Fast Similarities in Factorized Tensors

Michael E. HOULE [1] Hisashi KASHIMA [2] Michael NETT [1,2]

[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.

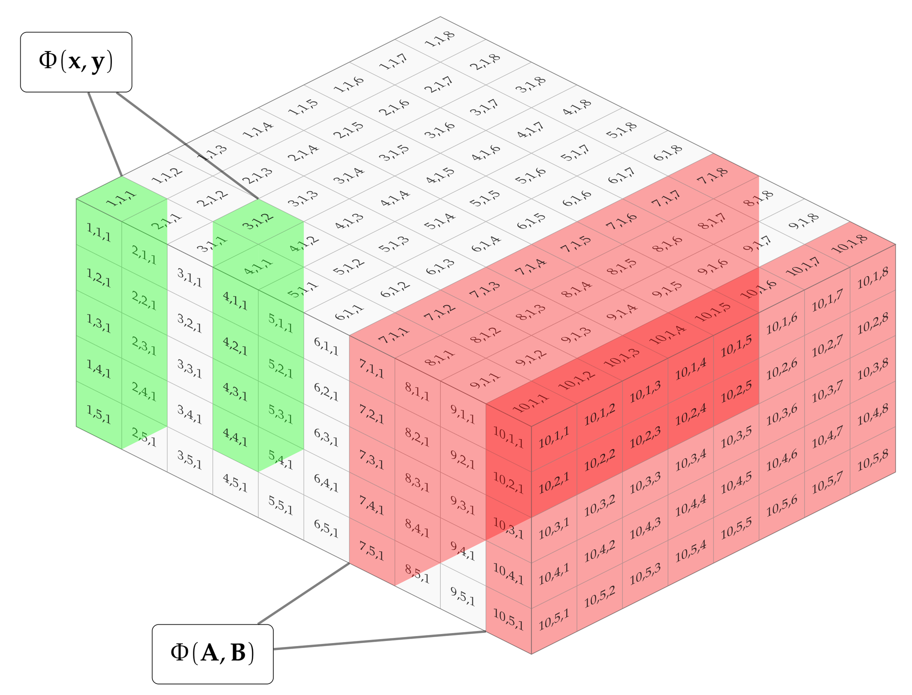
Motivation

Within many emerging areas of computing, such as data mining, recommendation systems, security, and multimedia, applications of similarity search naturally arises 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 $\mathfrak{X}$ as a conical combination of rank-1 tensors $\mathfrak{X}^{[1]},\dots,\mathfrak{X}^{[m]}$. The individual $\mathfrak{X}^{[i]}$ may be expressed in terms of vector products.

$$\hat{\mathfrak{X}} = \begin{bmatrix} 4 & 6 & 11 \\ 2 & 2 & 5 \\ 3 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 1 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 6 & 9 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \circ \begin{bmatrix} 2 & 1 & 1 \end{bmatrix}$$

- The *Tucker model* is a form of higher-order SVD. It represents a tensor $\mathfrak{X}$ as the 'mode-wise' product of a core tensor $\mathfrak{C}$ and p factor matrices.

$$\mathfrak{C} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{U}^{[1]} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 2 & 1 \end{bmatrix}, \quad \mathbf{U}^{[2]} = \begin{bmatrix} 1 & 2 \\ 0 & 2 \\ 0 & 1 \end{bmatrix}, \quad \hat{\mathfrak{X}} = \begin{bmatrix} 5 & 2 & 1 \\ 1 & 0 & 0 \\ 7 & 4 & 2 \end{bmatrix}$$

$$\hat{x}_{31} = c_{11}u^{[1]}_{31}u^{[2]}_{11} + c_{12}u^{[1]}_{31}u^{[2]}_{12} + c_{21}u^{[1]}_{32}u^{[2]}_{11} + c_{22}u^{[1]}_{32}u^{[2]}_{12}$$

- The *pairwise interaction model* expresses a tensor $\mathfrak{X}$ by independent interaction between individual pairs of factors $\mathbf{U}^{[i,j]}$ and $\mathbf{U}^{[j,i]}$.

$$\hat{\mathfrak{X}} = \begin{bmatrix} 1 & 3 & 0 \\ 2 & 7 & 2 \\ 1 & 3 & 0 \end{bmatrix}, \quad \mathbf{U}^{[1,2]} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{U}^{[2,1]} = \begin{bmatrix} 1 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Preprocessing

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

- Given a rank-m CP model, store inner products of the form $\Phi(\mathbf{u}^{[i]}_k, \mathbf{u}^{[j]}_k)$, for each mode $1 \leqslant k \leqslant p$ with $1 \leqslant i \leqslant j \leqslant m$.
- Given a rank-$(m_1,\dots,m_p)$ Tucker model, store $\Phi(\mathbf{u}^{[k]}_i, \mathbf{u}^{[k]}_j)$ for each mode $k \in \{1,\dots,p\}$ and $1 \leqslant i \leqslant j \leqslant m_k$.
- Given a PITF model, store inner products of the form $\Phi(\mathbf{u}^{[i,j]}_a)$ and $\Phi(\mathbf{u}^{[i,j]}_a, \mathbf{u}^{[i,k]}_b)$ for any choice of $a,b \in \{1,\dots,m\}$ and $i,j,k \in \{1,\dots,p\}$.

Fast Similarity Computation

When computing similarities between substructures (such as fibers or slices) we can avoid the full model size $(n_1,\dots,n_p)$ 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:

$$\begin{aligned}
&\Phi_{CP}(\hat{x}_{\square i_2\cdots i_p}, \hat{x}_{\square j_2\cdots j_p}) \\
&= \sum_{k=1}^{n_1} \hat{x}_{k i_2\cdots i_p}\hat{x}_{k j_2\cdots j_p} \\
&= \sum_{k=1}^{n_1} \left(\sum_{a=1}^{m} w_a u^{[a]}_{1k} u^{[a]}_{2i_2} \cdots u^{[a]}_{pi_p} \sum_{b=1}^{m} w_b u^{[b]}_{1k} u^{[b]}_{2j_2} \cdots u^{[b]}_{pj_p} \right) \\
&= \sum_{a,b=1}^{m} w_a w_b u^{[a]}_{2i_2} \cdots u^{[a]}_{pi_p} u^{[b]}_{2j_2} \cdots u^{[b]}_{pj_p} \sum_{k=1}^{n_1} u^{[a]}_{1k} u^{[b]}_{1k} \\
&= \sum_{a,b=1}^{m} w_a w_b u^{[a]}_{2i_2} \cdots u^{[a]}_{pi_p} u^{[b]}_{2j_2} \cdots u^{[b]}_{pj_p} \Phi(\mathbf{u}^{[a]}_1, \mathbf{u}^{[b]}_1)
\end{aligned}$$

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.

Complexity	Original	Tucker	CP	PITF
Size	$\prod_i n_i$	$\prod_i m_i + \sum_i m_i n_i$	$m \sum_i n_i$	$mp \sum_i n_i$
Access	1	$p \prod_i m_i$	mp	mp^2
Prepr.	1	$\sum_i m_i^2$	$m^2 p$	$m^2 p^2$
Naïve Sim.	n_k	$n_k p \prod_i m_i$	$n_k m p$	$n_k m p^2$
Fast Sim.	—	$p m_k \prod_i m_i$	$m^2 p$	$m^2 p^4$

Performance

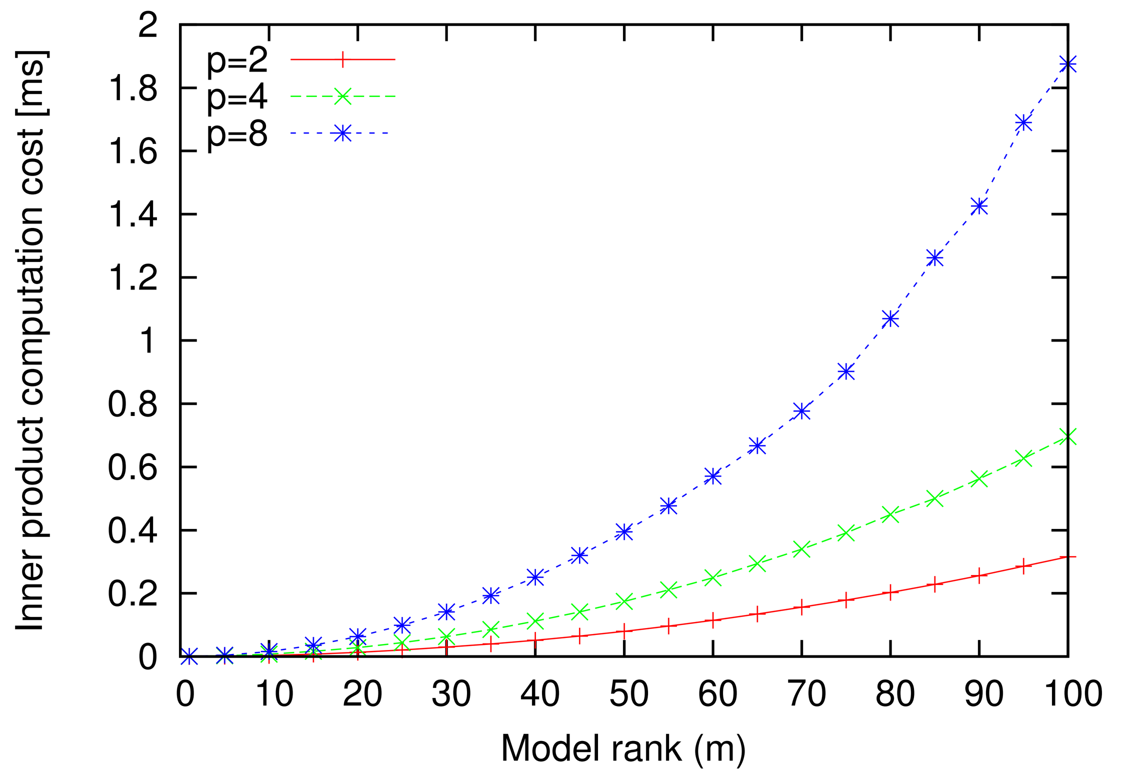


Figure: Speed-ups achieved for a rank-m CP factorization of an order-p tensor.

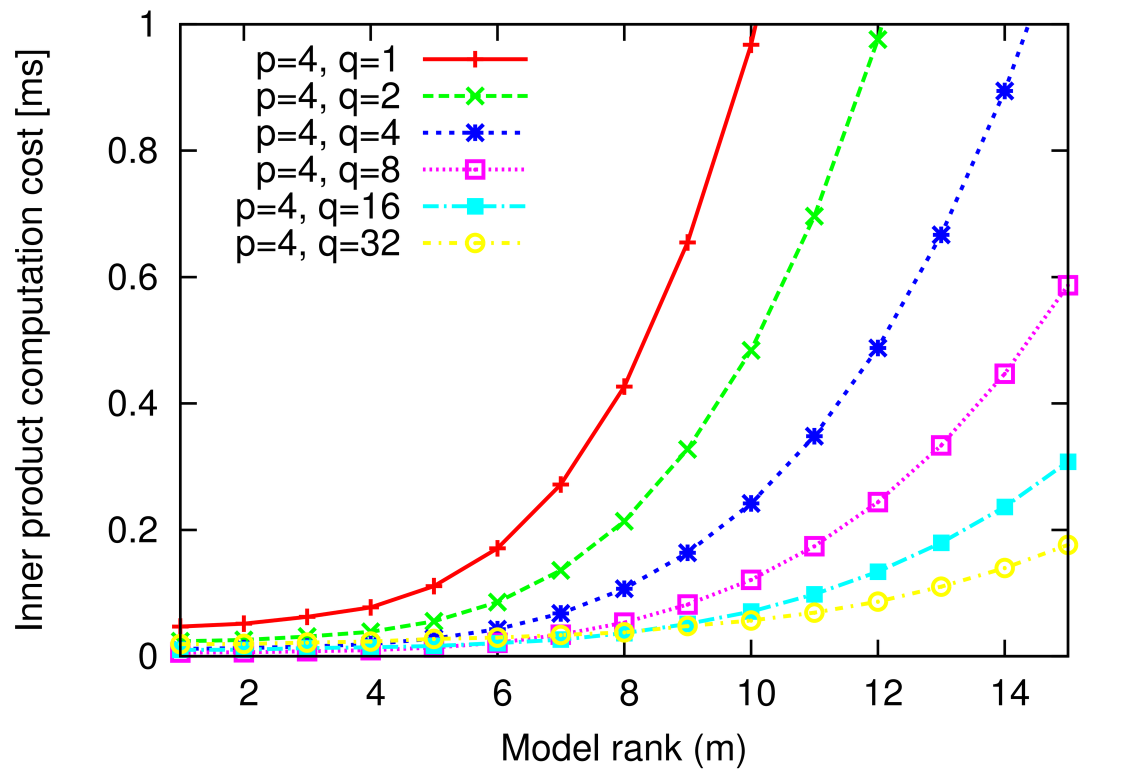


Figure: Speed-ups achieved for a rank-$(m \times \cdots \times m)$ Tucker factorization of an order-p tensor. The computation is distributed across q processors.

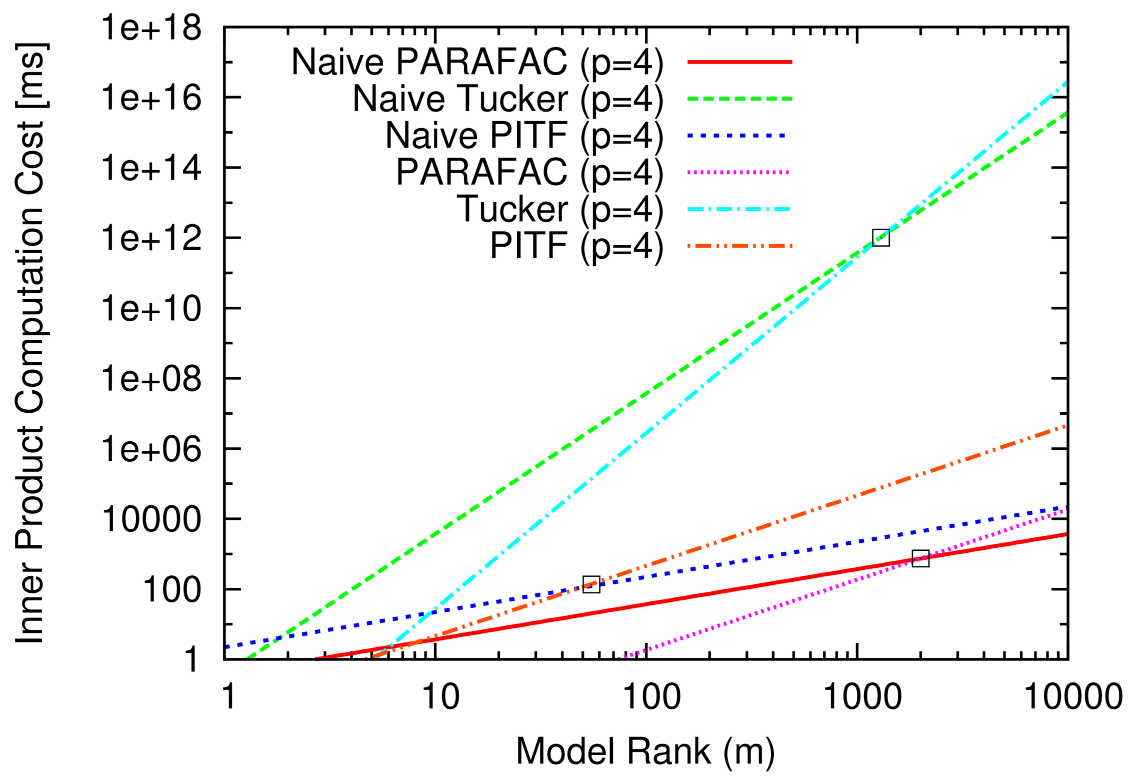


Figure: Break-even points of naïve and fast similarity computations.

Additional Material

- Technical Report
- Poster
- Implementation
- Documentation

NII 連絡先: Michael E. HOULE (フール マイケル) / 国立情報学研究所 客員教授
☎: 03-4212-2538 FAX: 03-3556-1916 ✉: meh@nii.ac.jp