Finding Large Elements 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 tensor decompositions have become valuable tools for many practical problems arising in machine learning and data mining applications concerned with higher-order data. While much research has focused on specific methods for decomposition, only a limited amount of attention has been given to the process of locating elements of interest from within a tensor.
Efficient strategies for identifying and extracting targeted elements are very important in practical applications, which frequently operate on scales at which one cannot afford the time requirements associated with brute force methods. We propose several algorithms for solving or approximating the problem of locating the k largest elements of a tensor given only its factorization. We provide experimental evidence that our methods enable follow-on applications in areas such as link prediction and recommender systems to operate at a significantly larger scale.

Motivation

In virtually all practical applications of high-order tensors, the number of elements of the tensor is far larger than the number of objects in the data set. Generally speaking, in order to be effective, tensor-based applications must take advantage of the sparsity of the representation, and avoid the expensive brute-force scanning of tensor elements. One of the most important issues in the area of tensor analysis is therefore that of tensor completion, where after observing only a limited number of element values, one seeks to deduce properties of the remaining unobserved values. Depending on the application, one usually requires knowledge of a relatively small set of tensor elements, rather than the entire unobserved portion of the tensor at hand. Tensor completion strategies have many real world applications including, but not limited to, personalized tag recommendation and link prediction in social web services.

Vector, Matrix and Tensor Products

- Let $\mathfrak{X}$ and $\mathfrak{Y}$ be tensors of dimensions $n_1, \ldots, n_p$. Their *element-wise* product $\mathfrak{X} * \mathfrak{Y}$ is given by

$$[\mathfrak{X} * \mathfrak{Y}]_{i_1 \ldots i_p} = [\mathfrak{X}]_{i_1 \ldots i_p} \cdot [\mathfrak{Y}]_{i_1 \ldots i_p}.$$

- Let $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(p)}$ be vectors in $\mathbb{R}^n$. Their *inner product* is

$$\Phi(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(p)}) = \sum_{i=1}^{n} \prod_{j=1}^{p} u_i^{(j)}.$$

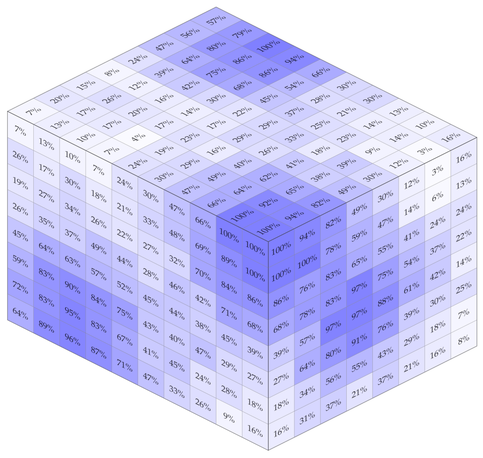
- Let $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(p)}$ be vectors with $\mathbf{u}^{(i)} \in \mathbb{R}^{n_i}$ for $1 \leqslant i \leqslant p$. Their *outer product* is the order-p tensor satisfying

$$[\Phi(\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(p)})]_{i_1 \ldots i_p} = \prod_{j=1}^{p} u_{i_j}^{(j)}.$$

Note that, whenever only two vectors $\mathbf{u}$ and $\mathbf{v}$ are involved, one may conveniently use the simpler notation of $\Phi(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v}$ and $\Psi(\mathbf{u}, \mathbf{v}) = \mathbf{u}\mathbf{v}^\top$.

Problem Definition

Let $\mathfrak{X}$ be a real-valued order-p tensor in $\mathbb{R}^{n_1 \times \cdots \times n_p}$ and let $k \in \mathbb{N}$. The *top-k problem* is to find a set S of tensor elements whose sum is maximal, subject to $|S| = \min\{k, n_1 \cdots n_p\}$.



Data Model

The provided data is modeled using a low-rank factorization with factor matrices $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(p)}$:

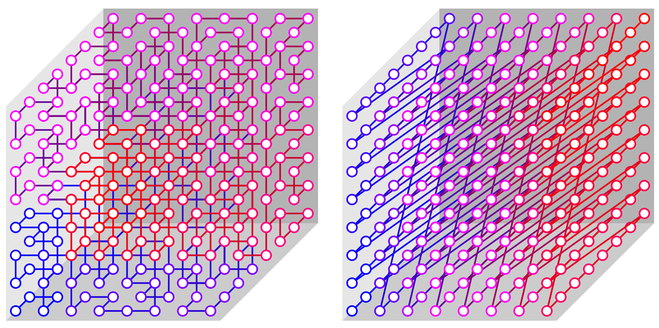
$$\mathfrak{X} = \sum_{i=1}^{m} \mathfrak{X}^{(i)} = \sum_{i=1}^{m} \Psi(\mathbf{u}_{:i}^{(1)}, \ldots, \mathbf{u}_{:i}^{(p)})$$

The tensor element at location $(i_1, \ldots, i_p)$ is the inner product of the correpsonding rows of the factor matrices:

$$[\mathfrak{X}]_{i_1 \ldots i_p} = \sum_{j=1}^{r} \left[\Psi(\mathbf{u}_{:j}^{(1)}, \ldots, \mathbf{u}_{:j}^{(p)})\right]_{i_1 \ldots i_p} = \sum_{j=1}^{r} \prod_{k=1}^{p} u_{i_k j}^{(k)}$$
$$= \Phi(\mathbf{u}_{i_1:}^{(1)}, \ldots, \mathbf{u}_{i_p:}^{(p)})$$

Method I – Enumeration

An exhaustive enumeration of all tensor elements provides exact answers to *top-k queries* in time $\Theta(n_1 \cdots n_p (rp + \log k))$.



Method II – Greedy Heuristic

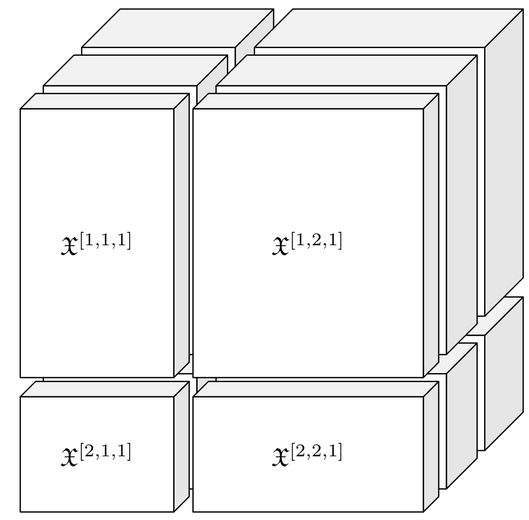
An indexing-based method which provides approximate answers to *top-k queries* in time $\mathbf{O}(p^2 sktr)$.

Algorithm 2 The `Greedy` heuristic provides approximate answers to top-k queries by greedily combining only the most promising parts of the factor models. Parameters are the desired solution size $k \in \mathbb{N}$ and a scaling factor $s \geq 1$.

for $i \leftarrow 2 \ldots p$ **do**
  Build similarity search index T_i on $U^{(1)}$, the set of rows of $\mathbf{U}^{(i)}$.
end for
$R \leftarrow \emptyset$
$Q \leftarrow U^{(1)}$
for $i \leftarrow 2, \ldots, p$ **do**
  for $\mathbf{q} \in Q$ **do**
    Compute fitness $\varphi(\mathbf{q}) = \|\mathbf{q}\|_2$.
  end for
  Compute $\varphi(Q) = \sum_{\mathbf{q} \in Q} \varphi(\mathbf{q})$.
  $Q' \leftarrow \emptyset$
  for $\mathbf{q} \in Q$ **do**
    Let $c \leftarrow \lceil sk \cdot \varphi(\mathbf{q}) / \varphi(Q) \rceil$.
    for $\mathbf{p} \in NN_{T_i}(\mathbf{q}, c)$ **do**
      $Q' \leftarrow Q' \cup \{\mathbf{q} * \mathbf{p}\}$
    end for
  end for
  Retain the $\lceil sk \rceil$ best candidates from Q'.
  $Q \leftarrow Q'$
end for
$R \leftarrow \{\Phi(\mathbf{q}) \mid \mathbf{q} \in Q\}$
Return the k-largest elements in R.

Method III – Divide and Conquer

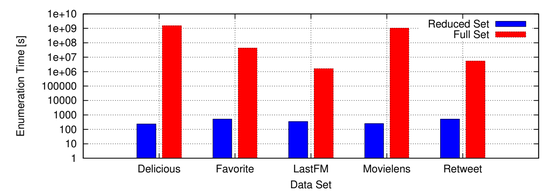
A clustering-based divide and conquer method which aims to reduce the time complexity by discarding as many subproblems as possible.



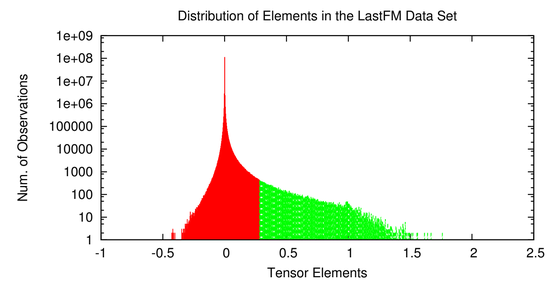
Individual subproblems can either be solved exactly via enumeration, or approximate using the previously proposed heuristic.

Importance of Scalability

Projections show that for real application data, exhaustive query processing may require up to 31 years.

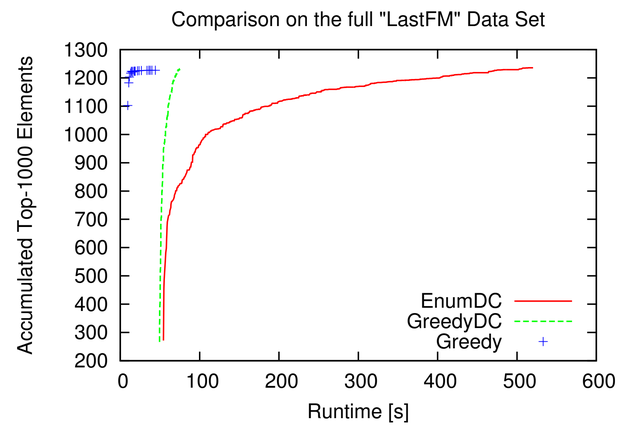


The proportion of uninteresting tensor elements is overwhelmingly large.



Evaluation

Performance of the methods for the *LastFM* data set. The input tensor has size 2,100 × 18,744 × 12,647.



Additional Material

- Technical Report
- Poster
- Implementation
- Documentation

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