Numerical tensor methods: algorithms and tools

I. V. Oseledets,

Skolkovo Institute of Science and Technology

12 May 2014

Сколковский институт науки и технологий

What this talk is about

Many people work on tensors

What this talk is about

Many people work on tensors

Meaning different things

To name a few

- Eigenvalues of tensors (E-, Z-)
- Canonical forms
- Low parametric representation

Our goal

We work on a specific problem on low-parametric approximation of high-dimensional tensors using the idea of separation of variables

Our goal

We work on a specific problem on low-parametric approximation of high-dimensional tensors using the idea of separation of variables

Not only approximation, but also efficient large computation of basic operations

What kind of operations

We want fast operations in the tensor format

- Basic linear algebra operations
- Recovery from few samples
- Computation of such operations as convolution, solution of linear systems
- Ability to do “rounding” (recompression)

Standard

The standard approach is the canonical format

$$A(i_1, \dots, i_d) = \sum_{\alpha=1}^{r} U_1(i_1, \alpha) \dots U_d(i_d, \alpha)$$

See review by Kolda and Bader

Problems with canonical format

de Silva, Lim:

- Best approximation may not exist
- No robust algorithms
- Rank depends on the field

Novel tensor formats

This motivated the development of novel tensor formats

- H-Tucker format (Hackbusch, Kuhn, Gradedyck, 2009)
- Tensor Train (Oseledets, Tyrtyshnikov, 2009)
- Known to physists and chemists under different names, Matrix Product States, MCTDH

Novel tensor formats

This motivated the development of novel tensor formats

- H-Tucker format (Hackbusch, Kuhn, Gradedyck, 2009)
- Tensor Train (Oseledets, Tyrtyshnikov, 2009)
- Known to physists and chemists under different names, Matrix Product States, MCTDH

Great development in algorithms in software has been made

I will talk about the Tensor Train (TT) format

Definition

Tensor is said in the TT-format, if

$$A(i_1, \dots, i_d) = G_1(i_1)G_2(i_2)\dots G_d(i_d),$$

where $G_k(i_k)$ — matrix of size $r_{k-1} \times r_k$, $r_0 = r_d = 1$

r_k are called TT-ranks

$G_k(i_k)$ (which are $r_{k-1} \times n_k \times r_k$ tensors) are called cores

TT in a nutshell

- $\mathbf{A}$ — canonical rank $r \to r_k \leq r$
- TT-ranks are matrix ranks, TT-SVD
- All arithmetic, linear in d, polynomial in r
- Fast TENSOR ROUNDING
- TT-cross, exact interpolation formula, recent: quasioptimality results (D. Savostyanov)
- Q(Quantics, Quantized)-TT decomposition — binarization (or tensorization) of vectors and matrices (B. Khoromskij, O.)
- TT-Toolbox – software, S. V. Dolgov, I.V. Oseledets, D. V. Savostyanov, V. A. Kazeev

Skoltech
Сколковский институт науки и технологий

TT-ranks — matrix ranks

Define unfoldings:

$A_k = A(i_1 \ldots i_k; i_{k+1} \ldots i_d)$, $n^k \times n^{d-k}$ matrix

TT-ranks — matrix ranks

Define unfoldings:

$A_k = A(i_1 \dots i_k; i_{k+1} \dots i_d)$, $n^k \times n^{d-k}$ matrix

Theorem: There exists a TT-decomposition with TT-ranks

$$r_k = \operatorname{rank} A_k$$

TT-ranks — matrix ranks

The proof is constructive and gives the TT-SVD algorithm (Vidal algorithm in quantum information)

TT-ranks — matrix ranks

There is no exact low ranks need stability estimate!

Theorem (Approximation theorem)

If $A_k = R_k + E_k$, $||E_k|| = \varepsilon_k$

$$||\mathbf{A} - \mathbf{TT}||_F \leq \sqrt{\sum_{k=1}^{d-1} \varepsilon_k^2}.$$

Fast linear algebra

Addition, Hadamard product, scalar product
All linear in d

Fast linear algebra

$$C(i_1, \dots, i_d) = A(i_1, \dots, i_d) B(i_1, \dots, i_d)$$

$$C_k(i_k) = A_k(i_k) \otimes B_k(i_k),$$

ranks are multiplied

Tensor rounding

$\mathbf{A}$ is given in TT-format with suboptimal ranks.
Who to reapproximate?

Сколковский институт науки и технологий

Tensor rounding

It can be done in $\mathcal{O}(dnr^3)$ operations

Cross approximation in d-dimensions

What if a tensor is given as a “black box”?

Cross approximation in d-dimensions

What if a tensor is given as a “black box”?

O., Tyrtyshnikov, 2010:

TT-cross approximation of multidimensional arrays

We can exactly interpolate a rank-r tensor on $\mathcal{O}(dnr^2)$ elements

$$\mathcal{I}_k = (i_1^{(\alpha)}, \dots, i_k^{(\alpha)}),$$

$$\mathcal{J}_k = (i_k^{(\beta)}, \dots, i_d^{(\alpha)})$$

$$A_k = A(\mathcal{I}_k, i_k, \mathcal{J}_{k+1})$$

Making everything a tensor: QTT

- Prequel: E. E. Tyrtyshnikov (2003)
- I. V. Oseledets (2009)
- B. N. Khoromskij (2009)

"Simple" idea: to make everything a tensor (we have software, need examples)

Making everything a tensor: QTT

Let $f(x)$ – function of one variable ($f(x) = \sin x$).

If v – vector of values on a uniform grid with 2^d nodes.

Reshape v into a $2 \times 2 \times ... \times 2$ d-dimensional tensor.

Compute TT-decomposition!

It is a QTT-format

Making everything a tensor: QTT

If $f(x)$ is such that

$$f(x+y) = \sum_{\alpha=1}^{r} u_{\alpha}(x) v_{\alpha}(y),$$

then QTT-ranks are bounded by r

Conclusion:

- $f(x) = \exp(\lambda x)$
- $f(x) = \sin(\alpha x + \beta)$
- $f(x)$ - polynom
- $f(x)$ - Rational function

TT-Toolbox

Software: `http:/github.com/oseledets/TT-Toolbox`

- Basic operations in TT-format
- Advanced operations in TT-format (linear systems, eigenvalues, non-stationary probems, interpolation)
- Main operators
- Open-source
- S. V. Dolgov, V. A. Kazeev, I. V. Oseledets, D. V. Savostyanov, ...

What we can do

TT-Toolbox:

Available both in MATLAB and in Python,

`http://github.com/oseledets/TT-Toolbox`

`http://github.com/oseledets/ttpy`

- We have a very efficient implementation of a

  black-box tensor approximation (cross method)
- We have a very efficient implementation of basic operations
- Several advanced solvers (linear systems, convolution)

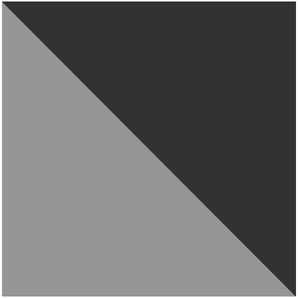
Wavelet tensor train decomposition

The QTT idea seems to be very reasonable for the compression of images

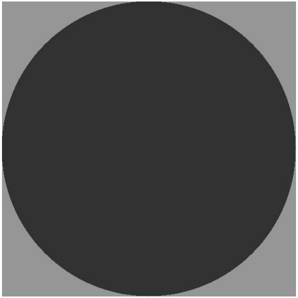
Idea of QTT can be interpreted as a wavelet-type transform!

QTT of simple images

QTT compression: 50 bytes



QTT compression: 70 KBytes (no compression)



WTT idea

The idea of wavelet tensor train (Oseledets, Tyrtyshnikov, 2011):

QTT can be interpreted as an application of filters!

The signal is written as

$y = Ws$, where W is constructed from data, s is (pseudo) sparse

WTT-matrix

V. Kazeev, I. V. Oseledets

Under a suitable permutation of coefficients, the WTT-matrix has small TT-ranks!

Order-r – matrix has TT-ranks $r^2 + 1$

WTT-matrix

V. Kazeev, I. V. Oseledets

Under a suitable permutation of coefficients, the WTT-matrix has small TT-ranks!

Order-r – matrix has TT-ranks $r^2 + 1$

Example: Haar wavelet transform has QTT-rank 2

Compare to FFT

Fast Fourier Transform (FFT) is one of the important signal processing tools.

The FFT matrix has large QTT ranks (D. V. Savostyanov)

But $y = Fx$ can be computed in $\mathcal{O}(\log^\alpha N)$ if the result is in the low-rank format

Compression

- We can apply WTT idea for the compression of large-scale data (images, video, 3D video)
- Simple to use (see the talk by Pavel Khariuk)
- Many questions: the current version uses the SVD to compute filters

Conclusions

- The software is in good shape
- Need to find the limits of applicability

`http://oseledets.github.io` – my Skoltech group webpage

`http://spring.inm.ras.ru/osel` – personal webpage

`http://github.com/oseledets/TT-Toolbox/` – MATLAB code
`http://github.com/oseledets/ttpy/` – Python code