Mathematical Methods of Tensor Factorization Applied to Recommender Systems

Giuseppe Ricci, Marco de Gemmis, and Giovanni Semeraro

Department of Computer Science University of Bari Aldo Moro, Italy
giuseppe.ricci@uniba.it, {marco.degemmis,giovanni.semeraro}@uniba.it
http://www.di.uniba.it/~swap/index.php

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 their interest. Recommender Systems (RS) [14] are techniques for information filtering which 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. Recommendation algorithms need to be continuously updated because of a constant increase in both the quantity of information and ways 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 Contest, the open competition for the best algorithm to predict user ratings for films, based on previous ratings. The contest showed the superiority of mathematical methods that discover latent factors which drives user-item similarity, with respect to classical collaborative filtering algorithms. With the ever-increasing information available in digital archives and textual databases, 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 techniques have proved to be a quite promising solution to the problem of designing efficient filtering algorithms in the *Big Data* Era. The main contribution of this paper is an analysis of these methods, which focuses on tensor factorization techniques, as well as the definition of a method for tensor factorization suitable for recommender systems.

Keywords: Recommender Systems, Matrix Factorization, Tensor Factorization, PARAFAC/CANDECOMP.

1 Matrix Factorization

Recommender systems guide users in a personalized way to interesting or useful objects in a large space of possible options, by providing a list of suggested

B. Catania et al. (eds.), *New Trends in Databases and Information Systems*, Advances in Intelligent Systems and Computing 241,
DOI: 10.1007/978-3-319-01863-8_40,

items that fits their interests. For example, Netflix, a provider of on-demand Internet streaming video and flat rate DVD-by-mail in the United States, adopts a recommendation algorithm to predict user interests for films, based on feedback provided by users on previously watched items. The most widely adopted recommendation techniques in literature are content-based and collaborative filtering ones.

Matrix Factorization (MF) techniques fall in the class of collaborative filtering (CF) methods and, particularly, in the class of latent factor models [10], which 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 genre, key players, duration, 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. MF techniques represent users and items by vectors of features derived from ratings given by users for the items seen or 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 given by the user. Missing values correspond to movies not rated by the user.

Let U be the set of users, D the set of items, R the matrix of ratings. MF aims to factorize R into two matrices P and Q such that their product approximates R: $R \approx P \times Q^T$. Each row of P represents the strength of the association between user and k latent features. Similarly, each column of Q represents the strength of the association between an item and the latent features. Let p_i be the i-th row of P and q_j the j-th row of Q. They are the user profile vector and the item profile vector respectively, which represent the projection of user i and item j in a common space of k latent features. The scalar product $p_i \cdot q_j^T$ approximates the rating r_{ij} of user i for item j: $\hat{r}_{ij} = p_i \cdot q_j^T$. Once these vectors are discovered, recommendations are calculated using the expression of $\hat{r}_{ij}$. A factorization used in the literature is *Singular Value Decomposition* (SVD), introduced by Simon Funk in the NetFlix Prize [5], [3], has the objective of reducing the dimensionality, i.e. the rank, of the user-item matrix, in order to capture latent relationships between users and items [15]. Different SVD algorithms were used in RS literature: in [15], the authors uses a small SVD obtained retaining only $k \ll r$ singular values by discarding other entries; in [11], the authors propose an algorithm to perform SVD on large matrices, by focusing the study on parameters that affect the convergence speed; in [9], Koren presents an approach oriented on factor models which projected users and items in the same latent space where some measures for comparison are defined. He propose several versions of SVD with the objective of having better recommendations as well as good scalability.

2 Tensor Factorization

The main limitation of MF 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 choose a movie whose genre is suitable for families. Indeed, in another context (friends or colleagues), the same user might prefer other kind of movies. Contextual information (the place where the user see the movie, the device, the company, etc.) cannot be managed with simple user-item matrices. *Tensors*, which can be seen as higher-dimensional arrays of numbers [8], might be exploited in order to include additional contextual information in the recommendation process [2]. In standard multivariate data analysis, data are arranged in a two-dimensional structure, but for a wide variety of domains, more appropriate structures are required for taking into account more dimensions. The techniques that generalize the MF factorization can also be applied to tensors. Two particular tensor decompositions [8] can be considered to be higher-order extensions of matrix singular value decomposition:

– **PARallel FACtor analysis** or CANonical DECOMPosition (PARAFAC/ CANDECOMP) [4], [6], which decomposes a tensor as a sum of rank-one tensors;
– **High Order Singular Value Decomposition** (HOSVD) [12], which is a higher-order form of Principal Component Analysis (PCA).

In RS literature, the most frequently used technique for Tensor Factorization (TF) is HOSVD, which is a generalization of the SVD for matrices. This technique decomposes the initial tensor in N matrices (where N is the size of the tensor) and a tensor whose size is smaller than the original one.

HOSVD is used in [7], where the factorization of a tensor is applied to manage data for users, movies, user ratings and contextual information such as age, day of the week, companion. A third-order tensor is constructed and HOSVD is applied to factorize it into three matrices and one core tensor. Recommendation score for a single user i, item j and context k is computed by using these matrices and tensor. Another application of HOSVD for TF is described in [13], in the context of social tagging to predict a personalized list of tags for a user. Users' data, items and tags are stored in a third-order tensor which is factored by HOSVD, with the aim of discovering latent factors which bind the associations user-item, user-tag and tag-item. In [17], 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. Data related to user, query and web pages are collected in a third-order tensor that is decomposed with the technique of HOSVD.

The major advantage of HOSVD is the ability of simultaneously taking into account more dimensions. This allows for a better data modeling than standard SVD, since dimensionality reduction can be performed not only in one dimension but also separately for each dimension. But HOSVD is not an optimal tensor decomposition, in the sense of least squares data fitting: the computation of HOSVD needs standard SVD computation only and has not the truncation property of the SVD, where truncating the first n singular values allows to find the best n-rank approximation of a given matrix. Despite this, the approximation obtained is not far from the optimal one and can be computed much faster. Since HOSVD cannot deal with missing values, they are treated as 0.

PARAFAC (PARallel FACtor analysis) is a decomposition method, which can be seen as a generalization of bilinear PCA. The PARAFAC model was independently proposed by Harshman [6] and by Carroll & Chang [4] who named the model *CANDECOMP* (CANonical DECOMPosition). A PARAFAC model of a three-dimensional array is given by three loading matrices A, B, and C with typical elements a_{if}, b_{jf}, and c_{kf}. The PARAFAC model is defined by the following structural model:

$$\hat{x}_{ijk} = \sum_{f=1}^{F} a_{if} b_{jf} c_{kf}. \tag{1}$$

where F is the number of rank-one components. PARAFAC is an alternative to HOSVD. One of the advantages of the PARAFAC is its simplicity which allows to use an analytical expression for solving the decomposition problem and to achieve linear scalability. Another advantage is linear computation time compared to HOSVD. PARAFAC does not collapse data, but it retains its natural three-dimensional structure. Despite PARAFAC mode's lack of ortogonalithy, Kruskal [8] showed that components are unique, up to permutation and scaling, under mild conditions.

In [16], PARAFAC is exploited for the computation of top-N context-aware recommendations of mobile applications. A tensor of three dimensions (users, items and context types) is factorized with PARAFAC. These dimensions are associated with the three factor matrices and used to calculate user preference for item i under context type k. In [1], PARAFAC is applied focusing on missing data. The authors developed a scalable algorithm called *CP-WOPT* (CP Weighted OPTimization), which uses first-order optimization to solve the weighted least squares objective function. Using extensive numerical experiments on simulated data sets, Acar et al. showed that CP-WOPT can successfully factor tensors with noise and up to 70% missing data. Moreover, CP-WOPT is significantly faster and accurate than the best published method in the literature [18].

3 CP-WOPT Adaptation: Preliminary Experiments

Our idea is to adapt CP-WOPT and to introduce it in the RS field, where the problem of missing values is very relevant, since the algorithm is suitable for very sparse user-items matrices. The adaptation allows the computation of a weighted factorization that models only know values, rather to simply employ 0 values for missing data. The main goal is to consider contextual information about users and to apply the weighted PARAFAC decomposition to achieve precise recommendations. In order to reach this goal, we made a preliminary user study with 7 real users who were asked to rate a fixed number of movies (11) in the Movielens 100k dataset on the basis of three contextual factors: if they like to see the movie (i) at home or cinema; (ii) with friends or with partner; (iii) with or without family. Ratings range from 1 to 5 in the sense that, for each contextual comparison:

– rating 1 and 2 express a strong and a modest preference, respectively, for the first term;
– rating 3 expresses neutrality;
– rating 4 and 5 express a modest and a strong preference, respectively, for the second term.

Results are measured in terms of accuracy (*acc*), i.e. the percentage of known values correctly reconstructed and coverage (*cov*), i.e. the percentage of non-zero values returned. Under the assumption of 10^5 maximum iterations, we obtained $acc = 94.4\%$ and $cov = 91.7\%$. Although coming from a limited study, the values of these measures suggest we are moving in a correct direction and seem to promise encouraging results when applying the algorithm to more complex context-aware recommendation scenarios. Moreover, the experiment showed that it is possible to express, through the n-dimensional factorization, not only the recommendations for the single user, but also more specific suggestions about the consumption of an item. For instance, *American Pie* is tipically watched at home, with friends and without family, while *Titanic* is preferably watched at cinema, with partner or family.

We performed also an in vitro preliminary experiment to test the adapted version of CP-WOPT on a subset of Movielens 100k dataset. We gave as input a tensor of dimensions 100 users, 150 movies, 21 occupations (the contextual factor) and we measured, besides *acc* and *cov*, also the classic Mean Average Error (MAE) and Root Mean Square Error (RMSE), in order to compare the results with those known in literature. The algorithm achieved: $acc = 92.09\%$, $cov = 99.96\%$, $MAE = 0.60$ and $RMSE = 0.93$, which are in line with results reported in literature.

In future we want to extend the evaluation of our version of CP-WOPT on tensor having high dimensionality extracted form Movielens dataset. In particular, we will investigate methods to assess whether contextual factors (occupation, company) influences the users' preferences, by using data mining techniques such as clustering. We plan also to test our approach in other domains such as news recommendation.

References

[1] Acar, E., Dunlavy, D.M., Kolda, T.G., Mørup, M.: Scalable tensor factorizations with missing data. In: SDM 2010: Proceedings of the 2010 SIAM International Conference on Data Mining, Philadelphia, pp. 701–712. SIAM (April 2010)

[2] Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. Inf. Syst. 23(1), 103–145 (2005)
[3] Bennett, J., Lanning, S.: The netflix prize. In: Proceedings of the KDD Cup Workshop 2007, pp. 3–6. ACM, New York (2007)
[4] Carroll, J., Chang, J.-J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. Psychometrika 35(3), 283–319 (1970)
[5] Funk, S.: Netflix update: Try this at home (2006), `http://sifter.org/~simon/journal/20061211.html`
[6] Harshman, R.A.: Foundations of the PARAFAC Procedure: Models and Conditions for an "explanatory" Multi-modal Factor Analysis. In: Working papers in phonetics, vol. 1(16), University of California at Los Angeles (1970)
[7] Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N.: Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: Proceedings of the fourth ACM Conference on Recommender Systems, RecSys 2010, pp. 79–86. ACM, New York (2010)
[8] Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. 51(3), 455–500 (2009)
[9] Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 426–434. ACM, New York (2008)
[10] Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer 42(8), 30–37 (2009)
[11] Kurucz, M., Benczúr, A.A., Torma, B.: Methods for large scale svd with missing values. In: KDDCup 2007 (2007)
[12] De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. 21, 1253–1278 (2000)
[13] Rendle, S., Marinho, L.B., Nanopoulos, A., Schmidt-Thieme, L.: Learning optimal ranking with tensor factorization for tag recommendation. In: KDD, pp. 727–736 (2009)
[14] Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.): Recommender Systems Handbook. Springer (2011)
[15] Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Fifth International Conference on Computer and Information Science, pp. 27–28 (2002)
[16] Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Hanjalic, A., Oliver, N.: Tfmap: optimizing map for top-n context-aware recommendation. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2012, pp. 155–164. ACM, New York (2012)
[17] Sun, J.-T., Zeng, H.-J., Liu, H., Lu, Y., Chen, Z.: Cubesvd: a novel approach to personalized web search. In: Proceedings of the 14th International Conference on World Wide Web, WWW 2005, pp. 382–390. ACM, New York (2005)
[18] Tomasi, G., Bro, R.: Parafac and missing values. Chemometrics and Intelligent Laboratory Systems 75(2), 163 (2005)