Multilinear Filtering Based on a Hierarchical Structure of Covariance Matrices

TFML2015, 16-21 February 2015, Będlewo, Poland

Andrzej Szwabe, Paweł Misiorek, Michał Ciesielczyk

Poznan University of Technology
Faculty of Electrical Engineering
Institute of Control and Information Engineering

February 17th, 2015

Outline

- Research background
- Research motivation
- Basic concepts
- Method description
- Experimentation
- Conclusions

Research Background: NCN Project

- Title: **Information-theoretic abductive reasoning for context-based recommendation** (DEC-2011/01/D/ST6/06788)
- Type: Sonata
- Leader: **Andrzej Szwabe, PhD**
- Main goals:
  - develop the tensor-based data representation model
  - develop the quantitative computational model of abductive reasoning based on a tensor-based data representation model
  - propose the quantitative reasoning plausibility measure
- Research team:
  - Pawel Misiorek
  - Michal Ciesielczyk
  - Czeslaw Jedrzejek
  - Przemyslaw Walkowiak
  - Michal Blinkiewicz
  - Tadeusz Janasiewicz

Research Motivation (1/2)

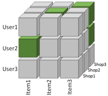
- Tensor-based data modeling and processing
  - Kroonenberg P. M.: *Three-mode principal component analysis: Theory and applications*, DSWO press, 1983.
  - Nickel M., Tresp V.:An Analysis of Tensor Models for Learning on Structured Data. In H. e. Blockeel, editor, *Machine Learning and Knowledge Discovery in Databases*, LNCS 8189, pages 272–287, 2013.
  - Sandin F., Emruli B., Sahlgren M.: Incremental dimension reduction of tensors with random index. *CoRR*, pages 240–56, 2011.
  - De Lathauwer L., De Moor B., Vandewalle, J.: A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl*, 21:1253–1278, 2000.

Research Motivation (2/2)

- Multilinear filtering methods
  - Baldassarre L., Rosasco L., Barla A., Verri A.: Multi-output learning via spectral filtering. *Machine learning*, 87(3):259–301, 2012.
  - Lu H., Plataniotis K.N., Venetsanopoulos A.N.: MPCA: Multilinear principal component analysis of tensor objects, *IEEE Trans. Neural Netw.*, 19(1):18–39, 2008.
- Related research
  - Szwabe A., Misiorek P., Walkowiak P.: Multi-Relational Learning for Recommendation of Matches between Semantic Structures, in: Grana, M. (Eds.), *Knowledge Engineering, Machine Learning and Lattice Computing with Applications*, LNAI 7828, pp. 98-107, 2013.
  - Szwabe A., Misiorek P., Janasiewicz T., Walkowiak P.: Holistic Entropy Reduction for Collaborative Filtering, Foundations of Computing and Decision Sciences. 39(3):209–229, 2014.

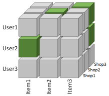
Research Motivation: Basic Goals and Assumptions

- Main goal is to provide a new model of multilinear filtering based on a hierarchical structures of covariance matrices.
- Additional assumptions
  - to use a tensor data representation, i.e. multidimensional (multi-mode) arrays, as a highly-expressive and suitable for heterogeneous relational data representation,
  - to address the standard collaborative filtering scenario of 'cold start' personalized recommendation (of very sparse input data),
  - to apply the tensor data compression enabling efficient storing and processing,
  - to evaluate the proposed approach in terms of recommendation quality (using AUROC measure) and computation time.



Research Motivation: The Tensor-based Approach

- Advantages of tensor-based data representation
  - ability to model relational data given as n-tuples,
  - a simple logical interpretation based on the tensor product,
  - an enabler for ‘seing’ the modeled data from a higher number of perspectives.
- The paper novel idea is to enhance the standard tensor-based modeling by additional analysis of generalized data representation obtained using the procedure of ‘neglecting’ (aggregating) the variance of one or more tensor modes (i.e., flattening the tensor).



Basic Concepts: Tensor

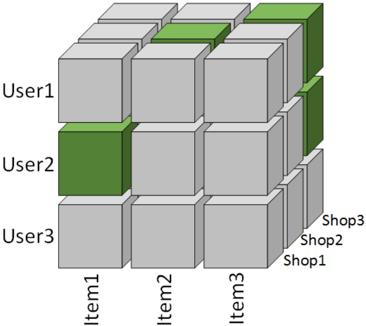


Figure: An example of the 3-mode tensor.

In the presented research a concept of the tensor may be seen (in a simplified way) as a multidimensional array used to model the relational dataset given as a set of n-tuples. The tensor dimensions we will refer to as tensor **modes**.

Basic Concepts: Tensor Unfolding and Fibres

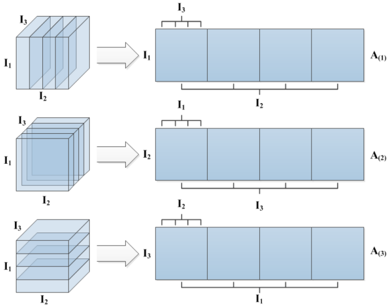


Figure: Unfoldings of the 3-mode tensor.

The tensor **unfolding** for a given mode is the Kronecker-product--based matrix representation of the tensor for which the rows are indexed this mode. The columns of the unfolding correspond to so-called tensor **fibres**, which are the vectors being the most specified samples describing the elements of the given mode.

Basic Concepts: Tensor Flattenings

- The operation of **flattening** the tensor $T = [t_{i_1,\ldots,i_n}]_{n_1\times\cdots\times n_n}$ over mode i leads to the new tensor T', such that:

$$T' = [t'_{i_1,i_2,\ldots,i_{i-1},i_{i+1},\ldots,i_n}]_{n_1\times n_2\times\ldots\times n_{i-1}\times n_{i+1}\times\ldots\times n_n},$$

  where

$$t'_{i_1,i_2,\ldots,i_{i-1},i_{i+1},\ldots,i_n} = \sum_{1\leq j\leq n_i} t_{i_1,i_2,\ldots,i_{i-1},i_{i+1},\ldots,i_n}.$$

- The flattening operation may be seen as aggregating the tensor entries across the mode being flattened (i.e., aggregating the tensor fibres to one number).
- The flattening operation enables to see new dependencies in the input data – different dependencies in data may be observed depending on the choice of attributes used to model tensor modes.

Basic Concepts: Tensor Overall Centring

- The standard centring operation is provided by the subtraction of the mean of values in cells of a given tensor slice and may be seen as a centring from the perspective of a given mode.
- The **overall centring** is done by consecutive centring of tensor fibres in each mode, i.e., for a given mode all fibres are centred and then this procedure is repeated for the next mode and so on.
- The overall centring operation for a given n-mode tensor T may be formulated in terms of the inclusion-exclusion principle.
- The overall centring is the centring operation which leads to the minimum Frobenius norm of the centered tensor.

Basic Concepts: Tensor-to-Tensor Transformation

- The **tensor-to-tensor transformation** is done as follows:

$$\widetilde{T} = T \times_1 U^{(1)} \times_2 \cdots \times_n U^{(n)},$$

where $T \times_i U^{(i)}$ is a tensor by matrix multiplication transforming tensor fibres of i-th mode of tensor T into new fibres in the corresponding mode of output tensor $\widetilde{T}$ in such a way that the entries of a new fibre are just inner products of the old fibre and columns of matrix $U^{(i)}$. The entries of the result tensor of each tensor-to-tensor transformation may be calculated as follows:

$$\widetilde{t}_{j_1,\dots,j_n} = \sum_{i_1 \in I^{(1)}} \dots \sum_{i_n \in I^{(n)}} t_{i_1,\dots,i_n} u^{(1)}_{j_1,i_1} \dots u^{(n)}_{j_n,i_n}.$$

The tensor-to-tensor transformation does not depend on the order of per mode operations.

CMF Method - Overview

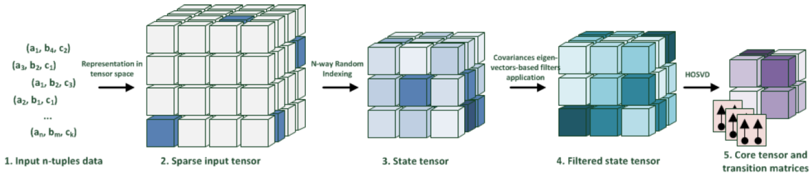


Figure: Overview of data processing in the CMF method.

CMF Method, Step 1: The n-tuple-based Input Data Representation

We assume that the input data are given as weighted n-tuples, where n is a number of attributes defining each event:

$$\Gamma = (n, \mathcal{V}^{(1)}, \ldots, \mathcal{V}^{(n)}, \Lambda, \psi),$$

where $\mathcal{V}^{(i)}$, $(i = 1, \ldots, n)$, is a set of values which may be used as the i-th element of an n-tuple, Λ is a set of n-tuples of the form $(v^{(1)}, \ldots, v^{(n)})$ where $v^{(i)} \in \mathcal{V}^{(i)}$, and $\psi : \mathcal{V}^{(1)} \times \cdots \times \mathcal{V}^{(n)} \to \mathbb{R}$ is a function used to assign the weight. To model the set of n-tuples as a tensor we define the tensor space $\mathcal{T} = \mathcal{I}^{(1)} \otimes \cdots \otimes \mathcal{I}^{(n)}$ where $\mathcal{I}^{(i)}$ is a basis of order $|\mathcal{V}^{(i)}| = n_i$ used to index elements of set $\mathcal{V}^{(i)}$. Finally, each set of n-tuples may be modeled as an element of $\mathcal{T}$, i.e. a **tensor**.

CMF Method, Step 2: State Tensor Construction (1/2)

- Due to its multi-dimensional nature the input tensor suffers from its big size and high sparsity.
- The idea is to transform the input tensor into a state tensor of reduced size.
- We apply the dimensionality reduction similar to N-way Random Indexing (NRI) approach [Sandin at al., 2009].
- The procedure may be seen as tensor-to-tensor transformation using $n_i \times m_i$ matrices $U^{(i)}$ $(i = 1, \ldots n)$, where n_i and m_i are the cardinality of i-th mode of the tensor before and after transformation, respectively.
- Each row of the transformation matrix (i.e, $(u^{(i)}_{k,1}, \cdots, u^{(i)}_{k,m_i})$) forms the random vector of specified length and specified 'seed' – each entry of the vector is set to be equal to 0 or 1, and then the vector is normalized using L^1-norm.

CMF Method, Step 2: State Tensor Construction (2/2)

- Let denote the state tensor $X = [x_{i_1,\dots,i_n}]_{m_1 \times \dots \times m_n}$.
- The proposed model assumes that before being used for the processing and querying procedures the state tensor needs to be preprocessed according to two following steps (i) *scaling in order to get the probability distribution* done as follows

$$x_{i_1,i_2,\dots,i_n} := \frac{x_{i_1,i_2,\dots,i_n}}{\omega},$$

where ω is the number of n-tuples used to build state tensor X, and (ii) *preparing to be used in L^2-norm operations* done by taking each entry square root value, i.e:

$$x_{i_1,i_2,\dots,i_n} := \left(x_{i_1,i_2,\dots,i_n}\right)^{1/2}$$

CMF Method, Step 3: Hierarchical Structure of Tensor Flattenings

- The main idea of CMF framework is to use the filter for each tensor mode which are calculated as the linear combination of covariance matrices determined based on state tensor X.
- The approach is motivated by the observation that the different relations in data may be seen depending on the choice of attributes used to model tensor modes.
- The procedure assumes the use of most detailed state tensor and its flattenings. Each flattening except the totally flatten tensor (i.e., the tensor flatten to the scalar), and flattenings to one mode (i.e., to vectors), takes part in the procedure of filters' construction.

CMF Method, Step 4: Overall centring and Covariance Matrices generation

- Each state tensor flattening X_j is centred according to the overall centring procedure. Let denote the result by X_j^c.
- The covariance matrices are generated as follows:
  - the unfolding matrix $X_j^{c,(i)} \in \mathbb{R}^{J_i \times (J_1 \times \cdots \times J_{i-1} \times J_{i+1} \times \ldots J_n)}$ is constructed, which collects i-th mode fibres of centred state tensor X_j^c as columns,
  - then, the symmetric matrix $A_j^{(i)} = [a_j^{(i)}]_{m_i \times m_i}$ such that:

$$A_j^{(i)} = X_j^{c,(i)} \left(X_j^{c,(i)} \right)^T$$

is obtained as a matrix representing the covariance between random dimensions used to enumerate the i-th mode.

CMF Method, Step 3-4: Flattenings and Covariance Matrices (example)

Tensor modes	Flattening code and symbol	Covariance matrices
user,item,time,location	1111, X_{15}	$A_{15}^{(1)}, A_{15}^{(2)}, A_{15}^{(3)}, A_{15}^{(4)}$
user,item,time	1110, X_{14}	$A_{14}^{(1)}, A_{14}^{(2)}, A_{14}^{(3)}$
user,item,location	1101, X_{13}	$A_{13}^{(1)}, A_{13}^{(2)}, A_{13}^{(4)}$
user,time,location	1011, X_{11}	$A_{11}^{(1)}, A_{11}^{(3)}, A_{11}^{(4)}$
item,time,location	0111, X_7	$A_7^{(2)}, A_7^{(3)}, A_7^{(4)}$
user,item	1100, X_{12}	$A_{12}^{(1)}, A_{12}^{(2)}$
user,time	1010, X_{10}	$A_{10}^{(1)}, A_{10}^{(3)}$
user,location	1001, X_9	$A_9^{(1)}, A_9^{(4)}$
item,time	0110, X_6	$A_6^{(2)}, A_6^{(3)}$
item,location	0101, X_5	$A_5^{(2)}, A_5^{(4)}$
time,location	0011, X_3	$A_3^{(3)}, A_3^{(4)}$

Figure: Flattenings and covariance matrices for the 4-mode tensor describing events concerning users (1st mode), items (2nd mode), time (3rd mode), and location (4th mode).

CMF Method, Step 5: Filter construction

- For mode i the optimal filter $F^{(i)}$ is constructed as a sum of an identity transformation and the average of matrices $A_j^{(i)}$:

$$F^{(i)} = I_i + \frac{1}{k} \sum_j A_j^{(i)},$$

where I_i is the identity matrix of size m_i, and k is a number of covariance matrices built for the i-th mode.

CMF Method, Step 6: Building the Prediction Tensor

- Just before applying the filters the tensor X is centred according to the overall centring approach.
- The filters $F^{(i)}$ are used in order to transform centred tensor X^c into its filtered version $\widetilde{X}^c$ according to the formula:

$$\widetilde{X}^c = X^c \times_1 F^{(1)} \times_2 \cdots \times_n F^{(n)}.$$

- As a final step the prediction tensor $\widetilde{X}$ is calculated as

$$\widetilde{X} = X - X^c + \widetilde{X}^c.$$

.

- The tensor $\widetilde{X}$ is used for calculating the prediction results according to the querying procedure.
- The prediction tensor may be compressed using the HOSVD approach [De Lathauwer, 2000] what leads to reduction of tensor size and, as consequence, shortens the time needed for querying.

CMF Method, Step 7: Querying Procedure

- The querying procedure is aimed at recalculating the entries of the input tensor using the prediction tensor.
- The procedure is again the tensor-to-tensor transformation, but due to practical reasons it is used as to reconstructing the single entry of the input data tensor.
- For a given n-tuple $\gamma = (k_1, \ldots, k_n)$ the query tensor $Q^\gamma = [q^\gamma_{i_1,\ldots,i_n}]_{m_1 \times \cdots \times m_n}$ is constructed as a tensor of the same size as the prediction tensor with the entries are calculated according to the formula:

$$q^\gamma_{i_1,\ldots,i_n} = (u^{(1)}_{k_1,i_1})^{1/2} \cdot \ldots \cdot (u^{(n)}_{k_n,i_n})^{1/2}.$$

- Finally, the result of the state tensor querying procedure is calculated as an inner product of the prediction tensor and query tensor Q^γ, as follows:

$$\widetilde{t}_\gamma = \sum_{1 \le i_1 \le m_1} \cdots \sum_{1 \le i_n \le m_n} \widetilde{x}_{i_1,\ldots,i_n} q^\gamma_{i_1,\ldots,i_n}.$$

Experimentation: Main Assumptions

- The scope of the experiments has been limited to a recommendation scenario that involves extreme data sparsity (training ratio equal to 0.1).
- Such a scenario is considered as challenging, but is also very common in the area of e-commerce, as an online merchandising recommender system is frequently provided with a few ratings per user.
- We followed the approach to the evaluation of recommendation system which is standard in Information Retrieval.
- The experimentation involves the evaluation of the proposed method both with (**R-CMF**) and without (**CMF**) the final HOSVD-based dimensionality reduction step.

Experimentation: Dataset Preparation

- We used one of the most widely referenced data sets for testing recommendation systems – the MovieLens ML100k set, which contains 100.000 ratings for 1682 movies given by 943 unique users.
- Additionally, we used the information about each movie's genre (19 distinct values).
- The input data was modeled as a tensor T with size equal to $2 \times 943 \times 1682 \times 19$ containing 100.000 non-zero entries.
- The training ratio was set to be equal to 0.1.
- The goal of the recommendation algorithm was to predict whether each rating in the test set is positive (given user likes a given movie) or negative (given user dislikes a given movie).
- Each recommendation quality measurement result represents the averaged result of 50 individual experiments.

Experimentation: Compared Methods

- The evaluation has been provide, both in terms of recommendation accuracy and computational efficiency.
- SotA methods used for comparison:
  - **NRI**,
  - **HOSVD**,
  - SVD-based collaborative filtering (referred to as **2-mode CF**) used as a baseline method,
  - **NRI+HOSVD**.
- For each of these methods we set all the necessary parameters optimally in order to provide the best possible accuracy:
  - the k-cut in **2-mode CF** to was set to 7,
  - the size of the final "core tensor" in HOSVD is set to $(2 \times 18 \times 24 \times 8)$,
  - the size of the state tensor (and prediction tensor) is set to $(2 \times 64 \times 64 \times 19)$, and the random vector's *seed* value was set to 4.

Experimentation: AUROC

2-mode CF	HOSVD	NRI	NRI+HOSVD	CMF	R-CMF
0.527 (±0.004)	0.562 (±0.004)	0, 550 (±0.004)	0, 549 (±0.004)	**0, 585** (±0.004)	*0,582* (±0.005)

Figure: The average AUROC results for all tested algorithms

Experimentation: Computation Time

	HOSVD	NRI	NRI+HOSVD	CMF	R-CMF
model build	48.6	**9.0**	*9.1*	*9.1*	9.2
model query	*28.7*	77.7	**20.9**	77.7	**20.9**
total	77.3	86.7	**30.0**	86.8	*30.1*

Figure: The average execution times (in seconds).

Conclusions and Future Work

- Each of the methods based on tensor modeling outperforms the standard collaborative filtering based on matrix factorization.
- The use of filters based on the hierarchical structure of covariance matrices built using various tensor flattenings (various generalization), enables to extract additional dependencies in input data what leads to a further performance improvement.
- The proposed method could be effectively used for large data sets and for multi-mode processing (as a result application of dimensionality reduction techniques – both based on RI and HOSVD).
- Future work directions:
  - Filtering methods for streaming data processing.
  - Optimization of the final filter construction formula.

Thank you!