Optimization and tensor decompositions

Eugene Tyrtyshnikov

Marchuk Institute of Numerical Mathematics
Russian Academy of Sciences

Lomonosov Moscow State University

Moscow Institute of Physics and Tecnology

tee@inm.ras.ru

ICOMP-2024, Innopolis

Why do we need data models?

An array $A = [a(i_1, \ldots, i_d)]$ of size $n \times \ldots \times n$ cannot be given by the list of all entries even for modest values of d.

If $d = 83$ and $n = 10$, then the number of entries 10^{83} equals the number of atoms in the Universe!

We need *data representation models* with small sets of parameters and *algorithms dealing only with these small sets*.

A data-model paradigm of computations

- "Big objects" are represented through "small vectors"
  $A = A(p), \quad B = B(q), \quad C = C(s).$

- Computing a "big object" $C = A * B$ is done by a *fast algorithm* that finds s from p and q. "Big objects" A and B never arise explicitly.

- Data models for a wide class of applications can be based on low-rank matrices.

Low-rank matrix model

If A is a matrix of rank r, then

$$A = \sum_{\alpha=1}^{r} u_\alpha v_\alpha^\top = UV^\top$$

$$U = [u_1, \dots, u_r], \quad V = [v_1, \dots, v_r]$$

(matrices of size $n \times r$)

$$2rn \ll n^2$$

Advantages of low rank

- Data compression
  - for matrices of size $n \times n$:
    $$r \ll n \quad \Rightarrow \quad 2nr \ll n^2$$
  - for d-tensors of size $n \times \ldots \times n$:
    $$r \ll n^{d-1} \quad \Rightarrow \quad dnr \ll n^d$$
- Extraction of most meaningful information and suppression of noise
- Fast computations in low-rank formats

Singular Value Decomposition (SVD)

It is a skeleton decomposition in which vectors $u_1, \dots, u_r$ и $v_1, \dots, v_r$ are orthogonal..

Afer normalization

$$u_\alpha v_\alpha^\top = \sigma_\alpha u_\alpha v_\alpha^\top, \quad u_\alpha := u_\alpha / ||u_\alpha||, \quad v_\alpha := v_\alpha / ||v_\alpha||,$$

we come up with a *Singular Value Decomposition*:

$$A = \sum_{\alpha=1}^{r} \sigma_\alpha u_\alpha v_\alpha^\top = U \Sigma V^\top, \quad \Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix},$$

$$\sigma_1 \geqslant \sigma_2 \geqslant \dots \geqslant \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0.$$

Low-rank approximations by a truncated SVD

$$\min_{\operatorname{rank} B \leqslant k} ||A - B|| = ||A - A_k|| = \sqrt{\sum_{\alpha=k+1}^{n} \sigma_\alpha^2}$$

$$A_k = \sum_{\alpha=1}^{k} \sigma_\alpha u_\alpha v_\alpha^\top$$

$$||A|| = \sqrt{\sum_{i,j} |a_{ij}|^2} \quad \text{(Frobenius norm)}$$

Tensor Train Model

$$x_{i_1,\ldots,i_d} = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} g^{(1)}_{i_1,\alpha_1}\, g^{(2)}_{\alpha_1,i_2,\alpha_2} \cdots g^{(d-1)}_{\alpha_{d-2},i_{d-1},\alpha_{d-1}}\, g^{(d)}_{\alpha_{d-1},i_d}$$

Symbolic notation: $x = g^{(1)} \ldots g^{(d)}$

Total number of TT-model parameters depends on d

linearly: $\sum_{k=1}^{d} r_{k-1} r_k n_k \leqslant dnr^2, \quad n = \max_k n_k, \quad r = \max_k r_k.$

The train car sizes r_k are called TT-ranks.

Options to represent the train cars

- by matrices of size of size $r_{k-1} \times r_k$: $\quad (G^{(k)}_{i_k})_{\alpha_{k-1},\alpha_k} := g^{(k)}_{\alpha_{k-1},i_k,\alpha_k}$

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

- by column vectors of size n_k: $\quad \left(g^{(k)}(\alpha_{k-1},\alpha_k)\right)_{i_k} := g^{(k)}_{\alpha_{k-1},i_k,\alpha_k}$

$$x = \sum_{\alpha_1=1}^{r_1} \dots \sum_{\alpha_{d-1}=1}^{r_{d-1}} g^{(1)}(\alpha_0,\alpha_1) \otimes g^{(2)}(\alpha_1,\alpha_2) \otimes \dots \otimes g^{(d)}(\alpha_{d-1},\alpha_d).$$

Everything reduces to matrices

$$r_k \geqslant \operatorname{rank}(A_k), \quad (A_k)_{(i_1,\dots,i_k),(i_{k+1},\dots,i_d)} := x_{i_1,\dots,i_d}.$$

Theorem.

Any tensor admits a TT-representation with $r_k = \operatorname{rank}(A_k)$.

TT-rank reduction

Find an approximation

$$\tilde{x} = \tilde{g}^{(1)} \dots \tilde{g}^{(d)} \;\approx\; x = g^{(1)} \dots g^{(d)}$$

with smaller TT-ranks and a guaranteed accuracy.

Reducing the rank of kth car

$$x_{i_1,\dots,i_d} = G^{(1)}_{i_1} \dots G^{(d)}_{i_d}$$

$$u_{i_1,\dots,i_k} = G^{(1)}_{i_1} \dots G^{(k)}_{i_k} \text{ (rows)}, \quad v_{i_{k+1},\dots,i_d} = G^{(k+1)}_{i_{k+1}} \dots G^{(d)}_{i_d} \text{ (columns)}$$

$$A_k = U_k V_k$$

- Orthogonalize columns in U_k and rows in V_k:

$$U_k = P_k S_k, \;\; dV_k = T_k Q_k \;\; \Rightarrow \;\; A_k = U_k V_k = P_k B_k Q_k, \;\; B_k = S_k T_k.$$

- Approximate $B_k \approx \tilde{B}_k$ with $\tilde{r}_k < r_k$ and $||B_k - \tilde{B}_k|| = \varepsilon$.
- Define tensor $\tilde{x}$ by $\tilde{A}_k := P_k \tilde{B}_k Q_k$. Then

$$||x - \tilde{x}|| = ||A_k - \tilde{A}_k|| = ||B_k - \tilde{B}_k|| = \varepsilon.$$

Rank reduction can be fast

$$\Pi'_k = \begin{bmatrix} G_1^{(k)} \\ \dots \\ G_{n_k}^{(k)} \end{bmatrix}, \quad \Pi''_k = \begin{bmatrix} G_1^{(k)} & \dots & G_{n_k}^{(k)} \end{bmatrix}.$$

- Train car $g^{(k)}$ is called *left-orthogonal* if Π'_k has orthonormal columns.
- Train car $g^{(k)}$ is called *right-orthogonal* if Π''_k has orthonormal rows.

- *If $g^{(1)}, \dots, g^{(k)}$ are left-orthogonal, then U_k has orthonormal columns.*
- *If $g^{(k+1)}, \dots, g^{(d)}$ are right-orthogonal, then V_k has orthonormal rows.*

Make the first car left-orthogonal

$$(U_k)_{(i_1,\ldots,i_k),\alpha_k} = \sum_{\alpha_1,\ldots,\alpha_{k-1}} g^{(1)}_{i_1,\alpha_1} \cdots g^{(k)}_{\alpha_{k-1},i_k,\alpha_k}$$

$$g^{(1)}_{i_1,\alpha_1} = \sum_{\beta_1} p^{(1)}_{i_1,\beta_1} s^{(1)}_{\beta_1,\alpha_1}$$

$$g^{(2)}_{\beta_1,i_2,\alpha_2} := \sum_{\alpha_1} s^{(1)}_{\beta_1,\alpha_1} g^{(2)}_{\alpha_1,i_2,\alpha_2}$$

$$(U_k)_{(i_1,\ldots,i_k),\alpha_k} = \sum_{\beta_1,\alpha_2,\ldots,\alpha_{k-1}} p^{(1)}_{i_1,\beta_1}\, g^{(2)}_{\beta_1,i_2,\alpha_2} \,\cdots\, g^{(k)}_{\alpha_{k-1},i_k,\alpha_k}$$

Fast structured SVD for the associated matrix A_k

$$(U_k)_{(i_1,\ldots,i_k),\alpha_k} = \sum_{\beta_1,\ldots,\beta_{k-1},\beta_k} p^{(1)}_{i_1,\beta_1}\, p^{(2)}_{\beta_1,i_2,\beta_2} \ \ldots\ p^{(k)}_{\beta_{k-1},i_k,\beta_k}\, s^{(k)}_{\beta_k,\alpha_k}$$

$$(V_k)_{\alpha_k,(i_{k+1},\ldots,i_d)} = \sum_{\gamma_k,\gamma_{k+1},\ldots,\gamma_{d-1}} t^{(k+1)}_{\alpha_k,\gamma_k}\, q^{(k+1)}_{\gamma_k,i_{k+1},\gamma_{k+1}} \ \ldots\ q^{(d)}_{\gamma_{d-1},i_d}$$

$$x_{i_1,\ldots,i_d} = \sum_{\beta_1,\ldots,\beta_k,\gamma_k,\ldots,\gamma_{d-1}} p^{(1)}_{i_1,\beta_1} \ \ldots\ p^{(k)}_{\beta_{k-1},i_k,\beta_k}\, h^{(k)}_{\beta_k,\gamma_k}\, q^{(k+1)}_{\gamma_k,i_{k+1},\gamma_{k+1}} \ \ldots\ q^{(d)}_{\gamma_{d-1},i_d}$$

$$h^{(k)}_{\beta_k,\gamma_k} = \sum_{\alpha_k} s^{(k)}_{\beta_k,\alpha_k}\, t^{(k+1)}_{\alpha_k,\gamma_k}$$

$$g^{(1)} \ldots g^{(d)} = p^{(1)} \ldots p^{(k)}\ h^{(k)}\ q^{(k+1)} \ldots q^{(d)}$$

Fast reduction of TT-ranks

$$g_1 g_2 g_3 g_4 g_5 \to g_1 g_2 g_3 g_4 g_5 \to g_1 g_2 g_3 g_4 g_5 \to g_1 g_2 g_3 g_4 g_5$$

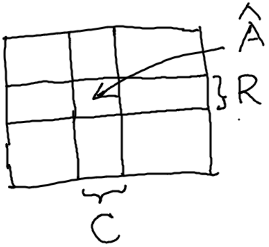
$$g_1 g_2 g_3 g_4 g_5 \to g_1 g_2 g_3 g_4 g_5 \to g_1 g_2 g_3 g_4 g_5 \to g_1 g_2 g_3 g_4 g_5$$

FAST: total number of operations is of order dnr^3.

SAFE: accuracy estimate is guaranteed.

Cross Column-Row Approximation

$A =$

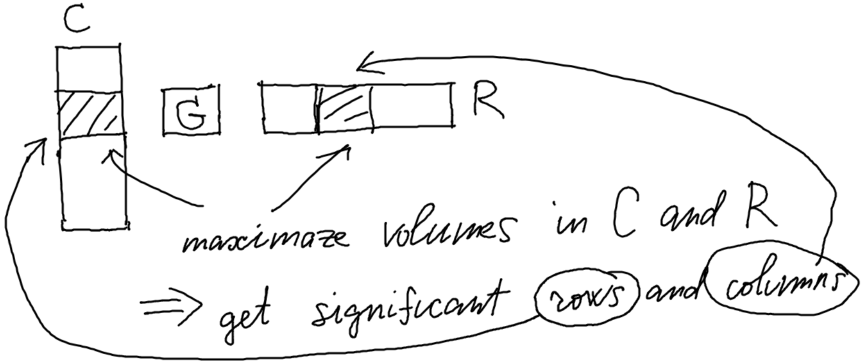


$$= C\hat{A}^{-1}R$$

$$\approx CGR$$

$\hat{A}, G$ — $r \times r$

How to find a "good cross"



C
G
R
maximaze volumes in C and R
⟹ get significant rows and columns

Maximal volume principle

THEOREM (Goreinov, Tyrtyshnikov).

Assume that $A_{11} \in \mathbb{C}^{r \times r}$ is a maximal volume block among all $r \times r$ blocks in

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad r = \mathrm{rank}\,(A+F).$$

Then

$$||A - CA_{11}^{-1}R||_C \leqslant (r+1)^2 ||F||_C,$$

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

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

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

How to find a "good volume block"

Pick up an invertible block $\hat{C} \in \mathbb{R}^{k\times k}$ in $C \in \mathbb{R}^{n\times k}$ and get to a matrix

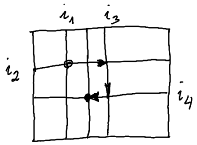
$$C\hat{C}^{-1} = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ q_{r+1,1} & \dots & q_{r+1,r} \\ \dots & \dots & \dots \\ q_{n1} & \dots & q_{nr} \end{bmatrix}$$

For $\hat{C}$ to be of maximal volume it is necessary that
$|q_{ij}| \leqslant 1, \quad r+1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant r.$
If not, then swap two rows and increase the volume!

D.Knuth, Semi-optimal bases for linear dependencies,

Linear and Mulilinear Algebra, 1985

Search for large elements in low-rank matrices

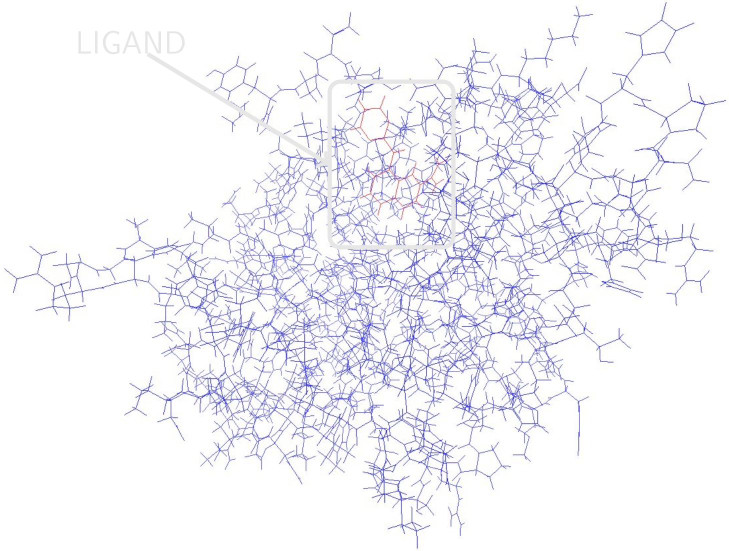


In the column submatrix $C \in \mathbb{R}^{n\times k}$ look for k “good rows”. They determine a row submatrix $R \in \mathbb{R}^{k\times n}$ where we select k “good columns”. The new columns give us a new columns submatrix $C \in \mathbb{R}^{k\times n}$ in which we pick up k “good rows”. And so on.

With high probability this algorithm leads to a block of close to maximal volume. It becomes a base for a new heuristic global optimization algorithm. For some applicaions (eg docking in the drug design) it appears 2-3 orders better than genetic algorithms.

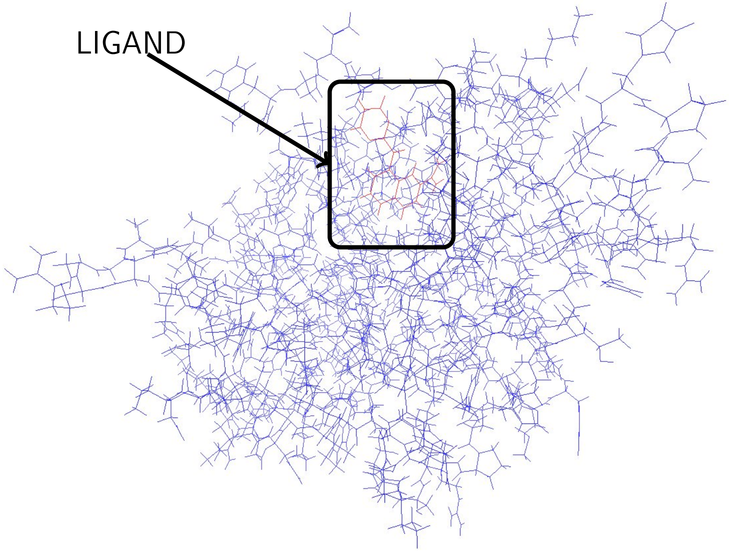
Direct docking in the drug design

ACCOMMODATION OF LIGAND INTO PROTEIN



Direct docking in the drug design

ACCOMMODATION OF LIGAND INTO PROTEIN



Tensor Train Learning and Optimization

Tensor Train:

$$a_{i_1,\ldots,i_d} = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} g^{(1)}_{i_1\alpha_1}\, g^{(2)}_{\alpha_1 i_2 \alpha_2} \, \ldots \, g^{(d-1)}_{\alpha_{d-2} i_{d-1} \alpha_{d-1}}\, g^{(d)}_{\alpha_{d-1} i_d}$$

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

Algorithms use structured low-rank representations
of certain associated matrices

$$A_k = [a^k_{i_1\ldots i_k; i_{k+1}\ldots i_d}]_{(n_1\ldots n_k)\times(n_{k+1}\ldots n_d)}$$

$$a^k_{i_1\ldots i_k; i_{k+1}\ldots i_d} = a_{i_1,\ldots,i_d}$$

Fast summation of astronomically big vector

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

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

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

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

Tensor Train and quadratures

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

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

$n = 11$ nodes per one dimension $\Rightarrow$ as amny as n^d values!
But only neglible part is computed.

d	$I(d)$	Rel.error	Time
100	-3.926795e-03	2.915654e-13	0.77
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

Tensorization of vectors and matrices

Any vector of size $N = n_1 \dots n_d$ can be regarded as a d-array, and any $N \times N$ matrix

$$a(i,j) = a(i_1 \dots i_d,\ j_1 \dots j_d)$$

can be viewed as as $2d$-array, or better, as a d-array of size $n_1^2 \times \dots \times n_d^2$. of the form

$$a(i_1 j_1, \dots, i_d j_d)$$

Tensorization followed up by Tensor Train construction can dramatically decrease the number of representation parameters in the model!

We may have a blessing of dimensionality!

- For astronomically large data we must look for an adequate low-parametric models.
- For conventional big data tensorization may lead to tremendously efficient representation models.

Thanks for your kind attention!

Эта книга отличается от традиционных учебников: понятия и факты линейной алгебры во многих случаях изложены как факты матричного анализа. Такой подход делает изложение менее абстрактным и позволяет познакомить читателя с матричным анализом как относительно самостоятельной дисциплиной.

Предмет линейной алгебры понимается в расширенном смысле, часто мы оказываемся на территории смежных дисциплин — математического анализа, вычислительных методов и, конечно, общей алгебры. Отдельные места в книге содержат материал, который вообще нельзя найти в каких-либо учебниках и даже монографиях. В частности, рассматриваются многомерные массивы, классические и относительно новые тензорные разложения, тензорные ранги и их связь с разработкой быстрых методов умножения матриц.

ISBN 978-5-4439-1879-2

Е. Е. Тыртышников

МАТРИЧНЫЙ АНАЛИЗ И ЛИНЕЙНАЯ АЛГЕБРА

02

Е. Е. Тыртышников

МАТРИЧНЫЙ АНАЛИЗ И ЛИНЕЙНАЯ АЛГЕБРА

Математические основы ИИ

02