value is 3.4051. The minimum error with Sarwar's approach is obtained for r = 12, which corresponds to an *MAE* = 3.7215; similar results are obtained for any rank value.

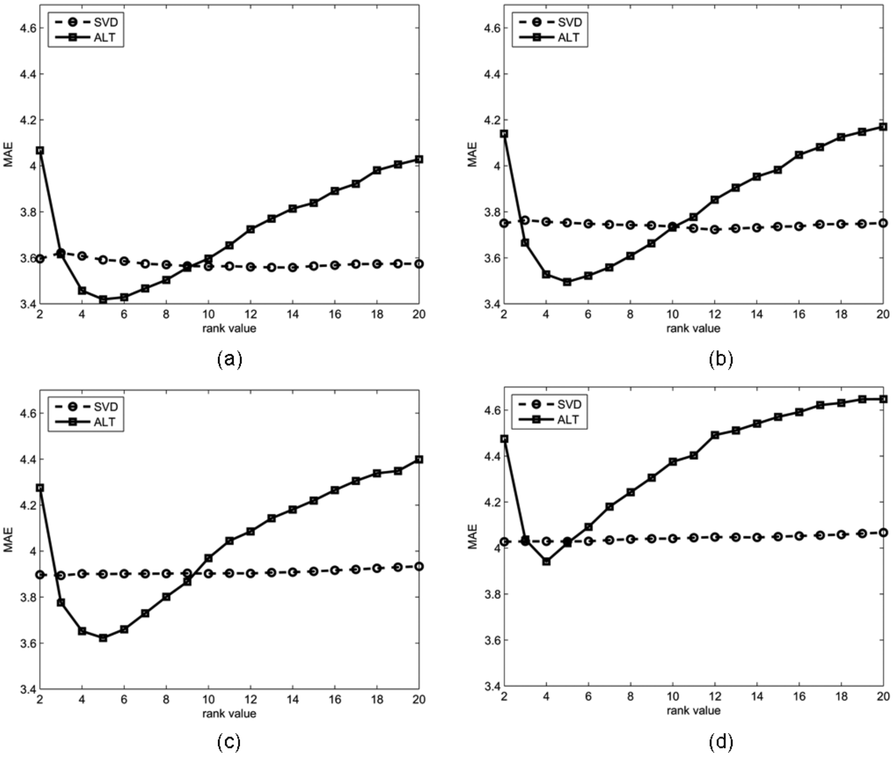

Figure 4. Obtained MAE values for different rank values and different percentages of missing ratings: (a) 60%; (b) 70%; (c) 80%; (d) 90%.

Jester Data Set

The ratings in this data set take continuous values from –10 up to 10. The obtained matrix contains ratings given by 73,421 users to 100 jokes, and the percentage of missing data is about 52 percent.

Different percentages of missing data are generated by randomly removing data (concretely, 60%, 70%, 80%, and 90%). The removed entries form the test data sets, while the rest of the data form the training data sets. Again, five different training/test data sets are considered in each experiment. Only 18,000 users from the total of 73,421 are randomly selected for the experiments, as in [8].

Figure 4 shows the error value (*MAE*) obtained considering different percentages of missing data and different rank values (from 2 to 20). As can be seen, in the case of Sarwar's approach, the obtained *MAE* is quite similar for any rank value. Concretely, the minimum *MAE* is achieved with $r = 14$ and $r = 12$, with 60 percent and 70 percent of missing data. Working with percentages of missing data of about 80 percent and 90 percent, the minimum *MAE* is obtained with $r = 3$ and $r = 2$, respectively. It can be appreciated in Figure 4 that in the case of the adapted factorization approach, the *MAE* value depends on the rank value. The minimum *MAE* is obtained for $r = 5$, while the percentage of missing data is below 90 percent; in the case of 90 percent of missing data, the minimum *MAE* is obtained with $r = 4$, as can be seen in Figure 4(d).

The obtained *MAE* values are similar to the ones presented by Goldberg et al., who studied the same data set. Unfortunately, no comparison can be performed with [8], since the authors do not provide accurate information about the percentage of missing data they considered, nor the used rows (they selected 18,000 rows randomly).

Although the obtained *MAE* seems to be higher than the one obtained with the MovieLens data set, the ratings lie in different ranges. The results are similar with both data sets if the *NMAE* proposed by Goldberg et al. is used [8]. In the case of the MovieLens data set, the values obtained with the proposed adapted factorization approach are as follows: *MAE* = 0.7704, which gives a *NMAE* = 0.1926, while in the case of the Jester data set, the *MAE* obtained with 90 percent of the missing data is 3.8959, which gives an *NMAE* = 0.1948. Therefore, the goodness of the predicted ratings is quite similar in both cases.

Summary

The preceding discussion shows that the proposed adapted factorization approach performs better than the SVD-based approach [15] in the three studied data sets. Additionally, it has a smaller computational cost. Collaborative filtering techniques would not give good results for data sets as large as the BookCrossing and the Jester data sets: On the one hand, the computational cost would be very high, and on the other, it would be very difficult to obtain good neighbors to compute the correlation coefficient, as pointed out in [15]. The performance of the collaborative filtering techniques in the MovieLens data set would depend on the percentages of the training and test data sets.

Conclusions

This paper presents an adapted factorization approach to predict missing ratings in recommender systems. The key point is that the prediction task in recommender systems can be reduced to finding the best low-rank approximation to the matrix of ratings, namely the data matrix. The proposed adapted factorization approach gives the best low-rank matrix approximation to the data matrix. At the same time, unknown ratings are filled in with the product of the factors recovered by the proposed technique. In particular, the Alternation technique is adapted to predict missing ratings in recommender systems. Although Alternation is not a new approach, it has not been used to tackle this problem, as far as can be determined. Concretely, the proposed approach uses the fact that the ratings take values in a known interval. Like the SVD-based approaches, it uses not only the correlated customers in the prediction task, but also the noncorrelated ones.

The proposed adapted factorization approach is compared with the SVD-based method presented in [15]. Three different public data sets, obtained from three different recommender systems, are studied. Experimental results show that the proposed approach performs better than the approach presented in [15] in respect both to error value and to computational cost.

Finally, it should be highlighted that good results are obtained with the proposed adapted factorization approach, even with percentages of missing data of more than 90 percent.

REFERENCES

<<need publishers>>

1. Berry, M.; Dumais, S.; and O'Brien, G. Using linear algebra for intelligent information retrieval. In **<<ed.?>>** *Society for Industrial and Applied Mathematics,* vol. 37. Charlotte: **<<publisher>>** 1995, pp. 573–595.

2. Billsus, D., and Pazzani, M. Learning collaborative information filters. In **<<ed.?>>** *15th International Conference on Machine Learning*. Madison **<<state?>>**: **<<publisher>>** 1998, pp. 46–54.

3. Brand, M. Fast online SVD revisions for lightweight recommender systems. In **<<ed.>>** *SIAM Conference on Data Mining*. Montreal: **<<publisher>>** 2003, pp. 37–46.

4. Brand, M. Incremental singular value decomposition of uncertain data with missing values. In **<<ed.>>** *European Conference on Computer Vision)*. Copenhagen: <<**publisher**>> 2002, pp. 707–720.

5. Buchanan, A., and Fitzgibbon, A. Damped Newton algorithms for matrix factorization with missing data. In **<<ed.>>** *IEEE Conference on Computer Vision and Pattern Recognition,* vol. 2. San Diego: **<<publisher>>** 2005, pp. 316–322.

6. Felfernig, A.; Friedrich, G.; Jannach, D.; and Zanker, M. An integrated environment for the development of knowledge-based recommender applications. *International Journal of Electronic Commerce, 11,* 2, (winter 2006–7), 11–34.

7. Guerreiro, R., and Aguiar, P. Estimation of rank deficient matrices from partial observations: Two-step iterative algorithms. In **<<what is?>>** *EMMCVPR,* 2003, pp. 450–466.

8. Goldberg, K.; Roeder, T.; Gupta, D.; and Perkins, C. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval, 4* (2001), 133–151.

9. Golub, G., and Van Loan, C. *Matrix Computations.* Baltimore: Johns Hopkins University Press, 1989.

10. Hartley, R., and Schaffalitzky, F. Powerfactorization**<<1 word correct?>>**: 3D reconstruction with missing or uncertain data. In **<<ed?>>** *Australian-Japan Advanced Workshop on Computer Vision.* Adelaide: **<<publisher>>**, 2003. <<**pp>**

11. Julià, C.; Sappa, A.D.; Lumbreras, F.; Serrat, J.; and López, A. An adapted alternation approach for recommender systems. In **<<ed.>>**, *IEEE International Conference on e-Business Engineering.* Xian: **<<publisher>>**, 2008, pp. 128–135.

12. McGinty, R., and Smyth, B. Adaptive selection: An analysis of critiquing and preference-based feedback in conversational recommender systems. *International Journal of Electronic Commerce, 11*, 2 (winter 2006–2007), 35–57.
13. Pryor, M. The effects of singular value decomposition on collaborative filtering. Computer Science Technical Report. Hanover, NH: Dartmouth College Computer Science, 1998.
14. Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, M.; and Riedl, J. GroupLens: An open architecture for collaborative filtering of netnews. In **<<ed.>>** *Conference on Computer Supported Cooperative Work*. Chapel Hill, NC: **<<publisher>>** 1994, pp. 175–186.
15. Sarwar, M.; Karypis, G.; Konstan, J.; and Riedl, J. Application of dimensionality reduction in recommender system—A case study. In **<<ed.>>** *Workshop on Web Mining for E-Commerce*. Boston: **<<publisher>>**, 2000, pp. 133–151.
16. Sarwar, B.; Karypis, G.; Konstan, J.; and Riedl, J. Incremental singular value decomposition algorithms for highly scalable recommender systems. In **<<ed.>>** *Fifth International Conference on Computer and Information Science*. **<<city? Publisher?>>,** 2002, pp. 27–28.
17. Tomasi, C., and Kanade, T. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision, 9* (1992), 137–154.
18. Wiberg, T. Computation of principal components when data is [*sic*] missing. In **<<ed.>>** *Second Symposium of Computational Statistics.* **<<place, publisher>>** 1976, pp. 229–326.
19. Ziegler, C.; McNee, S.; Konstan, J.; and Lausen, G. Improving recommendation lists through topic diversification. In **<<ed.>>** *International World Wide Web Conference.* Chiba **<<Japan?>>**: **<<publisher>>** 2005, pp. 22–32.

Appendix

Singular Value Decomposition (SVD)

This appendix summarizes the most useful properties of singular value decomposition (SVD). More details can be found in [9].

Theorem

Given an $m \times n$ matrix W, there exists a decomposition of this matrix of the form:

$$W_{m\times n} = U_{m\times m}\Sigma_{p\times p}V^{t}_{m\times n}, \tag{14}$$

where $U_{m\times m} = [u_1, \ldots, u_m]$ and $V_{n\times n} = [v_1, \ldots, v_n]$ are two orthogonal matrices (i.e., $U^tU = UU^t = I_m$ and $V^tV = VV^t = I_n$, where I_m is the $m \times m$ identity matrix) whose columns are the left and right singular vectors, respectively. $\Sigma_{p\times p}$ is a diagonal matrix that contains all the singular values σ_i of W, being $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$, $p = \min\{m, n\}$.

SVD and Rank of the Matrix

If the SVD of W is $W_{m\times n} = U_{m\times m}\Sigma_{p\times p}V^t_{m\times n}$, and r is defined by:

$$\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_p = 0 \tag{15}$$

then

$$rank(W) = r$$
$$null(W) = span\{v_{r+1}, \ldots, v_n\}$$
$$ran(W) = span\{u_1, \ldots, u_r\},$$

and W can be defined as:

$$W = \sum_{i=1}^{r} \sigma_i u_i v_i^t. \tag{16}$$

Hence, the r columns of U corresponding to the nonzero singular values span the column space, while the r columns of V span the row space of the matrix W.

SVD and Matrix Norms

The norm-2 and Frobenius norm have connections to the SVD. Concretely:

$$\|W\|_2 = \sigma_1 \tag{17}$$

$$\|W\|_F^2 = \sigma_1^2 + \ldots + \sigma_p^2, \quad p = \min\{m,n\}. \tag{18}$$

The Thin SVD

Given a matrix $W_{m\times n}$, being $m \geq n$, the thin SVD consists in computing only the n column vectors of U corresponding to the row vectors of V^t. That is:

$$W_{m\times n} = U_{m\times n}\Sigma_{m\times n}V^t_{m\times n}. \tag{19}$$

The remaining columns of U are not computed. This commonly used SVD is significantly faster than the full SVD.

CARME JULIÀ (carme.julia@urv.cat) received a Ph.D. in computer science in the Computer Vision Center at the Universitat Autónoma de Barcelona. Since September 2008 she has been a member of the Intelligent Robotics and Computer Vision Group at the Universitat Rovira i Virgili, Tarragona, Spain. Her research interests are focused on structure from motion through factorization, factorizing matrices with missing and noisy data, and photometric stereo through factorization.

ANGEL D. SAPPA (asappa@cvc.uab.es) received a Ph.D. in industrial engineering from the Polytechnic University of Catalonia, Barcelona, Spain, in 1999. In 2003, after holding research positions in France, the UK, and Greece, he joined the Computer Vision Center, Barcelona, Spain, where he is currently a member of the Advanced Driver Assistance Systems Group. His research interests span a broad spectrum within 2D and 3D image processing. His current research focuses on stereo image processing and analysis, 3D modeling, and model-based segmentation.

FELIPE LUMBRERAS (felipe@cvc.uab.es) received a Ph.D. in computer science from the Universitat Autónoma de Barcelona in 2001. He is currently an associate professor in the Computer Science Department and a member of the Centre de Visió per Computador (CVC). His research interests include wavelet and texture analysis, 3D reconstruction, and computer vision for automotive applications.

JOAN SERRAT (joans@cvc.uab.es) is associate lecturer at the computer science department of the Universitat Autòònoma de Barcelona, and member of the Computer Vision Center. His areas of interest are structure from motion, 3D reconstruction, and computer vision applied to driving assistance systems. He has headed several machine vision projects for local industries.

ANTONIO LÓPEZ (antonio@cvc.uab.es) received a Ph.D. from the Universitat Autónoma de Barcelona (UAB) in 2000. Since 1992, he has been lecturing in the Computer Science Department, UAB, where he is currently an associate professor. In 1996, he participated in the founding of the Computer Vision Center, UAB, where he has held different institutional responsibilities and is at present responsible for the research group on advanced driver-assistance systems by computer vision. He has coauthored more than 50 papers, all in the field of computer vision.