Matrix and Tensor Decomposition in Recommender Systems

Panagiotis Symeonidis
Department of Informatics
Aristotle University
Thessaloniki, 54124, Greece
psymeon@gmail.com

ABSTRACT

This turorial offers a rich blend of theory and practice regarding dimensionality reduction methods, to address the information overload problem in recommender systems. This problem affects our everyday experience while searching for knowledge on a topic. Naive Collaborative Filtering cannot deal with challenging issues such as scalability, noise, and sparsity. We can deal with all the aforementioned challenges by applying matrix and tensor decomposition methods. These methods have been proven to be the most accurate (i.e., Netflix prize) and efficient for handling big data. For each method (SVD, SVD++, timeSVD++, HOSVD, CUR, etc.) we will provide a detailed theoretical mathematical background and a step-by-step analysis, by using an integrated toy example, which runs throughout all parts of the tutorial, helping the audience to understand clearly the differences among factorisation methods.

1. INTRODUCTION

Representing data in lower dimensional spaces has been extensively used in many disciplines such as natural language and image processing, data mining and information retrieval. Recommender systems deal with challenging issues such as scalability, noise, and sparsity and thus, matrix and tensor factorization techniques appear as an interesting tool to be exploited [2, 4]. That is, we can deal with all the aforementioned challenges by applying matrix and tensor decomposition methods (also known as factorization methods).

The rest of this paper is organized as follows. Section 2 provides a detailed outline of the tutorial for matrix decomposition techniques. Section 3 provides a detailed outline of the tutorial for tensor decomposition. Finally, Section 4 concludes this paper.

2. MATRIX DECOMPOSITION

Matrix Factorization denotes a process, where a matrix is factorized into a product of matrices. Its importance relies on the exploitation of the latent associations that exist in data among the participating entities (e.g. between users and items). In a trivial form, the matrix factorization method uses two matrices, which hold the information of correlation between the user-feature and item-feature factors, respectively.

Figure 1 shows an example of the latent factors, which could be revealed after performing matrix decomposition. As shown, the X'X axis divides both people and movies according sex (e.g., male or female). When a movie is closer to the female part of X'X axis, it means that this movie is most popular in women rather than in men. The Y'Y axis divides people and movies as "war-like" and "romantic". A "war-like" viewer is assumed to prefer movies showing blood and deaths. In contrast, a "romantic" viewer chooses movies that present love and passion.

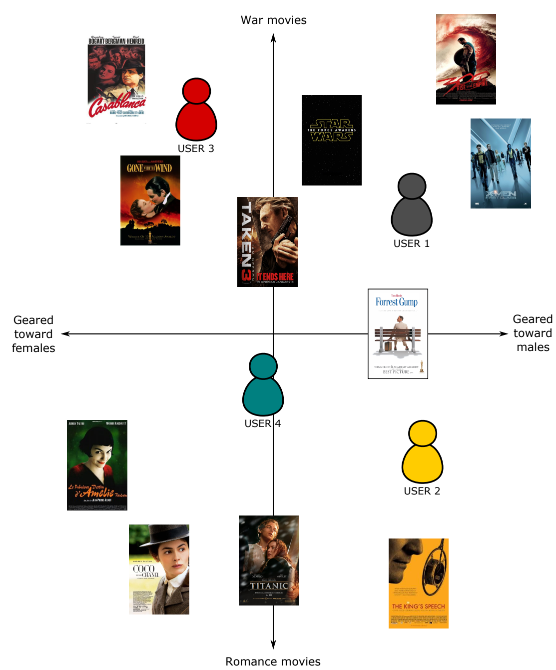

Figure 1: Visual representation of users and movies using two axes — male versus female and war-like versus romantic.

RecSys '16 September 15-19, 2016, Boston , MA, USA

ACM ISBN 978-1-4503-4035-9/16/09.
DOI: http://dx.doi.org/10.1145/2959100.2959195

To predict a user's rating over a movie, we can compute the dot product of the movie's and user's [x,y] coordinates on the graph. In addition, Figure 1 shows where movies and users might fall on the basic two dimensions. For example, we would expect *User 3* to love *"Casablanca"*, to hate *"The King's Speech"*, and to rate *"Amelie"* about average. Note that some movies (i.e. *"Taken 3"*) and users (i.e. *User 4*) would be characterised as fairly neutral on these two dimensions.

In this tutorial, we provide the related work of basic matrix decomposition methods. The first method that we discuss is known as Eigenvalue Decomposition, which decomposes the initial matrix into a canonical form. A second method is the Non-Negative Matrix Factorization (NMF), which factorizes the initial matrix into two smaller matrices with the constraint that each element of the factorized matrices should be non-negative. A third method is the Probabilistic Matrix Factorization (PMF), which scales well to large datasets. PMF mathod performs well on very sparse and imbalanced datasets using spherical Gaussian priors. The last but one method is Probabilistic Latent Semantic Analysis (PLSA) which is based on a mixture decomposition derived from a latent class model. The last method is CUR Decomposition, which confronts the problem of density in the factorized matrices (a problem that is faced on SVD method). Moreover, we describe Singular Value Decomposition (SVD) and UV-Decomposition in details. We minimise an objective function, which captures the error between the predicted and the real value of a user's rating. Moreover, an additional constraint of friendship is added in the objective function to leverage the quality of recommendations [1].

Finally, we study the performance of the described SVD and UV-decomposition algorithms, against an improved version of the original item-based CF algorithm combined with SVD.

3. TENSOR DECOMPOSITION

Because of the ternary relational nature of data in many cases (e.g., Social Tagging Systems (STSs), Location-based social network (LBSNs) etc.), many recommendation algorithms originally designed to operate on matrices cannot be applied. Higher order problems put forward new challenges and opportunities for recommender systems. For example, the ternary relation of STSs can be represented as a third-order tensor $\mathcal{A} = (a_{u,i,t}) \in \mathbb{R}^{|U|\times|I|\times|T|}$. Symeonidis et al. [3], for example, proposed to interpret the user assignment of a tag on an item, as a binary tensor where 1 indicates observed tag assignments and 0 missing values (see Figure 2):

$$a_{u,i,t} := \begin{cases} 1, & \text{if user } u \text{ assign on item } i \text{ tag } t \\ 0, & \text{otherwise} \end{cases}$$

Tensor factorization techniques can be employed in order to exploit the underlying latent semantic structure in tensor $\mathcal{A}$. The basic idea is to transform the recommendation problem as a third-order tensor completion problem, by trying to predict the non-observed entries in $\mathcal{A}$.

In this tutorial, we provide the related work on tensor decomposition methods. The first method that is discussed is the Tucker Decomposition method (TD), which is the underlying tensor factorization model of HOSVD. TD decomposes a tensor into a set of matrices and one small core tensor. The second one is PARAFAC method (PARAllel FACtor analysis), which is the same as TD method with the restriction that core tensor should be diagonal. The third method is the PITF method (Pairwise Interaction Tensor Factorization), which is a special case of TD method with linear runtime both for learning and prediction. The last method that is analyzed is the Low-order Tensor Decomposition (LOTD) method.

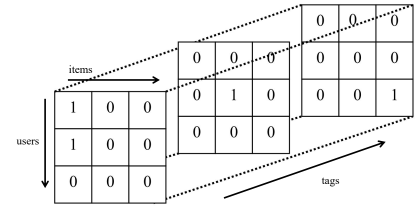

Figure 2: Tensor representation of a STS where positive feedback is interpreted as 1 (i.e., $a_{utr} := 1$) and the rest as 0 (i.e., $a_{utr} := 0$).

The main factorization method that will be presented in this tutorial is Higher Order SVD (HOSVD), which is an extended version of the SVD method. In particular, we will present a step-by-step implementation of HOSVD in a toy example. Then, we will present how we can update HOSVD when a new user is registered in our recommender system. We will also discuss how HOSVD can be combined with other methods for leveraging the quality of recommendations. Finally, we will provide experimental results of tensor decomposition methods on real datasets in STSs. Moreover, we will discuss about the metrics that we will use (i.e., Precision, Recall, RMSE, etc.). Our goal is to present the main factors that influence the effectiveness of algorithms.

4. CONCLUSIONS

In this tutorial, we discuss the advantages and limitations of each matrix and tensor decomposition algorithm for recommender systems, and provide future research directions.

References

[1] Yehuda Koren. 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 '08, pages 426–434, New York, NY, USA, 2008. ACM.

[2] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[3] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 43–50. ACM, 2008.

[4] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Matrix factorization and neighbor based algorithms for the netflix prize problem. RecSys '08, pages 267–274, New York, NY, USA, 2008. ACM.