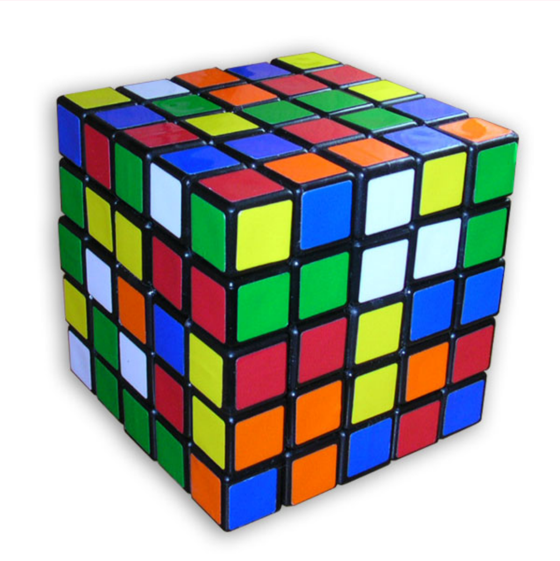
From Matrix to Tensor

Charles F. Van Loan

Department of Computer Science

January 28, 2016



Instead of just

$A(i, j)$

it's

$A(i, j, k)$

or

$A(i_1, i_2, \ldots, i_d)$

Where Might They Come From?

Discretization

$\mathcal{A}(i, j, k, \ell)$ might house the value of $f(w, x, y, z)$ at $(w, x, y, z) = (w_i, x_j, y_k, z_\ell)$.

High-Dimension Evaluations

Given a basis $\{\phi_i(\mathbf{r})\}_{i=1}^n$

$$\mathcal{A}(p, q, r, s) \;=\; \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{\phi_p(\mathbf{r}_1)\phi_q(\mathbf{r}_1)\phi_r(\mathbf{r}_2)\phi_s(\mathbf{r}_2)}{\|\mathbf{r}_1 - \mathbf{r}_2\|} d\mathbf{r}_1 d\mathbf{r}_2.$$

Multiway Analysis

$\mathcal{A}(i, j, k, \ell)$ is a value that captures an interaction between four variables/factors.

You May Have Seen them Before...

Here is a 3x3 **block matrix** with 2x2 blocks:

$$A = \left[\begin{array}{cc|cc|cc} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & \boxed{\mathbf{a}_{45}} & a_{46} \\ \hline a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{array}\right]$$

This is a **reshaping** of a $2 \times 2 \times 3 \times 3$ tensor:

Matrix entry a_{45} is the (2,1) entry of the (2,3) block.

Matrix entry a_{45} is $\mathcal{A}(2, 3, 2, 1)$.

A Tensor Has Parts

A matrix has columns and rows. A tensor has **fibers**.

A fiber of a tensor $\mathcal{A}$ is a vector obtained by fixing all but one $\mathcal{A}$'s indices.

Given $\mathcal{A} = \mathcal{A}(1{:}3, 1{:}5, 1{:}4, 1{:}7)$, here is a mode-2 fiber:

$$\mathcal{A}(2, 1{:}5, 4, 6) = \begin{bmatrix} \mathcal{A}(2, 1, 4, 6) \\ \mathcal{A}(2, 2, 4, 6) \\ \mathcal{A}(2, 3, 4, 6) \\ \mathcal{A}(2, 4, 4, 6) \\ \mathcal{A}(2, 5, 4, 6) \end{bmatrix}$$

This is the (2,4,6) mode-2 fiber.

Fibers Can Be Assembled Into a Matrix

The mode-1, mode-2, and mode-3 **unfoldings** of $\mathcal{A} \in \mathbb{R}^{4\times3\times2}$:

$$\mathcal{A}_{(1)} = \begin{bmatrix} a_{111} & a_{121} & a_{131} & a_{112} & a_{122} & a_{132} \\ a_{211} & a_{221} & a_{231} & a_{212} & a_{222} & a_{232} \\ a_{311} & a_{321} & a_{331} & a_{312} & a_{322} & a_{332} \\ a_{411} & a_{421} & a_{431} & a_{412} & a_{422} & a_{432} \end{bmatrix}$$

(1,1) (2,1) (3,1) (1,2) (2,2) (3,2)

$$\mathcal{A}_{(2)} = \begin{bmatrix} a_{111} & a_{211} & a_{311} & a_{411} & a_{112} & a_{212} & a_{312} & a_{412} \\ a_{121} & a_{221} & a_{321} & a_{421} & a_{122} & a_{222} & a_{322} & a_{422} \\ a_{131} & a_{231} & a_{331} & a_{431} & a_{132} & a_{232} & a_{332} & a_{432} \end{bmatrix}$$

(1,1) (2,1) (3,1) (4,1) (1,2) (2,2) (3,2) (4,2)

$$\mathcal{A}_{(3)} = \begin{bmatrix} a_{111} & a_{211} & a_{311} & a_{411} & a_{121} & a_{221} & a_{321} & a_{421} & a_{131} & a_{231} & a_{331} & a_{431} \\ a_{112} & a_{212} & a_{312} & a_{412} & a_{122} & a_{222} & a_{322} & a_{422} & a_{132} & a_{232} & a_{332} & a_{432} \end{bmatrix}$$

(1,1) (2,1) (3,1) (4,1) (1,2) (2,2) (3,2) (4,2) (1,3) (2,3) (3,3) (4,3)

There are Many Ways to Unfold a Given Tensor

Here is one way to unfold $\mathcal{A}(1{:}2, 1{:}3, 1{:}2, 1{:}2, 1{:}3)$:

$$B = \begin{array}{c} \begin{array}{cccccc} (1,1) & (2,1) & (1,2) & (2,2) & (1,3) & (2,3) \end{array} \\ \left[\begin{array}{cccccc}
a_{11111} & a_{11121} & a_{11112} & a_{11122} & a_{11113} & a_{11123} \\
a_{21111} & a_{21121} & a_{21112} & a_{21122} & a_{21113} & a_{21123} \\
a_{12111} & a_{12121} & a_{12112} & a_{12122} & a_{12113} & a_{12123} \\
a_{22111} & a_{22121} & a_{22112} & a_{22122} & a_{22113} & a_{22123} \\
a_{13111} & a_{13121} & a_{13112} & a_{13122} & a_{13113} & a_{13123} \\
a_{23111} & a_{23121} & a_{23112} & a_{23122} & a_{23113} & a_{23123} \\
a_{11211} & a_{11221} & a_{11212} & a_{11222} & a_{11213} & a_{11223} \\
a_{21211} & a_{21221} & a_{21212} & a_{21222} & a_{21213} & a_{21223} \\
a_{12211} & a_{12221} & a_{12212} & a_{12222} & a_{12213} & a_{12223} \\
a_{22211} & a_{22221} & a_{22212} & a_{22222} & a_{22213} & a_{22223} \\
a_{13211} & a_{13221} & a_{13212} & a_{13222} & a_{13213} & a_{13223} \\
a_{23211} & a_{23221} & a_{23212} & a_{23222} & a_{23213} & a_{23223}
\end{array}\right] \end{array}
\begin{array}{c} \\ (1,1,1) \\ (2,1,1) \\ (1,2,1) \\ (2,2,1) \\ (1,3,1) \\ (2,3,1) \\ (1,1,2) \\ (2,1,2) \\ (1,2,2) \\ (2,2,2) \\ (1,3,2) \\ (2,3,2) \end{array}$$

With the Matlab Tensor Toolbox: `B = tenmat(A,[1 2 3],[4 5])`

There are Many Ways to Unfold a Given Tensor

tenmat(A,[1 2 3],[4 5])	tenmat(A,[4 5],[1 2 3])
tenmat(A,[1 2 4],[3 5])	tenmat(A,[3,5],[1 2 4])
tenmat(A,[1 2 5],[4 5])	tenmat(A,[4 5],[1 2 5])
tenmat(A,[1 3 4],[2 5])	tenmat(A,[2 5],[1 3 4])
tenmat(A,[1 3 5],[2 5])	tenmat(A,[2 5],[1 3 5])
tenmat(A,[1 4 5],[2 3])	tenmat(A,[2 3],[1 4 5])
tenmat(A,[2 3 4],[1 5])	tenmat(A,[1 5],[2 3 4])
tenmat(A,[2 3 5],[1 4])	tenmat(A,[1 4],[2 3 5])
tenmat(A,[2 4 5],[1 3])	tenmat(A,[1 3],[2 4 5])
tenmat(A,[3 4 5],[1 2])	tenmat(A,[1 2],[3 4 5])

tenmat(A,[1],[2 3 4 5])	tenmat(A,[2 3 4 5],[1])
tenmat(A,[2],[1 3 4 5])	tenmat(A,[1 3 4 5],[2])
tenmat(A,[3],[1 2 4 5])	tenmat(A,[1 2 4 5],[3])
tenmat(A,[4],[1 2 3 5])	tenmat(A,[1 2 3 5],[4])
tenmat(A,[5],[1 2 3 4])	tenmat(A,[1 2 3 4],[5])

Choice makes life complicated...

Paradigm for Much of Tensor Computations

To say something about a tensor $\mathcal{A}$:

1. Thoughtfully unfold tensor $\mathcal{A}$ into a matrix A.

2. Use classical matrix computations to **discover** something interesting/useful about matrix A.

3. Map your insights back to tensor $\mathcal{A}$.

Computing (parts of) decompositions is how we do this in classical matrix computations.

$A = U\Sigma V^T \quad PA = LU \quad A = QR \quad A = GG^T \quad PAP^T = LDL^T \quad Q^TAQ = D$
$X^{-1}AX = J \quad U^TAU = T \quad AP = QR \quad A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T$
$PA = LU \quad A = QR \quad A = GG^T \quad PAP^T = LDL^T \quad Q^TAQ = D \quad X^{-1}AX = J$
$U^TAU = T \quad AP = QR \quad A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T \quad PA = LU$
$A = QR \quad A = GG^T \quad PAP^T = LDL^T \quad Q^TAQ = D \quad X^{-1}AX = J \quad U^TAU = T$
$AP = QR \quad A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T \quad PA = LU \quad A = QR$
$A = GG^T \quad PAP^T = LDL^T \quad Q^TAQ = D \quad X^{-1}AX = J \quad U^TAU = T$
$A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T \quad PA = LU \quad A = QR \quad A = GG^T$
$PAP^T = LDL^T \quad Q^TAQ = D \quad X^{-1}AX = J \quad U^TAU = T \quad AP = QR$
$A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T \quad PA = LU \quad A = QR \quad A = GG^T$
$PAP^T = LDL^T \quad Q^TAQ = D \quad X^{-1}AX = J \quad AP = QR \quad A = ULV^T$
$PAQ^T = LU \quad A = U\Sigma V^T \quad PA = LU \quad A = QR \quad A = GG^T \quad PAP^T = LDL^T$
$Q^TAQ = D \quad X^{-1}AX = J \quad U^TAU = T \quad AP = QR \quad A = ULV^T \quad PAQ^T = LU$
$A = U\Sigma V^T \quad PA = LU \quad A = QR \quad A = GG^T \quad PAP^T = LDL^T \quad Q^TAQ = D$
$X^{-1}AX = J \quad U^TAU = T \quad AP = QR \quad A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T$
$PA = LU \quad A = QR \quad PAP^T = LDL^T \quad Q^TAQ = D \quad X^{-1}AX = J \quad U^TAU = T$
$AP = QR \quad A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T \quad PA = LU \quad A = QR$

It's a Language

$A = U\Sigma V^T \quad PA = LU \quad A = QR \quad A = GG^T \quad PAP^T = LDL^T \quad Q^TAQ = D$

$X^{-1}AX = J \quad U^TAU = T \quad AP = QR \quad A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T$

$PA = LU \quad A = QR \quad A = GG^T \quad PAP^T = LDL^T \quad Q^TAQ = D \quad X^{-1}AX = J$

$U^TAU = T \quad AP = QR \quad A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T \quad PA = LU$

$A = QR \quad A = GG^T \quad PAP^T = LDL^T \quad Q^TAQ = D \quad X^{-1}AX = J \quad U^TAU = T$

$AP = QR \quad A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T \quad PA = LU \quad A = QR$

It's a Language

$A = GG^T \quad PAP^T = LDL^T \quad Q^TAQ = D \quad X^{-1}AX = J \quad U^TAU = T$

$A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T \quad PA = LU \quad A = QR \quad A = GG^T$

$PAP^T = LDL^T \quad Q^TAQ = D \quad X^{-1}AX = J \quad U^TAU = T \quad AP = QR$

$A = ULV^T \quad PAQ^T = LU \quad \ldots = LU \quad A = QR \quad A = GG^T$

$PAP^T = LDL^T \quad Q^TAQ = \ldots \quad AP = QR \quad A = ULV^T$

$PAQ^T = LU \quad A = U\Sigma V^T \quad \ldots \quad A = GG^T \quad PAP^T = LDL^T$

$Q^TAQ = D \quad X^{-1}AX = J \quad \ldots QR \quad A = ULV^T \quad PAQ^T = LU$

$A = U\Sigma V^T \quad PA = LU \quad A = \ldots \quad PAP^T = LDL^T \quad Q^TAQ = D$

$X^{-1}AX = J \quad U^TAU = T \quad AP = QR \quad A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T$

$PA = LU \quad A = QR \quad PAP^T = LDL^T \quad Q^TAQ = D \quad X^{-1}AX = J \quad U^TAU = T$

$AP = QR \quad A = ULV^T \quad PAQ^T = LU \quad A = U\Sigma V^T \quad PA = LU \quad A = QR$

The Singular Value Decomposition

Perhaps the most versatile and important of all the different matrix decompositions is the SVD:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix}^T$$

$$= \sigma_1 \begin{bmatrix} c_1 \\ -s_1 \end{bmatrix} \begin{bmatrix} c_2 \\ -s_2 \end{bmatrix}^T + \sigma_2 \begin{bmatrix} s_1 \\ c_1 \end{bmatrix} \begin{bmatrix} s_2 \\ c_2 \end{bmatrix}^T$$

$$= \sigma_1 \begin{bmatrix} c_1 \\ -s_1 \end{bmatrix} [c_2 \;\; -s_2] + \sigma_2 \begin{bmatrix} s_1 \\ c_1 \end{bmatrix} [s_2 \;\; c_2]$$

where $c_1^2 + s_1^2 = 1$ and $c_2^2 + s_2^2 = 1$.

This is a very special sum of rank-1 matrices.

Rank-1 Matrices: You have Seen Them Before

$T =$

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

Rank-1 Matrices: They Are "Data Sparse"

$T =$

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

$= vv^T \qquad v = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{bmatrix}$

The Matrix SVD

Expresses the matrix as a special sum of rank-1 matrices. If $A \in \mathbb{R}^{n \times n}$ then

$$A = \sum_{k=1}^{n} \sigma_k u_k v_k^T$$

Here $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0$ and

$$U = [u_1 \mid u_2 \mid \cdots \mid u_n] \qquad V = [v_1 \mid v_2 \mid \cdots \mid v_n]$$

have columns that are mutually orthogonal.

Expresses the matrix as a special sum of rank-1 matrices. If $A \in \mathbb{R}^{n \times n}$ then

$$A = \sum_{k=1}^{n} \sigma_k u_k v_k^T$$

Here $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0$ and

$$U = [u_1 \mid u_2 \mid \cdots \mid u_n] \qquad V = [v_1 \mid v_2 \mid \cdots \mid v_n]$$

have columns that are mutually orthogonal.

That's how far A is from being rank deficient.

The Matrix SVD: Data Sparse Approximation

Expresses the matrix as a special sum of rank-1 matrices. If $A \in \mathbb{R}^{n \times n}$ then

$$A \approx \sum_{k=1}^{\tilde{r}} \sigma_k u_k v_k^T = A_{\tilde{r}}$$

Here $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0$ and

$$U = [u_1 \mid u_2 \mid \cdots \mid u_n] \qquad V = [v_1 \mid v_2 \mid \cdots \mid v_n]$$

have columns that are mutually orthogonal.

That's the closest matrix to A that has rank $\tilde{r}$.

If $\tilde{r} << n$, then that is a **data sparse** approximation of A because $O(n\tilde{r}) << O(n^2)$.

There is a New Definition of Big

In **Matrix Computations**, to say that $A \in \mathbb{R}^{n_1 \times n_2}$ is "big" is to say that both n_1 and n_2 are big. E.g.,

$$n_1 = 500000 \qquad n_2 = 100000$$

In **Tensor Computations**, to say that $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is "big" is to say that $n_1 n_2 \cdots n_d$ is big and this need not require big n_k. E.g.

$$n_1 = n_2 = \cdots = n_{1000} = 2.$$

Why Data Sparse Tensor Approximation is Important

1. If you want to see this

Matrix-Based Scientific Computation

$$\Downarrow$$

Tensor-Based Scientific Computation

you will need tensor algorithms that scale with d.

2. This requires a framework for low-rank tensor approximation.

3. **This requires some kind of tensor-level SVD.**

What is a Rank-1 Tensor? Think Matrix First

This:

$$R = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} = fg^T = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} [g_1 \ \ g_2] = \begin{bmatrix} f_1 g_1 & f_1 g_2 \\ f_2 g_1 & f_2 g_2 \end{bmatrix}$$

Is the same as this:

$$\text{vec}(R) = \begin{bmatrix} r_{11} \\ r_{21} \\ \hline r_{12} \\ r_{22} \end{bmatrix} = \begin{bmatrix} g_1 f_1 \\ g_1 f_2 \\ \hline g_2 f_1 \\ g_2 f_2 \end{bmatrix}$$

Is the same as this:

$$\text{vec}(R) = \begin{bmatrix} r_{11} \\ r_{21} \\ \hline r_{12} \\ r_{22} \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \otimes \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

The Kronecker Product of Vectors

$$x \otimes y = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \otimes \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 y_1 \\ x_1 y_2 \\ \hline x_2 y_1 \\ x_2 y_2 \\ \hline x_3 y_1 \\ x_3 y_2 \end{bmatrix} = \begin{bmatrix} x_1 y \\ \hline x_2 y \\ \hline x_3 y \end{bmatrix}$$

So What is a Rank-1 Tensor?

$\mathcal{R} \in \mathbb{R}^{2\times 2\times 2}$ is rank-1 if there exist $f, g, h \in \mathbb{R}^2$ such that

$$\text{vec}(\mathcal{R}) = \begin{bmatrix} r_{111} \\ r_{211} \\ r_{121} \\ r_{221} \\ r_{112} \\ r_{212} \\ r_{122} \\ r_{222} \end{bmatrix} = \begin{bmatrix} h_1 g_1 f_1 \\ h_1 g_1 f_2 \\ h_1 g_2 f_1 \\ h_1 g_2 f_2 \\ h_2 g_1 f_1 \\ h_2 g_1 f_2 \\ h_2 g_2 f_1 \\ h_2 g_2 f_2 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \otimes \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \otimes \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

$$r_{ijk} = h_k \cdot g_j \cdot f_i$$

What Might a Tensor SVD Look Like?

$$\text{vec}(\mathcal{R}) = \begin{bmatrix} r_{111} \\ r_{211} \\ r_{121} \\ r_{221} \\ r_{112} \\ r_{212} \\ r_{122} \\ r_{222} \end{bmatrix} = h^{(1)} \otimes g^{(1)} \otimes f^{(1)} + h^{(2)} \otimes g^{(2)} \otimes f^{(2)} + h^{(3)} \otimes g^{(3)} \otimes f^{(3)}$$

A "special" sum of rank-1 tensors.

What Does the Matrix SVD Look Like?

This:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}^T$$

$$= \sigma_1 \begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix}^T + \sigma_2 \begin{bmatrix} u_{12} \\ u_{22} \end{bmatrix} \begin{bmatrix} v_{12} \\ v_{22} \end{bmatrix}^T$$

Is the same as this:

$$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{bmatrix} = \sigma_1 \begin{bmatrix} v_{11}u_{11} \\ v_{11}u_{21} \\ v_{21}u_{11} \\ v_{21}u_{21} \end{bmatrix} + \sigma_2 \begin{bmatrix} v_{12}u_{12} \\ v_{12}u_{22} \\ v_{22}u_{12} \\ v_{22}u_{22} \end{bmatrix}$$

$$= \sigma_1 \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} \otimes \begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} + \sigma_2 \begin{bmatrix} v_{12} \\ v_{22} \end{bmatrix} \otimes \begin{bmatrix} u_{12} \\ u_{22} \end{bmatrix}$$

What Might a Tensor SVD Look Like?

$$\text{vec}(\mathcal{R}) = \begin{bmatrix} r_{111} \\ r_{211} \\ r_{121} \\ r_{221} \\ r_{112} \\ r_{212} \\ r_{122} \\ r_{222} \end{bmatrix} = h^{(1)} \otimes g^{(1)} \otimes f^{(1)} + h^{(2)} \otimes g^{(2)} \otimes f^{(2)} + h^{(3)} \otimes g^{(3)} \otimes f^{(3)}.$$

A "special" sum of rank-1 tensors.

Getting that special sum often requires **multilinear optimiziation**.

We better understand that before we proceed.

A Nearest Rank-1 Tensor Problem

Find $\sigma \geq 0$ and

$$\begin{bmatrix} c_1 \\ s_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix} \quad \begin{bmatrix} c_2 \\ s_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{bmatrix} \quad \begin{bmatrix} c_3 \\ s_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta_3) \\ \sin(\theta_3) \end{bmatrix}$$

so that

$$\phi(\sigma, \theta_1, \theta_2, \theta_3) = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \sigma \cdot \begin{bmatrix} c_3 \\ s_3 \end{bmatrix} \otimes \begin{bmatrix} c_2 \\ s_2 \end{bmatrix} \otimes \begin{bmatrix} c_1 \\ s_1 \end{bmatrix} \right\|_2$$

is minimized.

A Nearest Rank-1 Tensor Problem

Find $\sigma \geq 0$ and

$$\begin{bmatrix} c_1 \\ s_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix} \quad \begin{bmatrix} c_2 \\ s_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{bmatrix} \quad \begin{bmatrix} c_3 \\ s_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta_3) \\ \sin(\theta_3) \end{bmatrix}$$

so that

$$\phi(\sigma,\theta_1,\theta_2,\theta_3) = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \sigma \cdot \begin{bmatrix} c_3c_2c_1 \\ c_3c_2s_1 \\ c_3s_2c_1 \\ c_3s_2s_1 \\ s_3c_2c_1 \\ s_3c_2s_1 \\ s_3s_2c_1 \\ s_3s_2s_1 \end{bmatrix} \right\|_2$$

is minimized.

Alternating Least Squares

Freeze c_2, s_2, c_3 and s_3 and minimize

$$\phi = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \sigma \cdot \begin{bmatrix} c_3 c_2 c_1 \\ c_3 c_2 s_1 \\ c_3 s_2 c_1 \\ c_3 s_2 s_1 \\ s_3 c_2 c_1 \\ s_3 c_2 s_1 \\ s_3 s_2 c_1 \\ s_3 s_2 s_1 \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \begin{bmatrix} c_3 c_2 & 0 \\ 0 & c_3 c_2 \\ c_3 s_2 & 0 \\ 0 & c_3 s_2 \\ s_3 c_2 & 0 \\ 0 & s_3 c_2 \\ s_3 s_2 & 0 \\ 0 & s_3 s_2 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \right\|_2$$

with respect to

$$x_1 = \sigma c_1 \qquad y_1 = \sigma s_1$$

This is an ordinary linear least squares problem. We then get "improved" σ, c_1, and s_1 via

$$\sigma = \sqrt{{x_1}^2 + {y_1}^2} \qquad \begin{bmatrix} c_1 \\ s_1 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} / \sigma$$

Alternating Least Squares

Freeze c_1, s_1, c_3 and s_3 and minimize

$$\phi = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \sigma \cdot \begin{bmatrix} c_3 c_2 c_1 \\ c_3 c_2 s_1 \\ c_3 s_2 c_1 \\ c_3 s_2 s_1 \\ s_3 c_2 c_1 \\ s_3 c_2 s_1 \\ s_3 s_2 c_1 \\ s_3 s_2 s_1 \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \begin{bmatrix} c_3 c_1 & 0 \\ c_3 s_1 & 0 \\ 0 & c_3 c_1 \\ 0 & c_3 s_1 \\ s_3 c_1 & 0 \\ s_3 s_1 & 0 \\ 0 & s_3 c_1 \\ 0 & s_3 s_1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \right\|_2$$

with respect to

$$x_2 = \sigma c_2 \qquad y_2 = \sigma s_2$$

This is an ordinary linear least squares problem. We then get "improved" σ, c_2, and s_2 via

$$\sigma = \sqrt{{x_2}^2 + {y_2}^2} \qquad \begin{bmatrix} c_2 \\ s_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} / \sigma$$

Alternating Least Squares

Freeze c_1, s_1, c_2 and s_2 and minimize

$$\phi = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \sigma \cdot \begin{bmatrix} c_3 c_2 c_1 \\ c_3 c_2 s_1 \\ c_3 s_2 c_1 \\ c_3 s_2 s_1 \\ s_3 c_2 c_1 \\ s_3 c_2 s_1 \\ s_3 s_2 c_1 \\ s_3 s_2 s_1 \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \begin{bmatrix} c_2 c_1 & 0 \\ c_2 s_1 & 0 \\ s_2 c_1 & 0 \\ s_2 s_1 & 0 \\ 0 & c_2 s_1 \\ 0 & c_2 s_1 \\ 0 & s_2 c_1 \\ 0 & s_2 s_1 \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} \right\|_2$$

with respect to

$$x_3 = \sigma c_3 \qquad y_3 = \sigma s_3$$

This is an ordinary linear least squares problem. We then get "improved" σ, c_3, and s_3 via

$$\sigma = \sqrt{{x_3}^2 + {y_3}^2} \qquad \begin{bmatrix} c_3 \\ s_3 \end{bmatrix} = \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} / \sigma$$

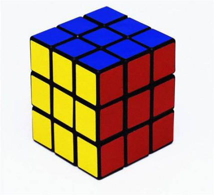
Componentwise Optimization

A Common Framework for Tensor-Related Optimization:

- Choose a subset of the unknowns such that if they are (temporarily) fixed, then we are presented with some standard matrix problem in the remaining unknowns.
- By choosing different subsets, cycle through all the unknowns.
- Repeat until converged.

The "standard matrix problem" that we end up solving is usually some kind of linear least squares problem.

We Are Now Ready For This!

U^T  V $=$

8	3	5	4	1	6	9	2	7
2	9	6	8	5	7	4	3	1
4	1	7	2	9	3	6	5	8
5	6	9	1	3	4	7	8	2
1	2	3	6	7	8	5	4	9
7	4	8	5	2	9	1	6	3
6	5	2	7	8	1	3	9	4
9	8	1	3	4	5	2	7	6
3	7	4	9	6	2	8	1	5

That is, we are ready to look at SVD ideas at the tensor level.

Motivation:

In the matrix case, if $A \in \mathbb{R}^{n_1 \times n_2}$ and $A = U_1 S U_2^T$, then

$$\text{vec}(A) = \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} S(j_1, j_2) \cdot U_2(:, j_2) \otimes U_1(:, j_1)$$

We are able to choose orthogonal U_1 and U_2 so that $S = U_1^T A U_2$ is diagonal.

Definition:

Given $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, compute the SVDs of the modal unfoldings

$$\begin{aligned} \mathcal{A}_{(1)} &= U_1 \Sigma_1 V_1^T \\ \mathcal{A}_{(2)} &= U_2 \Sigma_2 V_2^T \\ \mathcal{A}_{(3)} &= U_3 \Sigma_3 V_3^T \end{aligned}$$

and then compute $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ so that

$$\text{vec}(\mathcal{A}) = \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \sum_{j_3=1}^{n_3} \mathcal{S}(j_1, j_2, j_3) \cdot U_3(:, j_3) \otimes U_2(:, j_2) \otimes U_1(:, j_1)$$

Recall...

The mode-1, mode-2, and mode-3 **unfoldings** of $\mathcal{A} \in \mathbb{R}^{4\times3\times2}$:

$$\mathcal{A}_{(1)} = \begin{bmatrix} a_{111} & a_{121} & a_{131} & a_{112} & a_{122} & a_{132} \\ a_{211} & a_{221} & a_{231} & a_{212} & a_{222} & a_{232} \\ a_{311} & a_{321} & a_{331} & a_{312} & a_{322} & a_{332} \\ a_{411} & a_{421} & a_{431} & a_{412} & a_{422} & a_{432} \end{bmatrix}$$

(1,1) (2,1) (3,1) (1,2) (2,2) (3,2)

$$\mathcal{A}_{(2)} = \begin{bmatrix} a_{111} & a_{211} & a_{311} & a_{411} & a_{112} & a_{212} & a_{312} & a_{412} \\ a_{121} & a_{221} & a_{321} & a_{421} & a_{122} & a_{222} & a_{322} & a_{422} \\ a_{131} & a_{231} & a_{331} & a_{431} & a_{132} & a_{232} & a_{332} & a_{432} \end{bmatrix}$$

(1,1) (2,1) (3,1) (4,1) (1,2) (2,2) (3,2) (4,2)

$$\mathcal{A}_{(3)} = \begin{bmatrix} a_{111} & a_{211} & a_{311} & a_{411} & a_{121} & a_{221} & a_{321} & a_{421} & a_{131} & a_{231} & a_{331} & a_{431} \\ a_{112} & a_{212} & a_{312} & a_{412} & a_{122} & a_{222} & a_{322} & a_{422} & a_{132} & a_{232} & a_{332} & a_{432} \end{bmatrix}$$

(1,1) (2,1) (3,1) (4,1) (1,2) (2,2) (3,2) (4,2) (1,3) (2,3) (3,3) (4,3)

The Truncated Higher-Order SVD

The HO-SVD:

$$\text{vec}(\mathcal{A}) \;=\; \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \sum_{j_3=1}^{n_3} \mathcal{S}(j_1, j_2, j_3) \cdot U_3(:, j_3) \otimes U_2(:, j_2) \otimes U_1(:, j_1)$$

The **core tensor** $\mathcal{S}$ is not diagonal, but its entries get smaller as you move away from the (1,1,1) entry.

The Truncated HO-SVD:

$$\text{vec}(\mathcal{A}) \;=\; \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \sum_{j_3=1}^{r_3} \mathcal{S}(j_1, j_2, j_3) \cdot U_3(:, j_3) \otimes U_2(:, j_2) \otimes U_1(:, j_1)$$

The Tucker Nearness Problem

Assume that $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Given integers $\tilde{r}_1$, $\tilde{r}_2$ and $\tilde{r}_3$ compute

$$\begin{array}{lll} U_1: & n_1 \times \tilde{r}_1, & \text{orthonormal columns} \\ U_2: & n_2 \times \tilde{r}_2, & \text{orthonormal columns} \\ U_3: & n_3 \times \tilde{r}_3, & \text{orthonormal columns} \end{array}$$

and tensor $\mathcal{S} \in \mathbb{R}^{\tilde{r}_1 \times \tilde{r}_2 \times \tilde{r}_3}$ so that

$$\left\| \text{vec}(\mathcal{A}) - \sum_{j_1=1}^{\tilde{r}_1} \sum_{j_2=1}^{\tilde{r}_2} \sum_{j_3=1}^{\tilde{r}_3} \mathcal{S}(j_1, j_2, j_3) \cdot U_3(:, j_3) \otimes U_2(:, j_2) \otimes U_1(:, j_1) \right\|_2$$

is minimized.

Componentwise Optimization

1. Fix U_2 and U_3 and minimize with respect to $\mathcal{S}$ and U_1:

$$\left\| \mathrm{vec}(\mathcal{A}) - \sum_{j_1=1}^{\tilde{r}_1} \sum_{j_2=1}^{\tilde{r}_2} \sum_{j_3=1}^{\tilde{r}_3} \mathcal{S}(j_1, j_2, j_3) \cdot U_3(:, j_3) \otimes U_2(:, j_2) \otimes U_1(:, j_1) \right\|_2$$

2. Fix U_1 and U_3 and minimize with respect to $\mathcal{S}$ and U_2:

$$\left\| \mathrm{vec}(\mathcal{A}) - \sum_{j_1=1}^{\tilde{r}_1} \sum_{j_2=1}^{\tilde{r}_2} \sum_{j_3=1}^{\tilde{r}_3} \mathcal{S}(j_1, j_2, j_3) \cdot U_3(:, j_3) \otimes U_2(:, j_2) \otimes U_1(:, j_1) \right\|_2$$

3. Fix U_1 and U_2 and minimize with respect to $\mathcal{S}$ and U_3:

$$\left\| \mathrm{vec}(\mathcal{A}) - \sum_{j_1=1}^{\tilde{r}_1} \sum_{j_2=1}^{\tilde{r}_2} \sum_{j_3=1}^{\tilde{r}_3} \mathcal{S}(j_1, j_2, j_3) \cdot U_3(:, j_3) \otimes U_2(:, j_2) \otimes U_1(:, j_1) \right\|_2$$

The CP-Decomposition

It also goes by the name of the CANDECOMP/PARAFAC Decomposition.

$$\text{CANDECOMP} = \text{Canonical Decomposition}$$

$$\text{PARAFAC} = \text{Parallel Factors Decomposition}$$

A Different Kind of Rank-1 Summation

The Tucker representation

$$\text{vec}(\mathcal{A}) = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \sum_{j_3=1}^{r_3} \mathcal{S}(j_1, j_2, j_3) \cdot U_3(:, j_3) \otimes U_2(:, j_2) \otimes U_1(:, j_1)$$

uses orthogonal U_1, U_2, and U_3.

The CP representation

$$\text{vec}(\mathcal{A}) = \sum_{j=1}^{r} \lambda_j \cdot U_3(:, j) \otimes U_2(:, j) \otimes U_1(:, j)$$

uses nonorthogonal U_1, U_2, and U_3.

The smallest possible r is called the rank of $\mathcal{A}$.

Tensor Rank is Trickier than Matrix Rank

If $\begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} = \texttt{randn(8,1)}$, then $\begin{cases} \text{rank} = 2 & \text{with prob } 79\% \\ \text{rank} = 3 & \text{with prob } 21\% \end{cases}$

This is Different from the Matrix Case

If $A = \texttt{randn(n,n)}$, then $\text{rank}(A) = n$ with probability 1.

Componentwise Optimization

Fix $r \leq \text{rank}(\mathcal{A})$ and minimize:

$$\left\| \text{vec}(\mathcal{A}) - \sum_{j=1}^{r} \lambda_j \cdot U_3(:,j) \otimes U_2(:,j) \otimes U_1(:,j) \right\|_2$$

Improve U_1 and the λ_j by fixing U_2 and U_3 and minimizing

$$\left\| \text{vec}(\mathcal{A}) - \sum_{j=1}^{r} \lambda_j \cdot U_3(:,j) \otimes U_2(:,j) \otimes U_1(:,j) \right\|_2$$

Etc.

The component optimizations are highly structured least squares problems.

The Tensor Train Decomposition

Idea: Approximate a high-order tensor with a collection of order-3 tensors.

Each order-3 tensor is connected to its left and right “neighbor” through a simple summation.

An example of a tensor network.

Tensor Train: An Example

Given the "carriages"...

$$\begin{aligned}
\mathcal{G}_1: &\quad n_1 \times r_1 \\
\mathcal{G}_2: &\quad r_1 \times n_2 \times r_2 \\
\mathcal{G}_3: &\quad r_2 \times n_3 \times r_3 \\
\mathcal{G}_4: &\quad r_3 \times n_4 \times r_4 \\
\mathcal{G}_5: &\quad r_4 \times n_5
\end{aligned}$$

We define the train" $\mathcal{A}(1{:}n_1, 1{:}n_2, 1{:}n_3, 1{:}n_4, 1{:}n_5)$...

$$\mathcal{A}(i_1, i_2, i_3, i_4, i_5)$$
$$=$$
$$\sum_{k_1=1}^{r_1} \sum_{k_2=1}^{r_2} \sum_{k_3=1}^{r_3} \sum_{k_4=1}^{r_4} \mathcal{G}_1(i_1, k_1) \cdot \mathcal{G}_2(k_1, i_2, k_2) \cdot \mathcal{G}_3(k_2, i_3, k_3) \cdot \mathcal{G}_4(k_3, i_4, k_4) \cdot \mathcal{G}_5(k_4, i_5)$$

Tensor Train: An Example

Given the "carriages"...

$$\begin{aligned} \mathcal{G}_1 &: \quad n_1 \times r_1 \\ \mathcal{G}_2 &: \quad r_1 \times n_2 \times r_2 \\ \mathcal{G}_3 &: \quad r_2 \times n_3 \times r_3 \\ \mathcal{G}_4 &: \quad r_3 \times n_4 \times r_4 \\ \mathcal{G}_5 &: \quad r_4 \times n_5 \end{aligned}$$

We define the train" $\mathcal{A}(1{:}n_1, 1{:}n_2, 1{:}n_3, 1{:}n_4, 1{:}n_5)$...

$$\mathcal{A}(i_1, i_2, i_3, i_4, i_5)$$

$$=$$

$$\sum_{k_1=1}^{r_1} \sum_{k_2=1}^{r_2} \sum_{k_3=1}^{r_3} \sum_{k_4=1}^{r_4} \mathcal{G}_1(i_1, k_1) \cdot \mathcal{G}_2(k_1, i_2, k_2) \cdot \mathcal{G}_3(k_2, i_3, k_3) \cdot \mathcal{G}_4(k_3, i_4, k_4) \cdot \mathcal{G}_5(k_4, i_5)$$

Tensor Train: An Example

Given the "carriages"...

$$\begin{aligned} \mathcal{G}_1: &\quad n_1 \times r_1 \\ \mathcal{G}_2: &\quad r_1 \times n_2 \times r_2 \\ \mathcal{G}_3: &\quad r_2 \times n_3 \times r_3 \\ \mathcal{G}_4: &\quad r_3 \times n_4 \times r_4 \\ \mathcal{G}_5: &\quad r_4 \times n_5 \end{aligned}$$

We define the train" $\mathcal{A}(1{:}n_1, 1{:}n_2, 1{:}n_3, 1{:}n_4, 1{:}n_5)$...

$$\mathcal{A}(i_1, i_2, i_3, i_4, i_5) = \sum_{k_1=1}^{r_1} \sum_{k_2=1}^{r_2} \sum_{k_3=1}^{r_3} \sum_{k_4=1}^{r_4} \mathcal{G}_1(i_1, k_1) \cdot \mathcal{G}_2(k_1, i_2, k_2) \cdot \mathcal{G}_3(k_2, i_3, k_3) \cdot \mathcal{G}_4(k_3, i_4, k_4) \cdot \mathcal{G}_5(k_4, i_5)$$

Tensor Train: An Example

Given the "carriages" ...

$$\begin{aligned} \mathcal{G}_1 &: \quad n_1 \times r_1 \\ \mathcal{G}_2 &: \quad r_1 \times n_2 \times r_2 \\ \mathcal{G}_3 &: \quad r_2 \times n_3 \times r_3 \\ \mathcal{G}_4 &: \quad r_3 \times n_4 \times r_4 \\ \mathcal{G}_5 &: \quad r_4 \times n_5 \end{aligned}$$

We define the train" $\mathcal{A}(1{:}n_1, 1{:}n_2, 1{:}n_3, 1{:}n_4, 1{:}n_5)$...

$$\mathcal{A}(i_1, i_2, i_3, i_4, i_5)$$
$$=$$
$$\sum_{k_1=1}^{r_1} \sum_{k_2=1}^{r_2} \sum_{k_3=1}^{r_3} \sum_{k_4=1}^{r_4} \mathcal{G}_1(i_1, k_1) \cdot \mathcal{G}_2(k_1, i_2, k_2) \cdot \mathcal{G}_3(k_2, i_3, k_3) \cdot \mathcal{G}_4(k_3, i_4, k_4) \cdot \mathcal{G}_5(k_4, i_5)$$

Tensor Train: An Example

Given the "carriages"...

$$\begin{aligned} \mathcal{G}_1: &\quad n_1 \times r_1 \\ \mathcal{G}_2: &\quad r_1 \times n_2 \times r_2 \\ \mathcal{G}_3: &\quad r_2 \times n_3 \times r_3 \\ \mathcal{G}_4: &\quad r_3 \times n_4 \times r_4 \\ \mathcal{G}_5: &\quad r_4 \times n_5 \end{aligned}$$

We define the train" $\mathcal{A}(1{:}n_1, 1{:}n_2, 1{:}n_3, 1{:}n_4, 1{:}n_5)$...

$$\mathcal{A}(i_1, i_2, i_3, i_4, i_5) \\ = \\ \sum_{k_1=1}^{r_1} \sum_{k_2=1}^{r_2} \sum_{k_3=1}^{r_3} \sum_{k_4=1}^{r_4} \mathcal{G}_1(i_1, k_1) \cdot \mathcal{G}_2(k_1, i_2, k_2) \cdot \mathcal{G}_3(k_2, i_3, k_3) \cdot \mathcal{G}_4(k_3, i_4, k_4) \cdot \mathcal{G}_5(k_4, i_5)$$

Tensor Train: An Example

Given the "carriages"...

$$\begin{aligned} \mathcal{G}_1: &\quad n_1 \times r \\ \mathcal{G}_2: &\quad r \times n_2 \times r \\ \mathcal{G}_3: &\quad r \times n_3 \times r \\ \mathcal{G}_4: &\quad r \times n_4 \times r \\ \mathcal{G}_5: &\quad r \times n_5 \end{aligned}$$

$$\mathcal{A}(i_1, i_2, i_3, i_4, i_5)$$

$$\approx$$

$$\sum_{k_1=1}^{r} \sum_{k_2=1}^{r} \sum_{k_3=1}^{r} \sum_{k_4=1}^{r} \mathcal{G}_1(i_1, k_1) \cdot \mathcal{G}_2(k_1, i_2, k_2) \cdot \mathcal{G}_3(k_2, i_3, k_3) \cdot \mathcal{G}_4(k_3, i_4, k_4) \cdot \mathcal{G}_5(k_4, i_5)$$

Data Sparse: $O(nr^2)$ instead of $O(n^5)$.

The Kronecker Product SVD

A way to obtain a data sparse representation of an order-4 tensor.

It is based on the Kronecker product of matrices, e.g.,

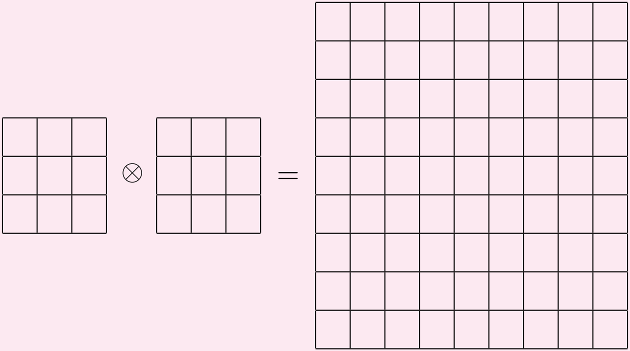
$$A = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \\ u_{31} & u_{32} \end{bmatrix} \otimes V = \begin{bmatrix} u_{11}V & u_{12}V \\ u_{21}V & u_{22}V \\ u_{31}V & u_{32}V \end{bmatrix}$$

and the fact that an order-4 tensor is a reshaped block matrix, e.g.,

$$\mathcal{A}(i_1, i_2, i_3, i_4) = U(i_1, i_2)V(i_3, i_4)$$

Kronecker Products are Data Sparse

If B and C are n-by-n, then $B \otimes C$ is n^2-by-n^2.



Thus, we need $O(n^2)$ numbers to describe an $O(n^4)$ object.

The Nearest Kronecker Product Problem

Find B and C so that $\| A - B \otimes C \|_F = \min$:

$$\left\| \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & \color{red}{a_{42}} & a_{43} & a_{44} \\ \hline a_{51} & a_{52} & a_{53} & a_{54} \\ a_{61} & a_{62} & a_{63} & a_{64} \end{array}\right] - \begin{bmatrix} b_{11} & b_{12} \\ \color{red}{b_{21}} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & \color{red}{c_{22}} \end{bmatrix} \right\|_F$$

$$=$$

$$\left\| \left[\begin{array}{cccc} a_{11} & a_{21} & a_{12} & a_{22} \\ \hline a_{31} & a_{41} & a_{32} & \color{red}{a_{42}} \\ \hline a_{51} & a_{61} & a_{52} & a_{62} \\ \hline a_{13} & a_{23} & a_{14} & a_{24} \\ \hline a_{33} & a_{43} & a_{34} & a_{44} \\ \hline a_{53} & a_{63} & a_{54} & a_{64} \end{array}\right] - \begin{bmatrix} b_{11} \\ \color{red}{b_{21}} \\ b_{31} \\ b_{12} \\ b_{22} \\ b_{32} \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} & c_{12} & \color{red}{c_{22}} \end{bmatrix} \right\|_F$$

The Kronecker Product SVD

If

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix} \qquad A_{ij} \in \mathbb{R}^{n \times n}$$

then there exist $U_1, \ldots, U_r \in \mathbb{R}^{n \times n}$, $V_1, \ldots, V_r \in \mathbb{R}^{n \times n}$, and scalars $\sigma_1 \geq \cdots \geq \sigma_r > 0$ such that

$$A \ = \sum_{k=1}^{r} \sigma_k U_k \otimes V_k .$$

A Tensor Approximation Idea

Unfold $\mathcal{A} \in \mathbb{R}^{n \times n \times n \times n}$ into an n^2-by-n^2 matrix A.

Express A as a sum of Kronecker products:

$$A = \sum_{k=1}^{r} \sigma_k B_k \otimes C_k \qquad B_k, C_k \in \mathbb{R}^{n \times n}$$

Back to tensor:

$$\mathcal{A}(i_1, i_2, j_1, j_2) = \sum_{k=1}^{r} \sigma_k C_k(i_1, i_2) B_k(j_1, j_2)$$

Sums of tensor products of matrices instead of vectors. $O(n^2 r)$

The Higher-Order Generalized Singular Value Decomposition

We are given a collection of m-by-n data matrices

$$\{A_1, \ldots, A_N\}$$

each of which has full column rank.

Do an "SVD thing" on each of them simultaneously:

$$\begin{aligned} A_1 &= U_1 \Sigma_1 V^T \\ &\vdots \\ A_N &= U_N \Sigma_N V^T \end{aligned}$$

that exposes "common features".

The 2-Matrix GSVD

If

$$A_1 = \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \qquad A_2 = \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}$$

then there exist orthogonal U_1, orthogonal U_2 and **nonsingular** X so that

$$U_1^T A_1 X = \Sigma_1 = \begin{bmatrix} c_1 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & c_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad U_2^T A_2 X = \Sigma_2 = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The Higher-Order GSVD Framework

1. Compute $V^{-1} S_N V = \text{diag}(\lambda_i)$ where

$$S_N = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left((A_i^T A_i)(A_j^T A_j)^{-1} + (A_j^T A_j)(A_i^T A_i)^{-1} \right).$$

2. For $k = 1{:}N$ compute

$$A_k V^{-T} = U_k \Sigma_k$$

where the U_k have unit 2-norm columns and the Σ_k are diagonal.

The eigenvalues of S are never smaller than 1.

The Common HO-GSVD Subspace: Definition

The eigenvectors associated with the unit eigenvalues of S_N define the common HO-GSVD subspace:

$$\text{HO-GSVD}(A_1, \ldots, A_N) = \{ v : S_N v = v \}$$

We are able to stably compute this without ever forming S explicitly.

A sequence of 2-matrix GSVDs.

The Common HO-GSVD Subspace: Relevance

In general, we have these rank-1 expansions

$$A_k \;=\; U_k \Sigma_k V^T \;=\; \sum_{i=1}^{n} \sigma_i^{(k)} u_i^{(k)} v_i^T \qquad k = 1{:}N$$

where $V = [v_1, \ldots, v_n]$.

But if (say) the HO-GSVD$(A_1, \ldots, A_N) = \text{span}\{v_1, v_2\}$, then

$$A_k \;=\; \sigma_1 u_1^{(k)} v_1^T \;+\; \sigma_2 u_2^{(k)} v_2^T \;+\; \sum_{i=3}^{n} \sigma_i^{(k)} u_i^{(k)} v_i^T \qquad k = 1{:}N$$

and $\{u_1^{(k)}, u_2^{(k)}\}$ is an orthonormal basis for $\text{span}\{u_3^{(k)}, \ldots, u_n^{(k)}\}^{\perp}$. Moreover, $u_1^{(k)}$ and $u_2^{(k)}$ are left singular vectors for A_k.

This expansion identifies features that are common across the datasets $A_1, \ldots, A_N$.

The Pivoted Cholesky Decomposition

$$
PAP^T = \left[\begin{array}{ccc|ccccc}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{x} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{x} & \mathbf{x} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\
\hline
\mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{1} & 0 & 0 & 0 & 0 \\
\mathbf{x} & \mathbf{x} & \mathbf{x} & 0 & \mathbf{1} & 0 & 0 & 0 \\
\mathbf{x} & \mathbf{x} & \mathbf{x} & 0 & 0 & \mathbf{1} & 0 & 0 \\
\mathbf{x} & \mathbf{x} & \mathbf{x} & 0 & 0 & 0 & \mathbf{1} & 0 \\
\mathbf{x} & \mathbf{x} & \mathbf{x} & 0 & 0 & 0 & 0 & \mathbf{1}
\end{array}\right]
\left[\begin{array}{ccc|ccccc}
\mathbf{d} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{d} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{d} & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & \blacksquare & \times & \times & \times & \times \\
0 & 0 & 0 & \times & \mathbf{x} & \times & \times & \times \\
0 & 0 & 0 & \times & \times & \mathbf{x} & \times & \times \\
0 & 0 & 0 & \times & \times & \times & \mathbf{x} & \times \\
0 & 0 & 0 & \times & \times & \times & \times & \mathbf{x}
\end{array}\right]
\left[\begin{array}{ccc|ccccc}
\mathbf{1} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\
0 & \mathbf{1} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\
0 & 0 & \mathbf{1} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\
\hline
0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1}
\end{array}\right]
$$

We will use this on a problem where the tensor has multiple symmetries and unfolds to a highly structured positive semidefinite matrix with multiple symmetries.

The Two-Electron Integral Tensor (TEI)

Given a basis $\{\phi_i(\mathbf{r})\}_{i=1}^{n}$ of atomic orbital functions, we consider the following order-4 tensor:

$$\mathcal{A}(p,q,r,s) = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \frac{\phi_p(\mathbf{r}_1)\phi_q(\mathbf{r}_1)\phi_r(\mathbf{r}_2)\phi_s(\mathbf{r}_2)}{\|\mathbf{r}_1 - \mathbf{r}_2\|} d\mathbf{r}_1 d\mathbf{r}_2.$$

The TEI tensor plays an important role in electronic structure theory and ab initio quantum chemistry.

The TEI tensor has these symmetries:

$$\mathcal{A}(p,q,r,s) = \begin{cases} \mathcal{A}(q,p,r,s) & \text{(i)} \\ \mathcal{A}(p,q,s,r) & \text{(ii)} \\ \mathcal{A}(r,s,p,q) & \text{(iii)} \end{cases}$$

We say that $\mathcal{A}$ is "((12)(34))-symmetric".

If $A = \mathcal{A}_{[1,2]\times[3,4]}$, then A is symmetric and (among other things) is "perfect shuffle" symmetric.

$$A = \left[\begin{array}{ccc|ccc|ccc} 11 & 12 & 13 & 12 & 14 & 15 & 13 & 15 & 16 \\ 12 & 17 & 18 & 17 & 19 & 20 & 18 & 20 & 21 \\ 13 & 18 & 22 & 18 & 23 & 24 & 22 & 24 & 25 \\ \hline 12 & 17 & 18 & 17 & 19 & 20 & 18 & 20 & 21 \\ 14 & 19 & 23 & 19 & 26 & 27 & 23 & 27 & 28 \\ 15 & 20 & 24 & 20 & 27 & 29 & 24 & 29 & 30 \\ \hline 13 & 18 & 22 & 18 & 23 & 24 & 22 & 24 & 25 \\ 15 & 20 & 24 & 20 & 27 & 29 & 24 & 29 & 30 \\ 16 & 21 & 25 & 21 & 28 & 30 & 25 & 30 & 31 \end{array}\right]$$

Each column reshapes into a 3x3 symmetric matrix, e.g., $A(:,)$ reshapes to

$$\begin{bmatrix} 11 & 12 & 13 \\ 12 & 14 & 15 \\ 13 & 15 & 16 \end{bmatrix}$$

What is perfect shuffle symmetry?

Perfect Shuffle Symmetry

An n^2-by-n^2 matrix A has perfect shuffle symmetry if

$$A = \Pi_{n,n} A \Pi_{n,n}$$

where

$$\Pi_{n,n} \;=\; I_{n^2}(:, v), \qquad v \;=\; [\, 1{:}n{:}n^2 \mid 2{:}n{:}n^2 \mid \;\cdots\; \mid n{:}n{:}n^2 \,].$$

e.g.,

$$\Pi_{3,3} \;=\; \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Structured Low-Rank Approximation

We have an n^2-by-n^2 matrix A that is symmetric and perfect shuffle symmetric and it basically has rank n.

Using $PAP^T = LDL^T$ we are able to write

$$A \;=\; \sum_{k=1}^{n} d_k u_k u_k^T$$

where each rank-1 is symmetric and perfect shuffle symmetric.

This structured data-sparse representation reduces work by an order of magnitude in the application we are considering.

Notation: The Challenge

Scientific computing is increasingly tensor-based.

It is hard to spread the word about tensor computations because summations, transpositions, and symmetries are typically described through multiple indices.

And different camps have very different notations, e.g.

$$t^{i_1 i_2 i_3 i_4 i_5} = a^{i_1}_{j_1} \, b^{i_2}_{j_1 j_2} \, c^{i_2}_{j_2 j_3} \, d^{i_2}_{j_3 j_4} \, e^{i_2}_{j_4}$$

"Brevity is the Soul of Wit"

Multiple Summations

$$\sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{n}} \quad \equiv \quad \sum_{j_1=1}^{n_1} \cdots \sum_{j_d=1}^{n_d}$$

Transposition

If $T = [2\ 1\ 4\ 3]$ then $\mathcal{B} = \mathcal{A}^T$ means

$$\mathcal{B}(i_1, i_2, i_3, i_4) \ = \ \mathcal{A}(i_2, i_1, i_4, i_3)$$

Contractions

For all $\mathbf{1} \le \mathbf{i} \le \mathbf{m}$ and $\mathbf{1} \le \mathbf{j} \le \mathbf{n}$:

$$\mathcal{A}(\mathbf{i}, \mathbf{j}) \ = \ \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{p}} \mathcal{B}(\mathbf{i}, \mathbf{k}) \mathcal{C}(\mathbf{k}, \mathbf{j})$$

From Jacobi's 1846 Eigenvalue Paper

A system of linear equations:

$$\begin{array}{rcl}
(a,a)\alpha + (a,b)\beta + (a,c)\gamma + \cdots + (a,p)\tilde{\omega}' & = & \alpha' x' \\
(b,a)\alpha + (b,b)\beta + (b,c)\gamma + \cdots + (b,p)\tilde{\omega}' & = & \beta' x' \\
\cdots & & \cdots \\
(p,a)\alpha + (p,b)\beta + (p,c)\gamma + \cdots + (p,p)\tilde{\omega}' & = & \tilde{\omega}' x'
\end{array}$$

Somewhere between 1846 and the present we picked up conventional matrix-vector notation: $Ax = b$

How did the transition from scalar notation to matrix-vector notation happen?

The Next Big Thing...

Scalar-Level Thinking

1960's $\Downarrow$ $\Leftarrow$ The factorization paradigm: LU, LDL^T, QR, $U\Sigma V^T$, etc.

Matrix-Level Thinking

1980's $\Downarrow$ $\Leftarrow$ Cache utilization, parallel computing, LAPACK, etc.

Block Matrix-Level Thinking

2000's $\Downarrow$ $\Leftarrow$ High-dimensional modeling, cheap storage, good notation etc.

Tensor-Level Thinking