Incorporating Side Information in Tensor Completion

Hemank Lamba*, Vaishnavh Nagarajan*, Kijung Shin*, Naji Shajarisales*
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15213, USA
{hlamba,vaishnavh,kijungs}@cs.cmu.edu, nshajari@andrew.cmu.edu

ABSTRACT

Matrix and tensor completion techniques have proven useful in many applications such as recommender systems, image/video restoration, and web search. We explore the idea of using external information in completing missing values in tensors. In this work, we present a framework that employs side information as kernel matrices for tensor factorization. We apply our framework to problems of recommender systems and video restoration and show that our framework effectively deals with the cold-start problem.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining

Keywords

Tensor Factorization, Kernels, Recommender Systems

1. INTRODUCTION AND RELATED WORK

Tensor Factorization (TF) has been widely used for completing missing entries in a tensor [1]. One problem of previous methods [2] is that they assume independence between entities in the same mode. However, there can be meaningful correlation between these entities, e.g., friends share common taste of movies and consecutive rows of pixels in an image are generally similar. These similarities, obtained from side information, can be exploited to improve the quality of tensor completion. In this work, we propose a tensor factorization method that incorporates these similarities through kernel matrices. This extends previous work in kernelized matrix factorizaton [3] for higher dimensions.

In this work, we apply our method to recommender systems where the entries of a tensor represent ratings and the modes correspond to users, movies and *context*. The side information used here is social networks among users and similarities among movies. We also use our tensor completion method to restore video, represented as a 3-way tensor. We show that our algorithm works better than baseline methods when it has to make predictions for unseen entities. This situation is referred to as the *cold-start* problem. One example is the case of new users, who do not have any reviewing history. Similarly, for video restoration, we consider the case where entire rows, columns, or frames are missing.

Table 1: Table of Symbols

Notation	Definition
$\mathcal{R}$	$N_1 \times N_2 \times ... \times N_Q$ data tensor
$r_{i_1 i_2 ... i_Q}$	$(i_1, i_2, ..., i_Q)$-th entry of $\mathcal{R}$
N_q	Length of q-th mode of $\mathcal{R}$
D	Dimension of the latent factors.
$\mathbf{U}^{(q)}$	$N_q \times D$ factor matrix for q-th mode of $\mathcal{R}$
$\mathbf{U}^{(q)}_{i_q,:}$	i_q-th row of $\mathbf{U}^{(q)}$
$\mathbf{U}^{(q)}_{:,d}$	d-th column of $\mathbf{U}^{(q)}$
$\mathbf{K}^{(q)}$	$N_q \times N_q$ covariance matrix for q-th mode of $\mathcal{R}$
$\mathbf{S}^{(q)}$	$N_q \times N_q$ inverse matrix of $\mathbf{K}^{(q)}$

2. METHODOLOGY

In a previous model [2] for PARAFAC decomposition of tensors with missing entries, each row of factor matrices is generated independently from a given distribution. A serious problem of this model is that the rows of factor matrices cannot be estimated when we are dealing with the cold-start problem.

In our approach, we first generate factor matrices by sampling each column of factor matrices from a Gaussian process. The mean is zero and the correlation across entries in the column is defined by a kernel matrix. Side information is employed to design this kernel matrix. On generating entries of a tensor, we consider the outer product of these factor matrices. Each entry is sampled from a Gaussian distribution where mean is the corresponding entry of the outer product and variance is given.

2.1 A Kernelized Probabilistic Model for TF

Under our modeling scheme, the prior distribution of each column of factor matrices is a Gaussian process as follows:

$$1 \leq \forall q \leq Q, \quad p(\mathbf{U}^{(q)}|\mathbf{K}^{(q)}) = \prod_{d=1}^{D} \mathcal{GP}(\mathbf{U}_{:,d}|\mathbf{0}, \mathbf{K}^{(q)}) \qquad (1)$$

Here $\mathbf{K}^{(q)}$ is the kernel matrix for q-th mode (see Section 2.2 for details about this kernel matrices). Also we assume the likelihood of $\mathcal{R}$ given the factor matrices to be normally distributed with a constant error parameter σ^2 as follows:

$$p(\mathcal{R}|\mathbf{U}^{(1)}, ..., \mathbf{U}^{(Q)}, \sigma^2) = \prod_{I \in \mathcal{I}} \left[\mathcal{N}\left(\mathcal{R}_I | \sum_{d=1}^{D} \prod_{q=1}^{Q} \mathbf{U}^{(q)}_{I_q d}, \sigma^2 \right) \right]^{\delta_I} .$$

*These authors contributed equally to this work.

Copyright is held by the author/owner(s).

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2889371.

where $\mathcal{I}$ is the set of all indices for $\mathfrak{R}$, i.e.

$$\mathcal{I} = \{I = (I_1, I_2, ..., I_Q) | 1 \le \forall I_q \le \ N_Q, 1 \le \forall j \le k\},$$

and δ_I is the indicator function which attains 1 if $\mathfrak{R}_I$ is observed and 0 otherwise. This gives rise to the following log posterior:

$$\log p(\mathbf{U}^{(1)}, ..., \mathbf{U}^{(Q)} | \mathfrak{R}, \sigma^2, \mathbf{K}^{(1)}, ..., \mathbf{K}^{(Q)}) = - \frac{1}{2\sigma^2} \sum_{I \in \mathcal{I}} \delta_I \left(\mathfrak{R}_I - \sum_{d=1}^{D} \prod_{q=1}^{Q} \mathbf{U}^{(q)}_{I_q d} \right)^2 - A \log \sigma^2 - \frac{1}{2} \sum_{q=1}^{Q} \sum_{d=1}^{D} \left(\mathbf{U}^{(q)}_{:,d} \right)^T \mathbf{S}^{(q)} \mathbf{U}^{(q)}_{:,d} - \frac{D}{2} (\sum_{q=1}^{Q} \log |\mathbf{K}^{(q)}|) + C$$

where A is the total number of non-missing entries in $\mathfrak{R}$, $|\mathbf{K}^{(q)}|$ is the determinant of $\mathbf{K}^{(q)}$, and C is a constant. For our purposes, we infer the parameters of this model based on maximum aposteriori estimation. We use stochastic gradient descent to compute the MAP estimate where the partial derivative is as follows:

$$\frac{\partial \log p}{\partial \mathbf{U}^{(r)}_{i_r,d}} = - \frac{2}{\sigma^2} \left(\mathfrak{R}_I - \sum_{d=1}^{D} \prod_{q=1}^{Q} \mathbf{U}^{(q)}_{i_q d} \right) \prod_{\substack{q=1 \\ q \ne r}}^{Q} \mathbf{U}^{(q)}_{i_q d} + \frac{1}{\tilde{M}^{(r)}_{i_r}} \left[\sum_{j_r=1}^{N_r} \mathbf{S}^{(r)}_{i_r j_r} \mathbf{U}^{(r)}_{j_r,d} + \mathbf{S}^{(r)}_{i_r i_r} \mathbf{U}^{(r)}_{i_r,d} \right]$$

where $\mathbf{S}^{(r)}$ is the inverse of $\mathbf{K}^{(r)}$ and $\tilde{M}^{(r)}_{i_r}$ is the number of observed entries in the i_r-th fiber of the rth mode of $\mathfrak{R}$.

2.2 Deriving Kernels

In this section, we design kernel matrices (i.e., $\mathbf{K}^{(q)}$ in (1)) for different modes of tensors using side information.

Kernel for Users: Given a social network among users, the similarity between two users is defined as the commute time (i.e., the expected number of steps of a random walker on the network to commute between two users). The kernel matrix is $L^{\dagger}$ where L is the Laplacian matrix and † represents the MoorePenrose pseudoinverse.

Kernel for Movies: We first build a graph among movies where each movie is adjacent to the 20 most similar movies in terms of the cosine similarity between the 200 most frequent words in their plots. As in users, similarity between two movies is defined as the commute time on this graph.

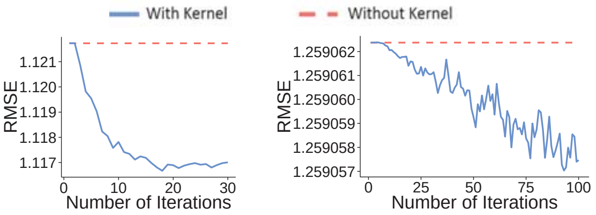
Kernel for Time: To encode smooth change across time. we use RBF kernel for similarity between two time points.

Kernel for Video: We first build a graph among pixels where each pixel is adjacent to k_{row} closest pixels in the same column and frame, k_{col} closest pixels in the same row and frame, and k_{frame} closest pixels in the same row and column. Then, we compute commute time on this graph.

3. EXPERIMENTS

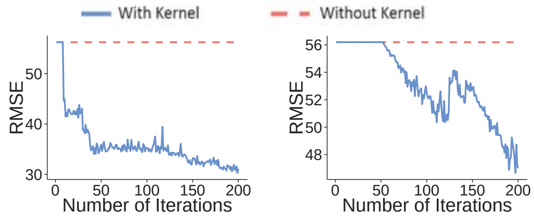
We apply our methods to two different domains (recommender systems and video restoration) and show that our method effectively deals with the cold-start problem.

We model ratings in the Flixster dataset[1] as a 3-way tensor where modes correspond to users, movies, and time (year-months). In terms of accuracy over the entire test set, our proposed method was comparable with a state-of-the-art method [2]. However, only our method could deal with the cold-start problem and learned the preference in new context, while the competitor could not, as seen in Figure 1.



(a) New Context (b) New Context, User, Movie

Figure 1: Accuracy of our method (with kernels) and its competitor [2] (without kernel) in Flixster Dataset in the case of the cold-start problem. Our method successfully learned the preference in new contexts (1(a)), while the competitor could not. Our method even learned the preference of a new user about a new movie in a new context (1(b)).



(a) Missing row (column) (b) Missing frame

Figure 2: Accuracy of our method (with kernels) and its competitor [2] (without kernel) in video restoration. Our methods successfully restored the video, while the competitor could not due to the cold start problem.

For the video restoration task, we model *xylophone.mp4*, provided by MATLAB, as a 4-way tensor where modes correspond to rows, columns, frames, and RGB. After removing 20% of rows, columns, and frames to model the cold start problem, we restored these missing pixels using tensor factorization methods. As seen in Figure 2, only our method successfully dealt with the cold-start problem.

4. CONCLUSIONS

In this work, we propose a framework that employs side information as kernel matrices for tensor factorization. This extends a previous work on kernelized matrix factorization [3] for high dimensional data. We apply our method to recommendation and video restoration with our designed kernels and show that our method successfully deals with the cold-start problem. Future work includes applying our method to other tensor decomposition methods, such as Tucker decomposition, and new domains, such as web search.

References

[1] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys*, 2010.

[2] K. Shin and U. Kang. Distributed methods for high-dimensional and large-scale tensor factorization. In *ICDM*, 2014.

[3] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *SDM*, 2012.

[1] http://www.cs.ubc.ca/~jamalim/datasets/