Tensor Based Algorithms

Eugene Tyrtyshnikov

Marchuk Institute of Numerical Mathematics
Russian Academy of Sciences

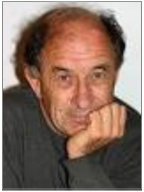
eugene.tyrtyshnikov@gmail.com

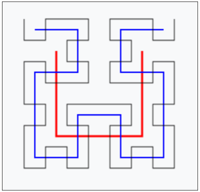
Moscow Institute of Physics and Technology
27 September 2019

Multi-variate reduce to few-variate

13th Hilbert's problem: *represent a d-variate continuous function by a superposition of 2-variate continuous functions.* The question was motivated by Galois theory for algebraic equations and Tschirnhaus transformation deleting 3 terms next to the senior one. V.I.Arnold and A.N.Kolmogorov proved that it is possible to use uni-variate functions and just one 2-variate one which is simply addition:

$$f(x_1, \ldots, x_d) = \sum_{q=0}^{2d} \Phi_q \left(\sum_{p=1}^{d} \psi_{qp}(x_p) \right).$$



Separation of variables and decompositions

$$a(i,j) = u(i)v(j) \quad \Leftrightarrow \quad A = uv^\top \quad \Leftrightarrow \quad a = u \otimes v$$

$$a(i,j) = \sum_{\alpha=1}^{r} u_\alpha(i)v_\alpha(j) \quad \Leftrightarrow \quad A = \sum_{\alpha=1}^{r} u_\alpha v_\alpha^\top \quad \Leftrightarrow \quad a = \sum_{\alpha=1}^{r} u_\alpha \otimes v_\alpha$$

Minimal number of summands is equal to rank of A.

Tensor Train Decomposition

Skeleton (dyadic) decomposition:

$$A(i_1, i_2) = G^1_{i_1} G^2_{i_2}, \quad G^1_{i_1} \in \mathbb{R}^{1\times r}, \quad G^2_{i_2} \in \mathbb{R}^{r\times 1}$$

$$1 \leqslant i_1 \leqslant n_1, \quad 1 \leqslant i_2 \leqslant n_2$$

Tensor train:

$$A(i_1, \dots, i_d) = G^1_{i_1} G^2_{i_2} \dots G^{d-1}_{i_{d-1}} G^d_{i_d}$$

$$G^1_{i_1} \in \mathbb{R}^{1\times r_1}, \quad G^2_{i_2} \in \mathbb{R}^{r_1\times r_2}, \quad \dots, \quad G^{d-1}_{i_{d-1}} \in \mathbb{R}^{r_{d-2}\times r_{d-1}}, \quad G^d_{i_d} \in \mathbb{R}^{r_{d-1}\times 1}$$

$$1 \leqslant i_1 \leqslant n_1, \quad \dots, \quad 1 \leqslant i_d \leqslant n_d$$

Tensor train in d dimensions

$$a(i_1 \dots i_d) =$$

$$\sum g_1(i_1 \alpha_1) g_2(\alpha_1 i_2 \alpha_2) \dots$$

$$\dots g_{d-1}(\alpha_{d-2} i_{d-1} \alpha_{d-1}) g_d(\alpha_{d-1} i_d)$$

d-tensor reduces to 3-tensors $g_k(\alpha_{k-1} i_k \alpha_k)$.

If the maximal size is $r \times n \times r$ then
the number of tensor-train elements does not exceed

$$dnr^2 \ll n^d.$$

Our class of tensor

$$A_k = [a(i_1 \dots i_k \,;\, i_{k+1} \dots i_d)] =$$

$$\left[\sum u_k(i_1 \dots i_k \,;\, \alpha_k)\ v_k(\alpha_k \,;\, i_{k+1} \dots i_d)\right] = U_k V_k^\top$$

$$u_k(i_1 \dots i_k \alpha_k) = \sum g_1(i_1 \alpha_1) \dots g_k(\alpha_{k-1} i_k \alpha_k)$$

$$v_k(\alpha_k i_{k+1} \dots i_d) = \sum g_{k+1}(\alpha_k i_{k+1} \alpha_{k+1}) \dots g_d(\alpha_{k-1} i_d)$$

THE MAIN PROPERTY OF THE CLASS:
all matrices A_k must be close to low-rank matrices
(I.Oseledets-E.Tyrtyshnikov)

Fast summation of elements of an astronomically huge vector

$$i = \overline{i_1 i_2 \dots i_d} \qquad d = 83$$

$$a(i) = a(i_1, \dots, i_d) = \sum_{\alpha_1, \dots, \alpha_{d-1}} g_1(i_1, \alpha_1) g_2(\alpha_1, i_2, \alpha_2) \dots g_d(\alpha_{d-1}, i_d)$$

$$\sum_{i_1, \dots, i_d} a(i_1, \dots, i_d) = \sum_{\alpha_1, \dots, \alpha_{d-1}} \hat{g}_1(\alpha_1) \hat{g}_2(\alpha_1, \alpha_2) \dots \hat{g}_d(\alpha_{d-1})$$

$$\hat{g}_k = \sum_{i_k} g_k$$

Tensor train integrator

Compute a d-dimensional integral

$$I(d) = \int \sin(x_1 + x_2 + \ldots + x_d)\ dx_1 dx_2 \ldots dx_d =$$

$$\mathrm{Im} \int_{[0,1]^d} e^{i(x_1 + x_2 + \ldots + x_d)}\ dx_1 dx_2 \ldots dx_d = \mathrm{Im}((\frac{e^i - 1}{i})^d).$$

$n = 11$ nodes along each dimension $\Rightarrow$ in total n^d values! Only a very small part of them in needed for the constrcution of TT.

d	$I(d)$	Relative error	Time
1000	-2.637513e-19	3.482065e-11	11.60
2000	2.628834e-37	8.905594e-12	33.05
4000	9.400335e-74	2.284085e-10	105.49

Tensor train truncation

$$g_1 g_2 g_3 \underbrace{g_4}_{r_4 q_4} = g_1 g_2 \underbrace{g_3 r_4}_{r_3 q_3} q_4$$

$$= g_1 \underbrace{g_2 r_3}_{r_2 q_2} q_3 q_4 = \underbrace{g_1 r_2}_{u_1 \sigma_1 v_1} q_2 q_3 q_4$$

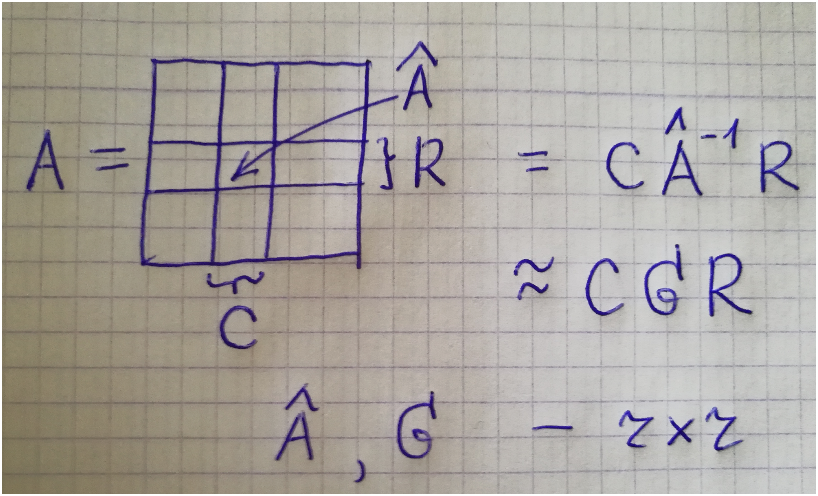
$$= u_1 \underbrace{\sigma_1}_{\approx \tilde{\sigma}_1} v_1 q_2 q_3 q_4 \approx u_1 \underbrace{\tilde{\sigma}_1 v_1 q_2}_{u_2 \sigma_2 v_2} q_3 q_4$$

$$\approx u_1 u_2 \underbrace{\tilde{\sigma}_2 v_2 q_3}_{u_3 \sigma_3 v_3} q_4 \approx u_1 u_2 u_3 \tilde{\sigma}_3 v_3 q_4$$

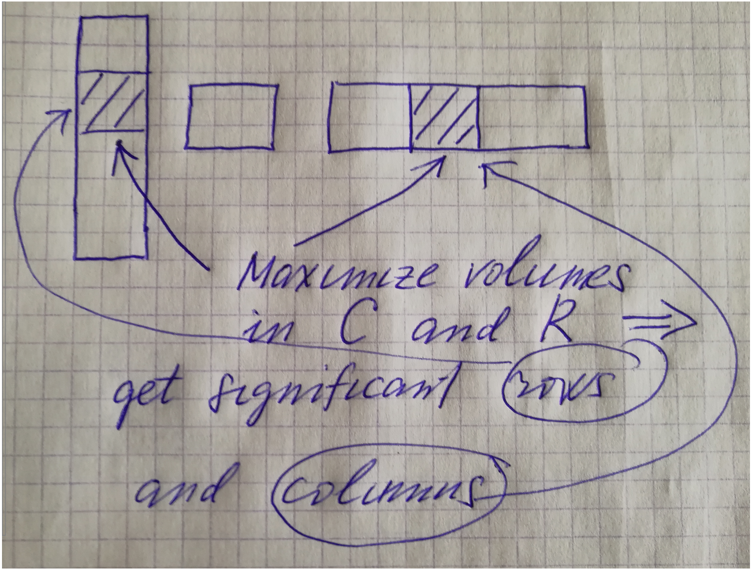
25 years of pseudo-skeleton approximation

- S. Goreinov, E. Tyrtyshnikov, N. Zamarashkin, *A theory of pseudo-skeleton approximations*, Linear Algebra Appl. 261 (1997) 1–21.
- A. Osinsky, N. Zamarashkin, *Pseudo-skeleton approximations with better accuracy estimates*, Linear Algebra Appl. 537 (2018) 221–249.
- A. Osinsky, *Probabilistic estimation of the rank 1 cross approximation accuracy*, arXiv:1706.10285 (2017).

Low-rank approximation using only most significant columns and rows



How to find a good cross?



How to maximize the volume

Let $\hat{C} \in \mathbb{R}^{k \times k}$ match the rows in $C \in \mathbb{R}^{n \times k}$. Then consider

$$C\hat{C}^{-1} = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ q_{r+1,1} & \cdots & q_{r+1,r} \\ \cdots & \cdots & \cdots \\ q_{n1} & \cdots & q_{nr} \end{bmatrix}$$

Necessary for $\hat{C}$ be of maximal volume in C:

$$|q_{ij}| \leqslant 1, \quad r+1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant r$$

Otherwise, swapping the rows increases the volume!

(related to [D.Knuth, Semi-optimal bases for linear dependencies, Linear and Mulilinear Algebra, 1985])

Maximal volume concept

Theorem (Goreinov, Tyrtyshnikov' 2000).

Let $A_{11} \in \mathbb{C}^{r \times r}$ be of maximal volume among all $r \times r$ blocks in the matrix

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}.$$

Then

$$||A - CA_{11}^{-1}R||_C \leqslant (r+1)||F||_2,$$

$$C = \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}, \quad R = \begin{bmatrix} A_{11} & A_{12} \end{bmatrix}.$$

Using more columns and rows

Can we obtain a better rank-r approximation using more columns and rows? And how much better?

Theorem (N.Zamarashkin, A.Osinsky'2017).

Assume that $A_{11} \in \mathbb{C}^{p \times q}$ is of rank not smaller than r and has the maximal r-projective volume

$$\mathcal{V}(A_{11}) := \prod_{i=1}^{r} \sigma_i(A)$$

among all $p \times q$ submatrices in A. Then

$$||A - C(A_{11})_r^{\dagger} R||_C \leqslant \sqrt{1 + \frac{r}{p - r + 1}} \sqrt{1 + \frac{r}{q - r + 1}}\, ||F||_2.$$

Recent Frobenius norm estimate

Theorem (N.Zamarashkin, A.Osinsky'2017).

There exist r columns and r rows providing the estimate

$$||A - C\hat{A}^{-1}R||_F \leqslant (r+1)\,||F||_F,$$

where $\hat{A} \in \mathbb{C}^{r\times r}$ is the intersection matrix for C and R.

Maximal volume is not everything

Note that $\hat{A}$ may not be of maximal volume!

Let $0 < \delta \ll \varepsilon \ll 1$ and consider an $n \times 2$ magtrix

$$A = \begin{bmatrix} 1+\delta & 1 & 1 & \cdots & 1 \\ \varepsilon & 0 & 0 & \cdots & 0 \end{bmatrix}^T .$$

The accuracy of best rank-1 approximation does not exceed ε. However, by a direct calculation we have

$$\|A - CA_{11}^{-1}R\|_F \geq \sqrt{n-1}\varepsilon.$$

How to find a good cross?

In a given matrix A, consider a column submatrix $C \in \mathbb{R}^{n \times k}$ and find significant k rows. Note relations with compressed sensing and restricted isometry property.

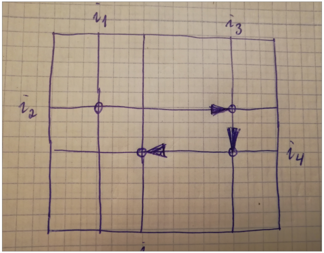
These rows determine a row submatrix $R \in \mathbb{R}^{k \times n}$ in which we find significant k columns.

These new columns produce another column submatrix $C \in \mathbb{R}^{k \times n}$ in which we find significant k rows.

Repeat this way a few times.

Chasing big entries

In a given matrix A, consider a column submatrix $C \in \mathbb{R}^{n \times k}$ and find significant k rows. Note relations with compressed sensing and restricted isometry property.



These rows determine a row submatrix $R \in \mathbb{R}^{k \times n}$ in which we find significant k columns.

These new columns produce another column submatrix $C \in \mathbb{R}^{k \times n}$ in which we find significant k rows.

Repeat this way a few times.

Chasing big entries, but how big?

Lemma.

Assume that

$$A = \sigma u v^\top + E, \quad ||E|| = \varepsilon\sigma ||uv^\top|| = \varepsilon\sigma ||u|| ||v||,$$

$$|a_{ij}| = \max_k |a_{ik}|, \quad |u_i| = \mu_1 ||u||, \quad |v_j| = \mu_2 ||v||.$$

Then

$$\boldsymbol{\mu_2 \geqslant 1 - \frac{2\varepsilon}{\mu_1}.}$$

Proof.

$$\mu_1 \sigma ||u|| ||v|| - ||E|| \ \leqslant \ |a_{ij}| \ \leqslant \ \mu_1\mu_2\sigma ||u|| ||v|| + ||E|| \quad \Rightarrow$$

$$\mu_1 - 2\varepsilon \leqslant \mu_1 \mu_2$$

Just in four steps

Consider indices i_1, i_2, i_3, i_4 and let

$$|u_{i_1}| = \mu_1 ||u||, \quad |v_{i_2}| = \mu_2 ||v||, \quad |u_{i_3}| = \mu_3 ||u||, \quad |v_{i_4}| = \mu_4 ||v||.$$

- If we know that $\mu_1 \geqslant 4\varepsilon$, then
- $\mu_2 \ \geqslant \ 1 - \frac{2\varepsilon}{4\varepsilon} = \frac{1}{2}$,
- $\mu_3 \ \geqslant \ 1 - \frac{2\varepsilon}{1/2} = 1 - 4\varepsilon$,
- $\mu_4 \ \geqslant 1 - \frac{2\varepsilon}{1-4\varepsilon}$, provided that $\varepsilon \ < \ \frac{1}{4}$.

$$\Rightarrow \qquad |a_{i_3 i_4}| \ \geqslant \ (1 - 6\varepsilon) ||A - E||$$

It is guaranteed if $\varepsilon < \frac{1}{6}$ and we start with a *good column* for which

$$\mu_1 \geqslant 4\varepsilon.$$

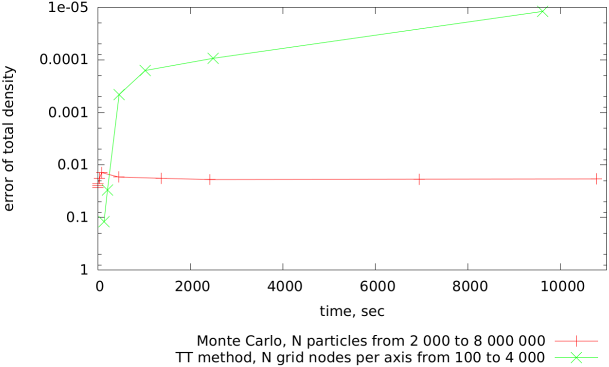
Now we can calculate probabilities

`Theorem  (A.Osinsky, N.Zamarashkin, D. Zheltkov).`

Let the vector v be randomly picked up from a unit sphere and $\varepsilon = O(n^{-1/2}||v||_C^{-1})$. Then the maximal in modulus entry in randomly chosen k columns of A contains a good column with probability larger than

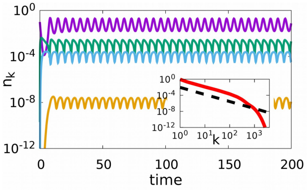
$$1 - O(n^{-ck/\log n}), \quad c > 0.$$

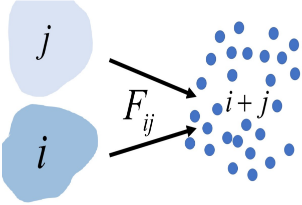
Tensor train for Smoluchowski equations

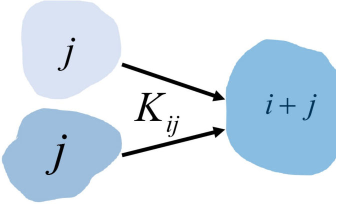


Ballistic kernel:

$$K(u,v) = (u^{\frac{1}{3}} + v^{\frac{1}{3}})^2 \sqrt{\frac{1}{u} + \frac{1}{v}}.$$





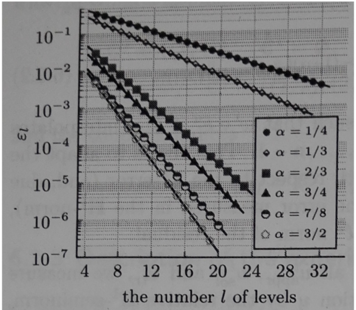


Tensor train PDE solvers

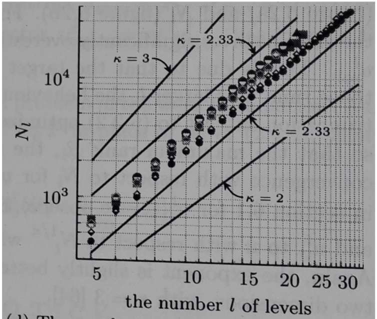
$$u_\Gamma(x) = r^\alpha \sin \alpha\phi(x), \quad x \in \Omega = (0,1)^2$$

$$\varepsilon_l \leqslant \exp\{-cN_l^{\frac{1}{\kappa}}\}, \quad N_l \text{ – the number of TT-elements}$$

THEOREM (V.Kazeev & C.Schwab). $\kappa \leqslant 5$.



(a) Convergence with respect to l. The reference lines correspond to the exponential convergence $\varepsilon_l = C_\alpha 2^{-\tilde{\alpha} l}$ with C_α independent of l and with $\tilde{\alpha} = \min\{\alpha, 1\}$.



(d) The number N_l (3.3.2) of QTT parameters vs. l. The reference lines correspond to the algebraic growth $N_l = C_\alpha l^\kappa$ with κ and C_α independent of l.

Global optimization using low rank approximations

Given a tensor $A = [a_{i_1 \dots i_d}]$, consider its associated matrices (appeared in [Tyrtyshnikov, Sbornik Math., 2003])

$$A_k = [a_{i_1 \dots i_k \, ; \, i_{k+1} \dots i_d}], \quad 1 \leqslant k \leqslant d-1,$$

and apply the cross algorithm to find r significant rows and columns in each of them. Then find the biggest entry over all those that were used by the algorithm.

NB. The matrix cross algorithm cannot be applied directly to A_k since too many entries must have been taken!

Approximations with nonnegativity

Two possible settings:

- Approximation (not factors) is required to be nonegative.

  Without scanning all entries not easy even to check!

- Each factor of the approximation must be nonnegative.

Notion of r-separability

A real matrix A is called r-separable if there is a `defining system` of r columns of A such that each column of A belongs to the cone spanned by this system.

$$A = WH, \quad A \in \mathbb{R}^{m\times n}, \quad W \in \mathbb{R}^{m\times r}, \quad H \in \mathbb{R}^{r\times n}$$

H is nonnegative and contains the identity block of size $r \times r$.

`Additional assumptions:`

- The columns of W are linear independent (r-separable of rank r).
- All elements in each column of H sum up to at most 1 (normalized r-separable).

How to find a defining system

- Maximal length column belongs to a defining system. (Just watch a triangle.)
- Modify each column subtracting the orthogonal projection onto the maximal one. Maximal length column corresponds to another vector in the defining sustem.
- And so on.

These observations lead to a simple procedure suggested by Gillis–Vavasis.

Do we still need to work with all elements of A?

If we know a skeleton decomposition

Assume that we already have

$$A = UV,$$

where U and V may have negative entries.

Using only U and V, we can propose a *reduced algorithm* for nonnegative factorization of A with much smaller complexity than the GV-algorithm.

First step is normalization

Let $A = UV$ be a nonnegative r-separable matrix of rank r and d_j be the sum of entries of the column j of A.
Then

$$d = [d_1, \dots, d_n] = (eU)V, \quad e = [1, \dots, 1],$$

can be computed with $2(m+n)r$ operations.

Set $D := \mathrm{diag}(d_1, \dots, d_n)$ and any zero value reset to 1.

Then $\widetilde{A} = AD^{-1}$ is a normalized r-separable matrix.

Further steps of the reduced algorithm

No need to have all entries of $\widetilde{A} = UVD^{-1}$:

- $\widetilde{V} = VD^{-1}$ is a normalized r-separable of rank r.
- Defining columns of $\widetilde{V}$ correspond to defining columns of the original matrix A.

THEOREM.

Let A be a nonnegative r-separable marix of rank r.
If we know some skeleton decompostion

$$A = UV, \quad U \in \mathbb{R}^{m \times r}, \quad V \in \mathbb{R}^{r \times n},$$

then a nonnegative factorization can be computed with the complexity linearly depending on $m + n$.

Lemma for the analysis of small perturbations

LEMMA. *Given arbitrary vectors $a_1, \dots, a_k$ and their convex combination with the coefficients $\alpha_1, \dots, \alpha_k$, we have the identity*

$$||\sum_{i=1}^{k} \alpha_i a_i||^2 = \sum_{i=1}^{k} \alpha_i ||a_i||^2 - \sum_{1 \leqslant i < j \leqslant k} \alpha_i \alpha_j ||a_i - a_j||^2.$$

COROLLARY. *For an arbitrary convex combination of the given vectors it holds*

$$||\sum_{i=1}^{k} \alpha_i a_i||^2 \quad \leqslant \quad \mu^2 \; - \; \left(\sum_{1 \leqslant i < j \leqslant k} \alpha_i \alpha_j \right) \omega^2,$$

$$\mu = \max_{1 \leqslant i \leqslant k} ||a_i||, \quad \omega = \min_{1 \leqslant i < j \leqslant k} ||a_i - a_j||.$$

One more corollary

COROLLARY. Suppose that in ε-vicinity of a convex combination vector

$$b = \sum_{i=1}^{k} \alpha_i a_i$$

there is a vector with the length not smaller than μ, and let a_j be a vector with the largest coefficient α_j. Then for all sufficiently small ε it holds

$$1 - \alpha_j = O\left(\frac{\mu}{\omega^2}\varepsilon\right),$$

$$||b - a_j|| = O\left(\frac{\mu^2}{\omega^2}\varepsilon\right).$$

Influence of small perturbations

THEOREM. Let $A \in \mathbb{R}^{m\times n}$ be a normalized r-separable matrix of rank r, and assume that GV-algorithm is applied to a perturbed matrix $\tilde{A} = A + E$, where the length of each column of the perturbation E does not exceed ε, and produces the columns $\widetilde{a}_{i_1}, \dots, \widetilde{a}_{i_r}$. Then the defining columns of A can be ordered so that for all sufficiently small ε there hold the inequalities

$$||\widetilde{a}_{i_k} - a_{j_k}|| \leqslant \left(1 + c\frac{\mu^2}{\sigma_r^2}\right)\varepsilon, \quad k = 1, \dots, r,$$

where μ is the largest column lenght of A, σ_r is the minimal singular value of the $m \times r$ matrix comprised of the defining columns of A, c is a positive constant.

Nonnegative Matrix Factorization methods

- Multiplicative Update Rules (Lee and Seung, 1999)
- Alternating Least Squares (Berry et al, 2006)
- Alternating Nonnegative Least Squares
  - Projected Gradient Descent (Lin, 2007)
  - Quasi-Newton (D. Kim et al, 2007)
  - Active-set (J. Kim and Park, 2008)
- Hierarchical ALS (Cichocki et al, 2007)
- Rank-one residue iteration (Ho, 2008)
- ...

Zhou, G., Cichocki, A., Xie, S.: Fast Nonnegative Matrix/Tensor Factorization Based on Low-Rank Approximation. IEEE Transactions on Signal Processing **60**(6), 2928–2940 (2012)

Nonnegativity through decompositions without nonnegativity

Minimization of $||A - WH||_F$ under nonnegativity constrains for W and H leads to the equations

$$W \circ (WHH^T) = W \circ (VH^T), \quad H \circ (W^T WH) = H \circ (W^T V).$$

Lee–Seung update:

$$W \leftarrow W \circ \frac{[VH^T]}{[WHH^T]}, \quad H \leftarrow H \circ \frac{[W^T V]}{[W^T WH]},$$

Frobenius norm does not increase.

Using UV-decomposition for A we perform updates in a fast way!

Nonnegativity in Tensor-Train factors reduces to Nonnegative Matrix Factorizaions

- $A(i_1; i_2 \dots i_d) = \sum\limits_{\alpha_1} G_1(i_1; \alpha_1) A_1(\alpha_1; i_2 \dots i_d)$

- $A_1(\alpha_1 i_2; i_3 \dots i_d) = \sum\limits_{\alpha_2} G_2(\alpha_1 i_2; \alpha_2) A_2(\alpha_2; i_3 \dots i_d)$

- $A_2(\alpha_2 i_3; i_4 \dots i_d) = \sum\limits_{\alpha_3} G_3(\alpha_2 i_3; \alpha_3) A_3(\alpha_3; i_4 \dots i_d)$

- And so on.

Example with Smoluchowski

Assuming that the particles are uniformly distributed in the space and may have only pairwise collisions we get the Smoluchowski coagulation equation:

$$\frac{\partial n(v,t)}{\partial t} = \frac{1}{2}\int_0^v K(v-u,u)n(v-u,t)n(u,t)du -$$

$$-n(v,t)\int_0^\infty K(v,u)n(u,t)du.$$

This equation describes the time-evolution of the concentration function $n(v,t)$ of the particles of size v per the unit volume of the system at the moment t.

$$K(\overline{u},\overline{v}) = (u_1+u_2)^\mu (v_1+v_2)^\nu + (u_1+u_2)^\nu (v_1+v_2)^\mu,$$

$$u_i \geq 0,\ i=\overline{1,2},\ v_i \geq 0,\ i=\overline{1,2},\ \mu+\nu \leq 1,\ |\mu-\nu| \leq 2.$$

Variables $u_i,\ v_i,\ i=\overline{1,2}$ take values from 0.1 to 10 with step 0.1. Hence, we get a fourth-order tensor with 10^8 elements.

Results for Smoluchowski

For each set of μ, ν we repeated our experiments 10 times. The results for this test are presented in table 2.
As we can see, even in the worst case for nonnegative tensor train we have to store less than one percent of original amount of values. At the same time, the achieved accuracy is good and the CPU time is acceptable for the tensors of this size.

μ	ν	Relative error(%)	Time($s, \pm std$)	Average ranks	Min ranks	Max ranks
0.1	-0.05	0.14	55.8 ± 4.5	$2, 21, 22$	$2, 2, 2$	$2, 96, 100$
0.2	0.1	0.12	63.2 ± 2.8	$3, 25, 41$	$3, 1, 3$	$3, 85, 100$

Table: NTTF (algorithm 1) to $K(\overline{u}, \overline{v})$, size: $100 \times 100 \times 100 \times 100$

DMRG-like strategy for selecting nonnegative-factorization ranks

Method	*Size*	Relative error(%)	Time($s, \pm std$)	Average ranks	Min ranks	Max ranks
$NTTF$	$10 \times 10 \times 10 \times 10$	1.6	0.12 ± 0.01	$5, 5, 5$	$4, 3, 4$	$5, 5, 5$
Algorithm 2	$10 \times 10 \times 10 \times 10$	3.5	0.14 ± 0.01	$4, 4, 4$	$3, 2, 3$	$5, 5, 5$
$NTTF$	$100 \times 100 \times 10 \times 10$	1.6	0.5 ± 0.04	$5, 5, 5$	$5, 5, 5$	$5, 5, 7$
Algorithm 2	$100 \times 100 \times 10 \times 10$	3.5	0.3 ± 0.03	$6, 6, 5$	$6, 5, 4$	$6, 6, 5$
$NTTF$	$100 \times 100 \times 100 \times 100$	1.7	73.6 ± 2.16	$5, 5, 7$	$5, 5, 5$	$5, 5, 12$
Algorithm 2	$100 \times 100 \times 100 \times 100$	3.7	4.4 ± 0.16	$5, 5, 5$	$5, 4, 5$	$5, 5, 5$

Table 1: NTTF and algorithm 2 to positive tensors with nonnegative TT ranks $r_1 = r_2 = r_3 = 5$.

Thank you for your attention!