Matrix and Tensor Factorization Techniques applied to Recommender Systems: a Survey

Giuseppe Ricci, Marco de Gemmis, Giovanni Semeraro
Department of Computer Science
University of Study "A. Moro"
Bari, Italy
giuseppe.ricci@uniba.it, degemmis@di.uniba.it, semeraro@di.uniba.it

Abstract—**On internet today, an overabundance of information can be accessed, making it difficult for users to process and evaluate options and make appropriate choices. This phenomenon is known as Information Overload.**

Over time, various methods of information filtering have been introduced in order to assist users in choosing what may be of interest to them.

Recommender Systems (RS) are a technique for filtering information and play an important role in e-commerce, advertising, e-mail filtering etc.

Therefore, RS are an answer, though partial, to the problem of Information Overload. Algorithms behind the recommendation techniques need to be continuously updated because of a constant increase in both the quantity of information and the availability of modes of access to that information, which define the different contexts of information use.

The research of more effective and more efficient methods than those currently known in literature is also stimulated by the interests of industrial research in this field, as demonstrated by the Netflix Prize Competition. The Company, which gives its name to the award, has invested one million dollars in acknowledgement of the best collaborative filtering algorithm that improves the accuracy of its own adopted RS, evidently in the belief that the RS can provide a competitive advantage. The mathematical techniques discussed in this article seem to be, at present, the most feasible way to calculate more efficient and accurate recommendations.

The main contribution of this paper is a survey about matrix and tensor factorization techniques adopted in the literature of RS. In particular, the discussion focuses on recent applications of High Order Singular Value Decomposition (HOSVD) in the area of information filtering and retrieval (Section IV). Finally, we suggest the application of PARAFAC (PARAllel FACtor) to multidimensional data for the computation of context-aware recommendations.

Recommender systems; information overload; matrix factorization; SVD; tensor factorization; HOSV; PARAFAC

I. INTRODUCTION

Recommender Systems (RS) are a well-established solution to the problem of Information Overload since they implement information filtering methods providing users with information that matches their needs in a consistent and timely manner [22]. With the ever-increasing information available in digital archives, the challenge of implementing personalized filters has become the challenge of designing algorithms able to manage huge amounts of data for the elicitation of user needs and preferences. In recent years, Matrix Factorization (MF) techniques have proved to be a quite promising solution to the problem of designing efficient filtering algorithms in the "Big Data" Era. They played a key role in the solution proposed by the team that won the Netflix Prize contest, the open competition for the best collaborative filtering algorithm to predict user ratings for films, based on previous ratings. The main contribution of this work is a critical analysis of matrix factorization and tensor factorization techniques, with the aim of pointing out their strengths and weaknesses from the specific point of view of recommender systems.

Matrix factorization techniques form part of Collaborative Filtering (CF) and, particularly, of latent factor models [12]. These models assume that similarity between users and items is induced by some factors hidden in the data. These models attempt to explain the ratings by charactering both items and users with the objective of disclosing the latent features deducted from ratings. In the same way a person can naturally define the characteristics of a movie (such as the genre, key players, length of the film, etc.), methods based on latent factors infer this characteristic data without exactly knowing each feature. In this case, latent factor models build a matrix of users and items (movies) and each element is associated with a vector of characteristics. As an example it can be assumed that the rating which a user assigns to a movie depends on few implicit factors such as the user's preferences about various movie genres. In MF, users and items are simultaneously represented by vectors of features derived from ratings expressed by users for the items seen/tried. A high correspondence between user and item factors leads to a recommendation.

RS data are collected in a matrix called **user-item matrix**: rows are referred to users and columns to items; the intersection between one row and one column is the rating expressed by the user, which corresponds to the line in relation to the item on the column. An example of a user-item matrix is shown in Figure 1. In this matrix the user Alice has expressed a rating for Matrix and X-Men, the user Bob has seen and voted Avatar and Matrix and the user Charlie has rated only Avatar. The "?" are the missing values that are movies not yet seen and for which the user has not given any rating. The

rating given by the users, and reported in the user-item matrix, is an example of explicit feedback. MF techniques primarily use this data type. However, they may also make use of implicit feedback, instead of explicit feedback [14].
This feedback indirectly reflects the user's preferences by observing user behavior through examining purchase history, the browser history, search patterns, or even the movements of the mouse and clicked links [12].

	Avatar	Matrix	X − Men
Alice	?	4	2
Bob	3	2	?
Charlie	5	?	?

Figure 1. User-item example matrix.

II. Matrix Factorization

MF aims to factorize the matrix of ratings into two matrices. Let's suppose we have a set of user *U* and a set of items *D*. *R* is matrix of ratings assigned by users to items. What you want to discover is a number of latent features: *k*. The task of the MF is to find two matrices *P* and *Q* such that their product approximates *R*:

$$R \approx P \times Q^T. \quad (1)$$

In this way, each row of *P* represents the strength of the association between user and features. Similarly, each column of *Q* represents the strength of the association between an item and features.
In order to understand how the recommendation is made, it is appropriate to introduce the user profile vector p_i and q_j the vector profile item. They represent the projection of the item and the user in a common space so that the product $p_i \cdot q^T$ is a similarity score which approximates the rating.
If you indicate with p_i the *i*-th row of *P* and q_j with the *j*-th row of *Q*, the scalar product of the vectors $p_i \cdot q^T$, deriving from equation (1) translated in terms of rows and columns, captures the interaction between the user *i* and the item *j*, for example, the overall interest of user with respect to the characteristics of the product. This scalar product approximates the rating of user *i* with respect to the item *j* and if the rating is denoted with r_{ij}, as an element of matrix *R*, the estimate of the rating is given by:

$$\widehat{r_{ij}} = p_i \cdot q_j^T. \quad (2)$$

The most difficult task is to collect the values to be inserted into the two vectors p_i and q_j; once the system has mapped these vectors, recommendations can be calculated using equation (2).

MF enters into action in estimating the vectors p_i and q_j. A factorization used in the literature is **Singular Value Decomposition** (SVD), which was introduced for the first time by Simon Funk [6] in the NetFlix Prize [2], [13]. The SVD applied to the RS has the objective of reducing the dimensionality, i.e. the rank, of the user-item matrix so as to capture latent relationships between users and items [23]. SVD decomposes the ratings matrix *R* in the product of three matrices in the following way:

$$R = P \cdot \Sigma \cdot Q^T \quad (3)$$

where *P* and *Q* are orthonormal matrices and Σ is a diagonal matrix.

An important definition is the singular value of a matrix. If *A* is a *nxm* matrix, we define **singular values** of *A*, $\sigma_{1,\ldots,}\ \sigma_r$, the root square of non-zero eigenvalues of the matrix AA^T and A^TA:

$$\sigma_i = \sqrt{\lambda_i}$$

the corresponding eigenvectors of the matrix AA^T and A^TA are called **singular vectors** of *A*. A property at the base of SVD are: given *A* a real matrix *nxm* with rank *r*, *A* can be factored into the product of a matrix *P mxr* with orthonormal columns, a diagonal matrix Σ, $\Sigma = diag(\sigma_{1,\ldots,}\ \sigma_r)$ that on the diagonal has the singular values of *A*, for a matrix Q^T *rxn* with orthonormal rows:

$$A = P\ \Sigma\ Q^T.$$

However, this technique has some disadvantages. Dataset from RS are very large and very sparse; take the Netflix Prize dataset as an example: it consists of 480,189 users, 17,770 movies and only ~ 100 million of ratings. The user-item matrix (480,189 x 17,770) is very sparse since the known elements are only 1% of all the elements in the matrix. Therefore, there are obvious problems of scalability: SVD-based algorithms require a large memory to store data to be processed, as well as very long processing time. We must also consider that the SVD can only be applied to full matrices and not to sparse matrices. To overcome the problem of missing values, different approaches have been proposed in literature. A very common approach uses the value 0 to replace missing values. In literature several SVD-based algorithms have been introduced that try to improve the scalability such as the SVD++, Asymmetric-SVD [11], IncrementalSVD [3], etc. The recommendation algorithms based on these improvements of SVD allow precise and reliable recommendations with a good level of scalability.

III. Multidimensional Approach

There are different MF techniques, other than SVD, such as the algorithm described in [8], which produce effective recommendations and has a good scalability.

One of the main limitations of matrix factorization techniques is that they take into account only the standard profile of users and items. This does not allow to integrate further information such as context. For example, if a user watches a movie at home with his children, he will certainly choose a movie whose genre is suitable for families, such as cartoon or animation. Indeed, when the same user goes to the cinema with friends or colleagues, he will prefer horror or war movies. Contextual information (the place where the user see the movie, the device used to display the movie, the company, etc.) cannot be managed with user-item matrices.

To embed this additional information in the data [1], [20] which can range from the context to the temporal dimension, we must make use of tensors, which is nothing other than a multidimensional array [10]. Kolda and Bader provide an overview of higher-order tensor decompositions and their applications. Decompositions of higher-order tensors (i.e., *N*-way arrays with $N \geq 3$) have applications in psychometrics, chemometrics, signal processing, numerical linear algebra, computer vision, numerical analysis, data mining, neuroscience, graph analysis, etc. Two particular tensor decompositions can be considered to be higher-order extensions of the matrix singular value decomposition: CANDECOMP/PARAFAC (CP) decomposes a tensor as a sum of rank-one tensors, and the High Order Singular Value Decomposition (HOSVD) is a higher-order form of principal component analysis.

The techniques that generalize the MF factorization can also be applied to tensors [1]. An application of Tensor Factorization (TF) is discussed by Symeonidis et al. [21]. The model for three-mode factor analysis is discussed in terms of newer applications of mathematical processes including a type of matrix process termed the Kronecker product and the definition of combination variables. Three methods of analysis to a type of extension of principal components analysis are discussed. Methods are applicable to analysis of data collected for a large sample of individuals.

Our aim is the factorization of a tensor and then the use of this factorization to formulate recommendations. In literature the most frequently used technique for the factorization of a tensor is HOSVD [17], [5] which is a generalization of the SVD for matrices. This technique decomposes initial tensor in *N* matrices (where *N* is the size of the tensor) and a tensor of size smaller than the original one.

IV. TENSOR FACTORIZATION: HOSVD

In Baltrunas et al. [9], the factorization of a tensor is applied to stored data for users, movies, users' ratings for movies seen and contextual information. A third-order tensor is constructed and HOSVD is applied to factorize the tensor.

A loss function and a regularization procedure are also introduced to improve the quality of the recommendations. In analogy with the approach of MF to measure the discrepancy between two matrices, a suitable loss function is introduced and the authors suggest some possible expressions for this function. A simple minimization of a loss function can lead to the problem of overfitting; a regularization term is introduced which is based on the standard l_2 norm of the factors. Missing values are treated as 0, not as a non-acceptance of the movie by the user, but as a special value that represents a missing value.

The algorithm obtained, called *Multiverse Recommendation TF*, turns out to be a compact algorithm which takes into account contextual information and provides quite effective recommendations. The authors used three datasets to test their algorithm: Yahoo! Webscope, the dataset used by Adomavicius et al. in [1], and the dataset used by Hideki et al [16]. The results obtained by Multiverse Recommendation algorithm, compared with standard MF non-contextual methods, show an improvement ranging from 5% to 30% on the 3 datasets. The algorithm also improves the traditional algorithms of context-aware recommendation from 2.5% to as much as 12%.

In contrast, the disadvantage of this approach turns out to be the high computational cost because it quite heavily applies a minimization algorithm of a function (Stochastic Gradient Descend, SGD).

Another application of the HOSVD for the TF is described in [17], where is the authors applied it in the context of social tagging [7]. A system for social tagging allows users to enter metadata in the form of keyword(s) in order to classify and categorize items.

Collaborative tagging systems suggest a tag to users based on the tags that other users have used for the same items, in order to develop a consensus about which tags best describe an item. Data acquired in this system are stored in a third-order tensor that will shape the three types of entities that are usually found in a social tagging system: users, items and tags. Tensor is factored with HOSVD with the aim of discovering the latent factors that bind the associations user-item, user-tag and tag-item. Tags will be suggested to a user according to any identified associations. This algorithm is very expensive from a computational point of view because it applies SVD to the matrices obtained by the operation of unfolding of the tensor (the unfolding matriciziting the tensor): the output of this operation are 9 matrix and the core tensor.

In [19] HOSVD is applied to the factorization of a tensor coming from a system of personalized web search, in order to discover the hidden relationships between objects typical of internet search: users, queries, web pages. The pages that are suggested to the user are based on the discovery of these associations.

Data related to user, query and web pages are collected in a third-order tensor that is decomposed with the technique of HOSVD. In a retrieval system the user enters a query q in the search box, the search engine returns a list of URLs with a description of target web pages, the user clicks on relevant page p. With the usage of the system, data related to the query and pages clicked by the user are collected in the form of triplets (u, q, p). This allows the building of a tensor of third order in which each dimension corresponds, respectively, to the data of users, queries, and clicked pages and any element of this tensor measures the preference of the pair (*u*, *q*) for the page *p*. The algorithm used to factor the tensor has been named *CubeSVD*. The output of the HOSVD is the reconstructed tensor which measures the associations between users, queries and web pages. It is the product of the core tensor and the three matrices obtained from HOSVD. The core tensor governs the interaction between users, queries and web pages. One element of the reconstructed tensor can be represented by a quadruple (*u*, *q*, *p*, *w*) where *w* measures the popularity of page *p* as a result of query *q* made by the user *u*. Pages recommended to a user are based on the weight associated with each pair (*u*, *q*).

V. Conclusion and Current Work

In this paper we discussed matrix and tensor factorization techniques most widely adopted in the area of recommender systems. We observed that the commonly adopted dimensions users and items are not sufficient to capture all the factors that influence users preferences and choices. In an attempt to overcome this limitation, we reviewed the literature of tensor factorization, by focusing on HOSVD as factorization technique. In the literature about TF, only the HOSVD has been applied to recommender systems so far. This technique, however, has some issues that could be addressed. An alternative solution may be to use a tensor to collect data from users, items and rating and put them together as multidimensional information such as: the context in which a movie has been seen, the users' profession, etc. in order to overcome the limitation of two-dimensionality of data represented by matrices.

Our aim is to explore the use of other techniques for factorization which can firstly, be more effective and able to make more accurate recommendations taking into account one or more dimensions, like context, secondly, can provide the diversified user recommendations compiled from multidimensional information and lastly, can be more efficient from a computational point of view.

There are other techniques in TF literature such as PARAFAC/CANDECOMP, Tucker decomposition, DEDICOM decomposition, PARATUCKER decomposition [18], etc. We want to test the application of PARAFAC (PARAllel FACtor) [4], [15] to multidimensional data of recommendation systems. We focus on one tensor factorizations that captures multi-linear structure, PARAFAC/ CANDECOMP.

If we indicate with $X = [x_{ijk}] \in \mathbb{R}^{I \times J \times K}$ a tridimensional tensor, PARAFAC's decomposition of *X* is:

$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} + \varepsilon_{ijk}$$

where a_{ip}, b_{jq}, c_{kr} are elements of the factor matrix, ε_{ijk} is the error. PARAFAC does not collapse the data but rather retains its natural three-dimensional structure. In the presence of missing data, PARAFAC can be formulated as a weighted least squares problem that models only the known entries.

In particular, we want to understand whether the application of this technique is as performing as HOSVD, if it requires computational resources less than or equal to HOSVD, if it can reveal more latent information than HOSVD and if the recommendation algorithm based on this technique, allows for recommendations that are as reliable, as efficient and not discountable. We therefore, want to reach two main objectives: to evaluate the accuracy that can be obtained with PARAFAC in the task of context-aware recommendation and to surpass mere accuracy: the recommendations provided should not only suggest the obvious but should be able to surprise the user.

References

[1] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. Inf. Syst 23, 1 (January 2005), 103-145.

[2] J. Bennet, S. Lanning, "The Netflix Prize", KDD Cup and Workshop, 2007.

[3] Matthew Brand. Fast online svd revisions for lightweight recommender systems. In SIAM International Conference on Data Mining, 2003.

[4] J. Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. Psychometrika, 35:283-319, 1970.

[5] Lieven De Lathauwer, Bart De Moor, Joos Vandewalle, A multilinear singular value decomposition, SIAM J. Matrix Anal. Appl, 2000, Vol. 21, pag.1253-1278.

[6] S. Funk, "Netflix Update: Try This At Home, http://sifter.org/~simon/journal/20061211.html, 2006.

[7] Halpin, Harry and Robu, Valentin and Shepard, Hana. The Dynamics and Semantics of Collaborative Tagging. Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW'06), 2006.

[8] Carme Julià, Angel D. Sappa, Felipe Lumbreras, Joan Serrat, and Antonio Lòpez. Predicting missing ratings in recommender systems: Adapted factorization approach. International Journal of Electronic Commerce, 2009.

[9] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context aware collaborative filtering. In Proceedings of the fourth ACM conference on Recommender systems, RecSys '10, pages 79{86, New York, NY, USA, 2010. ACM.

[10] Tamara G. Kolda and Brett W. Bader, Tensor Decompositions and Applications. SIAM Review, 51(3), 455-500 (2009).. Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In ACM Int. Conference on Knowledge Discovery and Data Mining (KDD'08), 2008.

[11] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In ACM Int. Conference on Knowledge Discovery and Data Mining (KDD'08), 2008.

[12] Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. IEEE Computer, 42(8):3037, 2009.

[13] Yehuda Koren, The BellKor Solution to the Netflix Grand Prize, 2009.

[14] Yehuda Koren, Yifan Hu, Chris Volinsky. Collaborative Filtering for Implicit Feedback Datasets. Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), pages: 263-272, 2008.

[15] Harshman R.A. Foundations of the parafac procedure: models and conditions for an “explanatory” multi-mode factor analysis. pages 16, 1-84, 1970.

[16] Chihiro Ono, Yasuhiro Takishima, Yoichi Motomura, and Hideki Asoh. 2009. Context-Aware Preference Model Based on a Study of Difference between Real and Supposed Situation Data. In Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization: formerly UM and AH (UMAP '09), pages 102-113, Berlin, Heidelberg, 2009, Springer-Verlag.

[17] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '09, pages 727-736, New York, NY, USA, 2009. ACM.

[18] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In Proceedings of the third ACM international conference on Web search and data mining, WSDM '10, pages 81-90, New York, NY, USA, 2010. ACM.

[19] Jian-Tao Sun, Hua-Jun Zeng, Huan Liu, Yuchang Lu, and Zheng Chen. Cube-svd: a novel approach to personalized web search. In Proceedings of the 14th international conference on World Wide Web, pages 382-390, New York, NY, USA, 2005. ACM.

[20] Cosimo Palmisano, Alexander Tuzhilin, Michele Gorgoglione. Using Context to Improve Predictive Modeling of Customers in Personalization Applications, Transactions on Knowledge and Data Engineering, 20, pages: 1535-1549, 2008.

[21] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In Proceedings of the 2008 ACM conference on Recommender systems, RecSys '08, pages 43-50, New York, NY, USA, 2008. ACM.

[22] Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor. Recommender Systems Handbook, Springer, 2011.

[23] Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl, Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems, Fifth International Conference on Computer and Information Science, pag. 27-28, 2002.