Journal of Information and Intelligence

Tesseract Factorization

--Manuscript Draft--

Manuscript Number:	InteCom-D-22-00006
Full Title:	Tesseract Factorization
Article Type:	Full Length Article
Keywords:	Modes, data sprase tensors, tesseract factorization, matrices in higher dimensionality.
Corresponding Author:	Christodoulos Asiminidis Imperial College London UNITED KINGDOM
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Imperial College London
Corresponding Author's Secondary Institution:	
First Author:	Christodoulos Asiminidis
First Author Secondary Information:	
Order of Authors:	Christodoulos Asiminidis
Order of Authors Secondary Information:	
Manuscript Region of Origin:	UNITED KINGDOM
Abstract:	Tensors and therefore modes are becoming of paramount importance in today's data science sector. In this example, we examine the real need for highly optimized sparse tensor tool. I introduced tesseract factorization so that a library of C shall be expanded therefore as future work. I presented a method of reordering sparse tensors and cache tiling to improve data locality taken into consideration bioinformatics and recommender systems real-world applications. On average, the particular problem is estimated to offere over 6 times faster than the baseline and 3 times faster than the previous version using one thread on the computer and scales to average 58 times faster at 16 threads. A challenging and distinctive characteristic of many tensor computations is the mode that requires representation of the tensor for each mode. A future work would be to this model investigating utilizing cachefriendly reordering and novel form of cache tiling.
Suggested Reviewers:	David Evans Imperial College London david.evans@imperial.ac.uk
	Alexander Ivanov Imperial College London a.ivanov@imperial.ac.uk
Additional Information:	
Question	**Response**

Declaration of interests

☐The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Christodoulos Asiminidis reports writing assistance was provided by Imperial College London. Christodoulos Asiminidis reports a relationship with Imperial College London that includes: non-financial support. Christodoulos Asiminidis has patent pending to Christodoulos Asiminidis. There is no other conflict of interest by the reader.

RESEARCH

Tesseract Factorization

Full list of author information is available at the end of the article

Abstract

First part title: Tesseract Factorization.

Second part title: Tesseract Factorization

Keywords: Tesseract; Factorization; Modes; Sparse tensors; Model Accuracy

Conflict of interests

The authors declare that they have no conflict of interest.

Authors' information

Christodoulos Asiminidis, Imperial College London, +306986076324, +357997758931.

elsarticle.cls – A better way to format your document

Elsevier Ltd

QUICK LINKS

1. Introduction

Data Structures are one of the most important part in computer science because that is a way in order for the data to be stored. Data structures such as queues, stacks, graphs, arrays are commonly used in order to store the data and therefore the data data shall be processed and analyzed. For that it is necessary to build highly and powerfully sufficiently data structure so that they offer high speeds in data processing as well as in the least computational cost. In this work, I propose a novel data structure, under the name of Tesseract factorization because it is a data structure that derives from matrices, in the forth dimension.

In every day computer and/or mobile device-wise usage, there is always the need to store, save and process data all the time. That need can be expanded from simple qna forms to a much more complex multi-dimensional arrays, tensors to process and solve complex issues. In these issues, signal preprocessing such as Wavelets signals found in the literature in bioinformatics. Recommender systems is an additional field in which matrix and especially tesseract factorization is of paramount importance in order to preprocess, analyze and carry out any additional tasks such as machine learning in order to reduce for example the dimension of the data through latent features, SVD and PCA.

In this work, I present a method in which it should be used in order to deal with successfully and sufficiently with common dimensionality reduction problems such as tensor factorization. It mostly focuses on novel data structure that is able to take advantage the sparsity of the data compared to the ones classified as zeros. This data structure has got a small memory footprint so that it can highly and quickly be taken advantage of.

2. Tesseract Notation

In this section, I provide a short descritpion on tensors. I therefore describe the Canconical Polyadic Decomposition, used in tensor factorization.

A. Tensor Notation

Tensors are the generalization of matrices to higher dimensions. Tesseract is the specialization into the fourth dimension compared to the second one which is the matrix one. Well, the dimensions in this case of tensor factorization follow the name modes. In addition, we can call a tensor with n modes is of order n mathematically-wise speaking. For instance, a tensor of order four receives the shape a box with a box included in it. In this work, I mostly focus on the four dimension order of the data structure because however the fourth dimension, it is still scientifically-

Version: 3.3
Date: July 6, 2022
Contact: elsarticle@stmdocs.in

wise speaking simple and can be visualized. In rder to present data structure for representing sparse tensors is list of oc coordinates as is in the for example, two dimension, in the matrix one. So, let's assume for example (i, j, k, l) is a coordinates which indices the i is the first mode, j in the second, k in the third and l in the forth coordinate accordingly. In this work, I denote tensors as X and matrices as A. I write the coordinate system (i, j, k, l) of X as X(i, j, k, l) as I would write a matrix A as A(i,j) in the mode-2 coordinate system. The sparse tensor X is of dimension $I \times J \times K \times L$. and has m nonzeros. In addition, X(:, j, :, :) denotes all the values of the column j in the X sparse tensor as well as A(i, :) denotes of the values of the row i in the matrix A. Tesseracts are a building block of tensors. In tesseract factorization, the first order is called rows, the second order is called columns, the third one is called tubes and the fourth one is called tesseract. Two of paramount importance on matrices used in the CD are the Hadamard product and Khatri-Rao products (Citation Karypis paublication).

3. Tesseract Factorization

I developed an algorithm under the name of tesseract factorization which uses a novel data structure for representing tensors. My algorithm computer entire rows of a matrix M, in this case, M would appropriately take the name of tesseract at a time and as a result only requires a single traversal of the sparse tensor structure. My work is specialized in the form of tesseract factorization, a four-mode tensors with shared-memory parallelism. In this section, we first show a derivation of our algorithm and an analysis of its data structure and computational performance. We discuss its generalization to an arbitrary number of modes and lastly discuss its parallelization.

B. TessearctTensor Tesseract Tensor is a parallel CPD-ALS Algorithm developed for MapReduce paradigm. Tesseract Tensor utilizes the massive paralelism of MapReduce by reformulating MTTKRP as a series of Hadamard product. That means that the products are calculated for each one separately, creating a distribution of products each time. That means that it avoids the multiplication of the product of the total ones. The matrix N has the same sparsity pattern as X(1) and each non zero entry N(i, y, z) = X(1)B(y

A. Derivation Let me shortly presume that X is dense, and so each row of X(1) has exactly JK and eaacho column has exactly LM nonzeros. If we start from (1),

$$M(i,f,g) = \sum_{y=0}^{LM} \sum_{z=0}^{JK} X(1)(i,z,y)B(zJ,f,g)C(z/J,f,g) \tag{1}$$

$$M(i,f,:) = sum_{y=0}^{LM} X\_(1)(i,f,z)(B(z\%J,f,:) * C(z/J,f,:)) \tag{2}$$

$$= \sum_{k=0}^{K} \sum_{j=0}^{J} \sum_{l=0}^{L} \sum_{m=0}^{M} X(i,j,k,l,m)(B(m,l,j,:) * C(m,l,:,k) * D(m,:,j,k) * E(:,l,j,k)) \tag{3}$$

$$M(i,f,:) = \sum_{k=0}^{K} B(m,l,j,:) \sum_{j=0}^{J} C(m,l,:,k) \sum_{l=0}^{L} D(m,:,j,k) \sum_{m=0}^{M} X(i,j,k,l,m)E(:,l,j,k) \tag{4}$$

Therefore, we write equation 1 Tesseract Tensor on a row of M. We break the columns of X into J, K, L and M components to arrive at equation 3. We are able to factor out the inner multiplication of C(m, l, :, k) and reach the more efficient solution to 4. The factored C term saves F(J-1) multiplications per X(i, :, k, l, m) tesseract. If X is spares and tessearact X(i, :, k ,l ,m) has J non zeros, then F(J-1) FLOPS are saved, resulting in a total 2F(m+P) FLOPs.

Algorithm 1 An algorithm with caption

Input: Slice X(i, :, : , :,:)
Output: Row M(i, :, :)
M(i, :, :) ¡- 0
for all unique k, j, l and m ∈ X(i, :, :, :, :) do each X(i, :, :, : , :) tesseract
accum(:)¡-0 Fx1 vector for accumulation
for all j ∈ X(i, :, k , l , m)
accum(:) ¡- accum(:) + X(i, j, k , l ,m)C(m, l, :, k)
end for
M(i, :, :) ¡- M(i, f) + C(k, f)accum(:)
end for

The aforementioned algorithm works for s single slice of tesseract X.

B. Computational cost and storage In this work, I represent resseract factorization algorithm. Each tensor is saved as a list of slices. Each slice includes a list of tensors, precisely tesseracts represented as sparse vectors. Slices are stored in a data structure that is similar to a compressed sparse row matrix. Nevertheless, slices in a data structure are very often sparse since they cannot be fully added with data. Each tesseract is accompanied by a tesseract which specifies its for example K coordinate

index. Let X tesseract includes with P slice-2 tesseract, with 2 dimensions. Tesseract library uses m floating point numbers and m integers to store each nonzero value and it j coordinate. For rows respectively, a CSR matrix, (P+1) integers are required to save the initial indices for the tesseract and a value used for each tesseract id. At last, (I+1) integers are used to mark the initial indices of each slice. The total memory footprint of X tesseract is (2m + I + J + K + L + M + P +2) words. The only additional memory is used to save the F inner products that are used to update the M dimensionality reduction data structure. Tesseract uses 2F(m+P) FLOPs which is identical to DFacto citation. Tesseract needs to save F additional floating point numbers during computation.

4. Conclusion

Tensors and therefore modes are becoming of paramount importance in today's data science sector. In this example, we examine the real need for highly optimized sparse tensor tool. I introduced tesseract factorization so that a library of C shall be expanded therefore as future work. I presented a method of reordering sparse tensors and cache tiling to improve data locality taken into consideration bioinformatics and recommender systems real-world applications. On average, the particular problem is estimated to offere over 6 times faster than the baseline and 3 times faster than the previous version using one thread on the computer and scales to average 58 times faster at 16 threads. A challenging and distinctive characteristic of many tensor computations is the mode that requires representation of the tensor for each mode. A future work would be to this model investigating utilizing cache-friendly reordering and novel form of cache tiling. This is the first work investigate reordering and cache tiling in this context.