Low-rank approximation of matrices & tensor with application to dynamical and optimization problems

I. V. Oseledets

Skolkovo Institute of Science and Technology

Institute of Numerical Mathematics, Russian Academy of Sciences

17 March 2015

Main points

I will talk about numerics in high dimensions

- High-dimensional problems are hard (curse of dimensionality)

Main points

I will talk about numerics in high dimensions

- High-dimensional problems are hard (curse of dimensionality)
- Fascinating algorithms appear in applications

Main points

I will talk about numerics in high dimensions

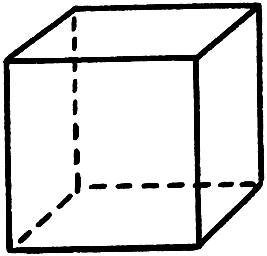
- High-dimensional problems are hard (curse of dimensionality)
- Fascinating algorithms appear in applications
- Generally do not become a universal computational tool (problem-dependent)

Our goal is to create a universal set of tools for high-dimensional problems!

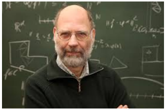
Tensor

- Matrix is a two-dimensional array
- Tensor is a d-dimensional array, $A(i_1, \ldots, i_d)$

Suddenly everything is much more complicated for tensors!



Important "tensor" people (not all!)



W. Hackbusch

R. Schneider

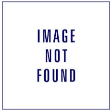
M. Mohlenkamp

D. Savostyanov

E. Tyrtyshnikov

L. Grasedyck

G. Beylkin



S. Dolgov

B. Khoromskij

D. Kressner

C. Lubich

L. De Lathauwer

Reviews

Now there are several books / reviews:

- T. Kolda, B. Bader "Tensor decompositions and applications", SIREV 2009 – outdated
- B. N. Khoromskij,

  Tensors-structured Numerical Methods in Scientific Computing: Survey on Recent Advances (2010)
- L. Grasedyck, D. Kressner, C. Tobler, A literature survey of low-rank tensor approximation techniques (2013)
- W. Hackbusch, Tensor spaces and numerical tensor calculus, Springer, 2012.

Motivation

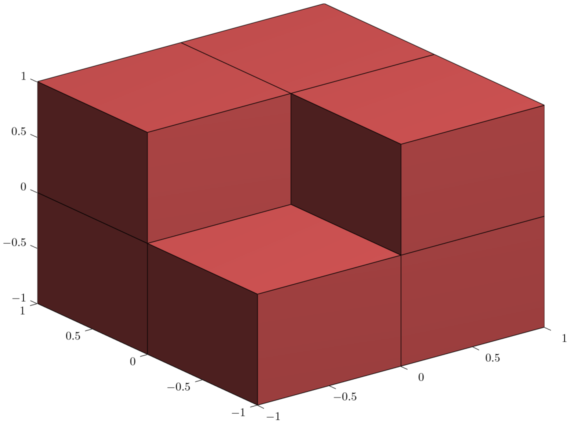
Main point

- High-dimensional problems appear in many applications
- Standard methods do not scale well with d

Motivation

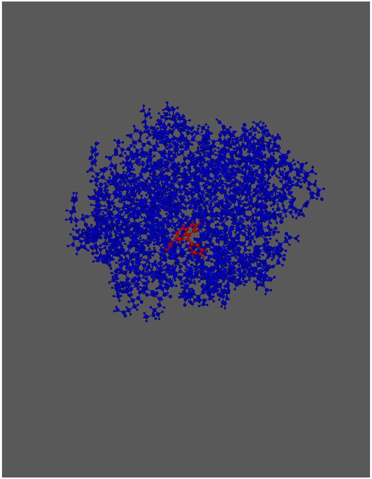
Solving differential / integral equations on fine grids

Typical cost: $\mathcal{O}(N^3) \to \mathcal{O}(N)$ or $\mathcal{O}(\log^\alpha N)$.

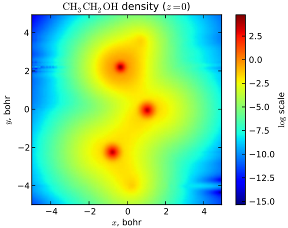


Motivation

Ab initio computations and computational material design
Protein-ligand docking

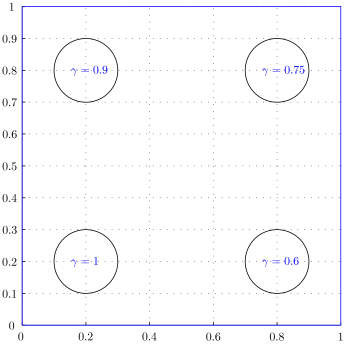


Solving the Hartree-Fock equation (great progress: V. Khoromskaya, B. Khoromskij; the picture is from our HF-solver)



Motivation

Model reduction



Diffusion equation
(Kressner, Tobler)

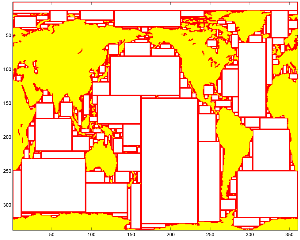
$\nabla a(p)\Delta u = f(p)$,
$p = (p_1, p_2, p_3, p_4)$
Approximate u from few snapshots.

Motivation

Data compression and mining

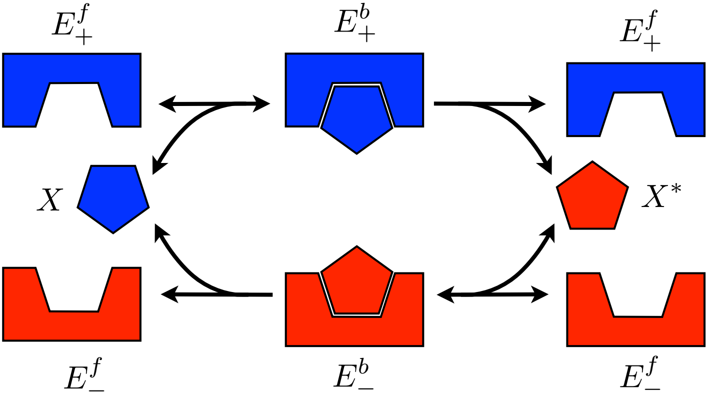
Картинки

Computational data compression



An ad

Biological modelling:
V. Kazeev, M. Khammash, M. Nip., C. Schwab



- tensor method: $10^3 - 10^4$ on a notebook in MATLAB
- 1500 cores, Monte-Carlo: 10^5-sec

Main problem

We need to approximate high-dimensional tensors

Separation of variables

One of the few fruitful ideas is separation of variables

Main task: how to do it numerically?

Canonical format

Starting point: CP-format

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

No robust algorithms: best approximation with fixed rank may not exist!

Everything is good in 2D

2D: $A = UV^\top$, we have the Singular Value Decomposition

We want the methods of such quality in many dimensions!

TT & HT formats

Independently, in 2009 two formats were proposed:

- Tree-Tucker (O. & Tyrtyshnikov) became the Tensor Train;
- HT-format (Hackbusch, Kuhn, Grasedyck).

Both are based on the hierarchical separation of indices

TT-format

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

where $G_k(i_k)$ — matrix of size $r_{k-1} \times r_k$.

Tensor networks, MPS(1)

Other areas:

TT is Matrix product states

(Used to represent spin wavefunctions)

$$H\psi = E\psi$$

$\psi = \psi(S_1, \dots, S_N)$ — spin system

Algorithms (Wilson renormalization group, Density Matrix Renormalization Group) were proposed a lot earlier.
Vidal, Cirac, Verstraete, …

Brought to mathematics by T. Huckle and R. Schneider

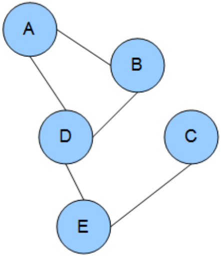
Tensor networks, MPS(2)

DMRG, MPS, tensor networks:

Big community, brilliant algorithms for eigenvalues / time-dependent problems / eigenvalue problems

Markov random fields

Markov random fields (wiki picture)



Edge corresponds to a function ψ_{AD},

$$p(A, B, C, D, E) = \psi_{AD}\psi_{AB}\psi_{DE}\psi_{CE}$$

Algorithm: belief propagation for trees!

Recent successes

Linear tree $\rightarrow$ hidden markov models

Spectral methods for learning HMM (Hsu, Kakade, 2009)
are based on the singular value decomposition

Definition

Tensor is said in the TT-format, if

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

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

- 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

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

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

Fast linear algebra

Addition, Hadamard product, scalar product
All linear in d

Fast linear algebra

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

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

ranks are multiplied

Tensor rounding

A is given in TT-format with suboptimal ranks.
Who to reapproximate?

Tensor rounding

It can be done in $\mathcal{O}(\mathrm{dnr}^3)$ 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 on $\mathcal{O}(dnr^2)$ elements

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

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

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

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 $2 \times 2 \times \ldots \times 2$ 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, …

Applications and main problems(1)

High-dimensional linear systems:

$$Ax = f, \; x = X(i_1, \ldots, i_d)$$

Typical cases:

- High-dimensional PDE on a tensor-product grid (Chemical master equation, Fokker-Planck equation)
- Parametric / stochastic PDE:

  $A(p)u(p) = f(p), \; p = (p_1, \ldots, p_m),$

  After discretization:

  $u = u(i, p_1, \ldots, p_M)$ — a tensor!

Applications and main problems (2)

High-dimensional eigenvalue problems:

$$Ax = \lambda x, \; x = X(i_1, \dots, i_d)$$

Typical cases:

- Spin systems (classical case, where MPS come from)
- Vibrational computations, $A = -\frac{1}{2}\Delta + V$
- Parametric problems (as well).

Applications and main problems (3)

High-dimensional unsteady problems:

$$\frac{dy}{dt} = Ay, y = Y(i_1, \ldots, i_d)$$

Typical cases:

- Chemical master equation
- Computation of vibrational spectra

Applications and main problems (4)

Interpolation of multivariate functions:

$f(x_1, \ldots, x_d)$ is given as a subroutine

Typical cases:

- Global optimization problems
- Approximation of expensive parametric dependencies
- Many more…

Summary

Several basic problems:

- $Ax = f$
- $Ax = \lambda x$
- $\frac{dy}{dt} = Ay$
- Interpolation

The solution is sought on a low-parametric manifold:

General strategy:

Reformulate as $J(x) \to \min$, minimize over a manifold.

Summary(2)

There are very efficient algorithms for all type of problems!

- Linear systems: AMEN-solver (Dolgov, Savostyanov)
- Eigenvalue solver: AMEN-solver, EIGB-solver (Dolgov, Savostyanov, Oseledets, Khoromskij)
- Nonstationary case: KSL-scheme (Oseledets, Lubich, Vanderbreycken)
- Interpolation: AMEN-cross (Dolgov, Savostyanov, Oseledets)

Solving non-stationary problems

Considerable interest:

$$\frac{dy}{dt} = Ay,$$

$$Y = Y(i_1, \ldots, i_d)$$

By writing down the equations for the parameters on the manifold!

We now have a very efficient integrator:

KSL-scheme

Dynamical low-rank approximation

Given $A(t)$, approximate by $X(t) \in \mathcal{M}$,

where $\mathcal{M}$ — manifold:

Dirac-Frenkel principle:

$$(\dot{A} - \dot{X}, v) = 0, \quad v \in \mathcal{T}(\mathcal{M}),$$

$\mathcal{T}$ is the tangent space.

Gives equations of motion

KSL-scheme for the TT-format

Equation of motions have been derived:

- Matrix case, Tucker case: (H.-D. Meyer, C. Lubich, O. Koch)
- TT-format, HT-format (C)

Matrix case

Matrix case

C. Lubich, I.V. Oseledets, A projector-splitting integrator for dynamical low-rank approximation

Dynamical low-rank appr. of matrices

The equations for U, S, V:

$$\dot{U} = (I - U(t)U(t)^{\top})\dot{A}(t)V(t)S(t)^{-1}$$
$$\dot{V} = (I - V(t)V(t)^{\top})\dot{A}(t)^{\top}U(t)S(t)^{-\top}$$
$$\dot{S} = U(t)^{\top}\dot{A}(t)V(t).$$

Dynamical low-rank appr. of matrices

$$\dot{X} = P_X(\dot{A}), \quad P_x(\dot{A}) = \dot{A} - (I - UU^\top)\dot{A}(I - VV^\top).$$

No multiplication by S^{-1}

KSL integrator

Algorithm:

- K-step: $(\dot{U}S) = \dot{A}V$
- QR: $K_1 = U_1\widehat{S}_1$
- S-step: $\dot{S} = -U^\top \dot{A}V$ (backward in time!)
- L-step: $(V\dot{S}^\top) = \dot{A}^\top U$
- QR: $L_1 = U_1\widetilde{S}_1$

TT-KSL integrator

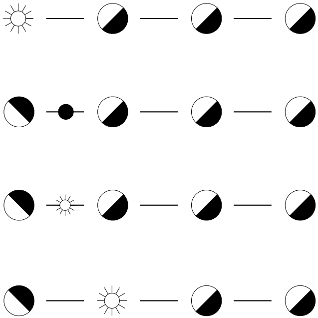
Just apply the KSL scheme recursively!

Update X_1

QR $\rightarrow$

Update S

Update X_2



KSL and MCTDH

$$\frac{d\psi}{dt} = iH\psi, \quad \psi(0) = \psi_0$$

$$H = -\tfrac{1}{2}\Delta + V,$$

Local problems:

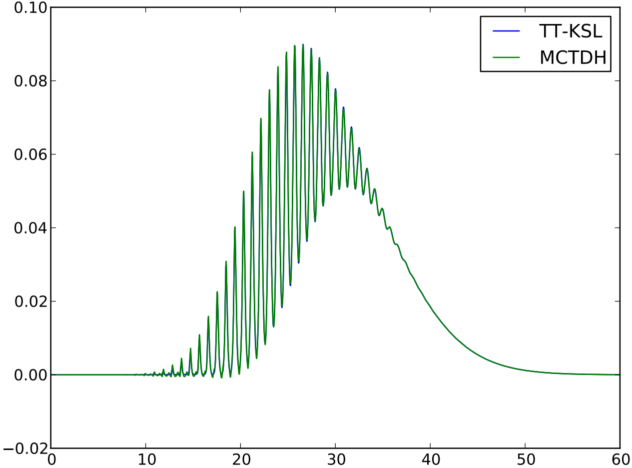
Small linear ODEs

Compute $a(t) = (\psi(t), \psi(0))$ and the spectrum of H from it.

KSL and MCTDH

$$V(q_1, \ldots, q_f) = \frac{1}{2}\sum_{k=1}^{f} q_k^2 + \lambda \sum_{k=1}^{f-1}\left(q_k^2 q_{k+1} - \frac{1}{3} q_k^3\right).$$

http://www.pci.uni-heidelberg.de/cms/mctdh.html



Relation to wavelets

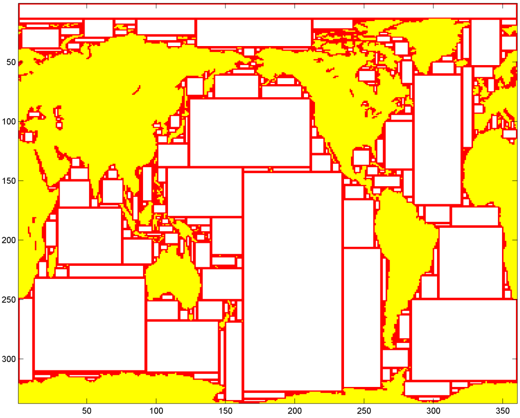
The idea of QTT has a deep connection to wavelets

- I. V. Oseledets, E. E. Tyrtyshnikov, Algebraic wavelet transform via quantics tensor train decomposition
- V. A. Kazeev, Oseledets, I. V. , The tensor structure of a class of adaptive algebraic wavelet transforms
- Boris N. Khoromskij, Sentao Miao, Superfast Wavelet Transform Using QTT Approximation. I: Haar Wavelets

You can use WTT as a general compression technique!

Ocean temperature

The temperature (4-d array), computed using the INM-RAS global circulation model
Array of size $360 \times 337 \times 40 \times 648$ — 12 Gb.



Ocean temperature

Memory	Abs err	Rel err	Comp time
497 MB	0.0392	0.0004	≈ 500 sec
277 MB	0.0984	0.0009	≈ 500 sec

Table: WTT decomposition compression

Latent variable models

Interesting applications

latent variable models

$$p(x_1, x_2) = \sum_{\alpha=1}^{r} p_1(x_1, h) w(h) p_2(x_2, h)$$

You can use tensors! (Ishteva, Le Song, Georgia Tech.)

Latent variable models

Interesting applications

latent variable models

Observe $S_1, \ldots, S_N$ (stock prices)

And here are the hidden variables

$p(x_1, x_2) = \sum_{\alpha=1}^{r} p_1(x_1, h)w(h)p_2(x_2, h)$

You can use tensors! (Ishteva, Le Song, Georgia Tech.)

Latent variable models

Interesting applications

latent variable models

Observe $S_1, \ldots, S_N$ (stock prices)

And here are the hidden variables

$$p(x_1, x_2) = \sum_{\alpha=1}^{r} p_1(x_1, h) w(h) p_2(x_2, h)$$

You can use tensors! (Ishteva, Le Song, Georgia Tech.)

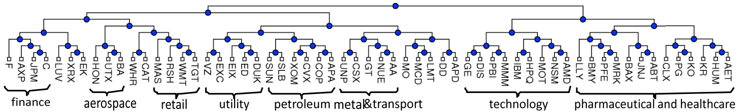
Latent variable models

$$p(x_1, x_2) = \sum_{\alpha=1}^{r} p_1(x_1, h) w(h) p_2(x_2, h)$$

You can use tensors! (Ishteva, Le Song, Georgia Tech.)

Recovering the tree

(M. Ishteva, Le Song)



Global optimization

Can we apply it to the global optimization problems?

$$f(x_1, \dots, x_d) \to \min$$

"Naive" idea:

1. Approximate f by low rank
2. Find maximum, for example, by $\min(Dx, x) \to \min$

What if no approximation exists?

Global optimization

The cross approximation method has a potential to find maximal absolute value!

Global optimization

The cross approximation method has a potential to find maximal absolute value!

Theorem

Let A be an $n \times m$ matrix, $\widehat{A}$ is an $r \times r$ submatrix with maximal volume, then

$$\|\widehat{A}\|_C \geq \frac{\|A\|_C}{r^2 + r}.$$

Global optimization

To force to the global minimum, we do shifts and transforms:

$$\widetilde{f} = \mathrm{arcctg}(f - f^*),$$

where f^* is the current record.

{Just run the standard dD-cross method, and compute maximal over all the samples!}

Conclusions

- Numerical algorithms are developing at fast rate
- High potential impact in many applications (biology, optimization, chemistry)
- Theory is trailing behind

Software

Papers and codes:

- My webpage: http://spring.inm.ras.ru/osel
- Publications: http://pub.inm.ras.ru
- TT-Toolbox
  http://github.com/oseledets/TT-Toolbox,
  http://github.com/oseledets/ttpy