Fast Similarities in Factorized Tensors

Michael E. HOULE\(^1\)  
Hisashi KASHIMA\(^2\)

\(^1\)National Institute of Informatics, Japan  \(^2\)The University of Tokyo, Japan

**Summary**

Low-rank factorizations of higher-order tensors have become an invaluable tool for researchers from many scientific disciplines. Tensor factorizations have been successfully applied for moderately sized multi-modal data sets involving a small number of modes. However, a significant hindrance to the full realization of the potential of tensor methods is a lack of scalability on the client side: even when low-rank representations are provided by an external agent possessing the necessary computational resources, client applications are quickly rendered infeasible by the space requirements for explicitly storing a (dense) low-rank representation of the input tensor.

We consider the problem of efficiently computing common similarity measures between entities expressed by fibers (vectors) or slices (matrices) within a given factorized tensor. We show that after appropriate preprocessing, inner products can be efficiently computed independently of the dimensions of the input tensor.

**Motivation**

Within many emerging areas of computing, such as data mining, recommendation systems, security, and multimedia, applications of similarity search naturally arise in the context of such fundamental tasks as clustering, classification, matching and detection. Multi-modal data can be naturally represented in the form of a tensor (also known as a multiway array), a higher-dimensional extension of the matrix representation. Tensor-based data modeling is particularly appealing whenever the data dynamics can be captured by truncated (low-rank) representations, in terms of a small number of latent variables. However, due to the complexity inherent in managing multi-modal data, effective and scalable strategies for similarity search are crucial to the overall performance of such systems.

**Tensors and Similarities**

Tensors are a suitable means to represent multi-object relationships, such as ratings given by customers to restaurants at certain times.

User profiles can be represented as slices (matrices) and rating profiles as fibers (vectors). The need to efficiently search for similar rating- or user profiles arises in applications such as, for example, recommender systems.

**Factorization Models**

- **The CP model** describes a tensor \(X\) as a conical combination of rank-1 tensors \(x_{1}^{\otimes m}, \ldots, x_{m}^{\otimes m}\). The individual \(x_{i}^{\otimes m}\) may be expressed in terms of vector products:

\[
X = \bigotimes_{i=1}^{m} x_{i}^{\otimes m} = \bigotimes_{i=1}^{m} (\bigotimes_{j=1}^{p} u_{ij}^{\otimes p}) \otimes (\bigotimes_{j=1}^{q} v_{ij}^{\otimes q})
\]

- **The Tucker model** is a form of higher-order SVD. It represents a tensor \(X\) as the ‘mode-wise’ product of a core tensor \(C\) and \(p\) factor matrices.

\[
X = \bigotimes_{i=1}^{m} (\bigotimes_{j=1}^{p} u_{ij}^{\otimes p}) \otimes (\bigotimes_{j=1}^{q} v_{ij}^{\otimes q})
\]

- **The pairwise interaction model** expresses a tensor \(X\) by independent interaction between individual pairs of factors \(U^{(k)}\) and \(V^{(k)}\).

\[
X = \bigotimes_{i=1}^{m} (\bigotimes_{j=1}^{p} u_{ij}^{\otimes p}) \otimes (\bigotimes_{j=1}^{q} v_{ij}^{\otimes q})
\]

**Preprocessing**

Depending on the factorization model, we precompute the following values:

- Given a rank-\(m\) CP model, store inner products of the form \(\Phi(u^{(i)}_{ij}, u^{(k)}_{ij})\), for each mode \(1 \leq k \leq p\) with \(1 \leq i \leq m\).
- Given a rank-\((m_1,\ldots,m_p)\) Tucker model, store \(\Phi(u^{(i)}_{ij}, u^{(k)}_{ij})\) for each mode \(1 \leq i \leq \ell \leq m\).
- Given a PITF model, store inner products of the form \(\Phi(u^{(i)}_{ij}, u^{(k)}_{ij})\) and \(\Phi(u^{(i)}_{ij}, u^{(k)}_{ij})\) for any choice of \(a,b \in \{1,\ldots,m\}\) and \(i,j,k \in \{1,\ldots,p\}\).

**Fast Similarity Computation**

When computing similarities between substructures (such as fibers or slices) we can avoid the full model size \((n_1,\ldots,n_p)\) by using the previously computed inner product values. For example, given a rank-\(m\) CP model of a tensor, we can compute the similarity of two fibers using:

\[
\Phi_{CP}(x_{ij}, y_{ij}) = \sum_{k=1}^{m} \Phi_{ij,k}(x_{ij}, y_{ij}) = \sum_{k=1}^{m} \sum_{r=1}^{w} \Phi_{ij,kr}(x_{ij}, y_{ij})
\]

The process is similar for higher-order substructures (slices, …).

**Complexities**

Complexities of the different factorization models, as compared to operations on the original unfactorized tensor.

**Performance**

Figure: Speed-ups achieved for a rank-\(m\) CP factorization of an order-\(p\) tensor.

Figure: Speed-ups achieved for a rank-\((m_1,\ldots,m_p)\) Tucker factorization of an order-\(p\) tensor. The computation is distributed across \(q\) processors.

**Additional Material**

- Technical Report
- Poster
- Implementation
- Documentation