Four Talks on Tensor Computations

1. Connecting to Matrix Computations

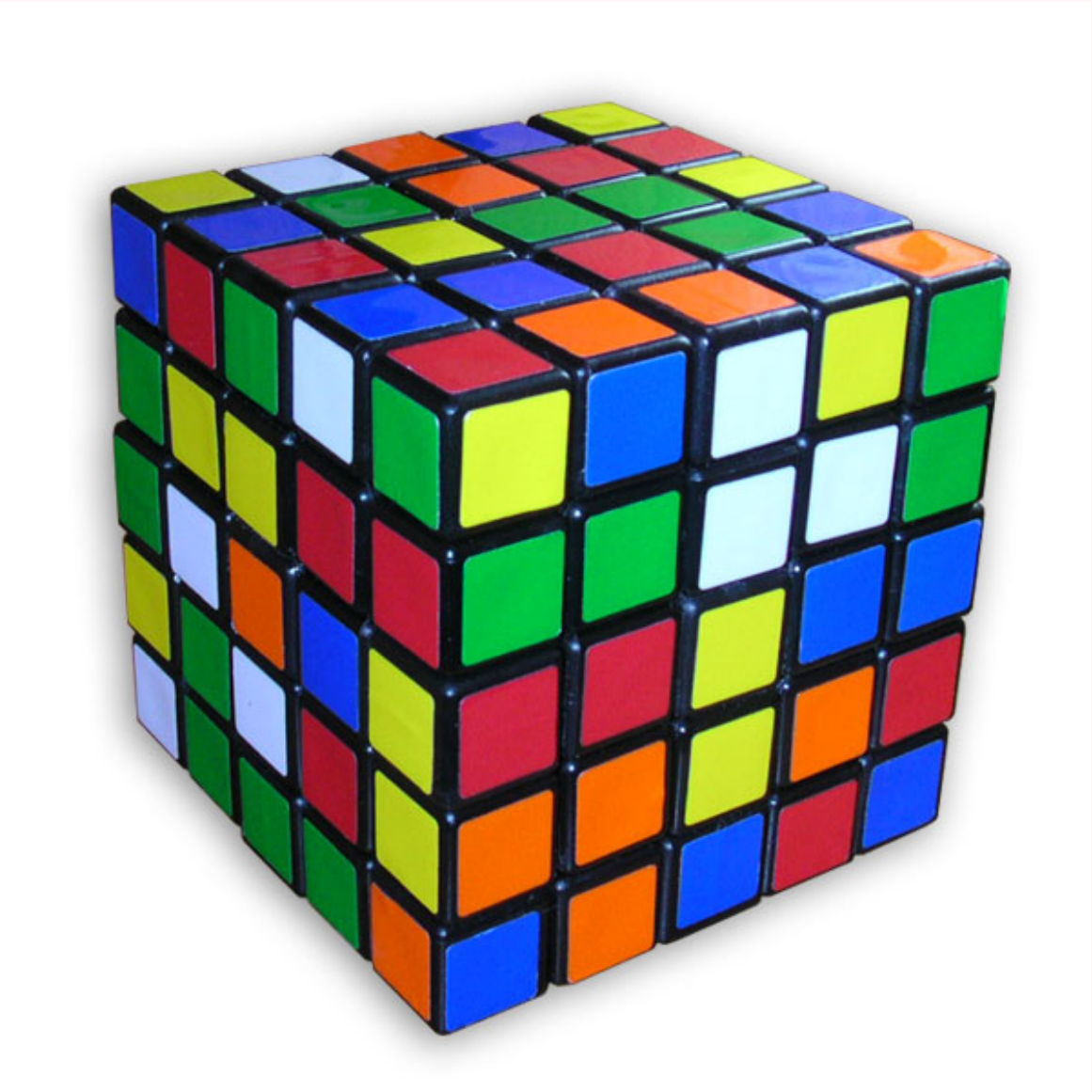
Charles F. Van Loan

Cornell University

SCAN Seminar

October 27, 2014

Its About This



Overview

Preparation for 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 $\Downarrow$ $\Leftarrow$ New applications, factorizations, data structures, nonlinear analysis, optimization strategies, etc.

Tensor-Level Thinking

The Matrix Factorizations Paradigm

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

It's a Language

Some Ways They Are Used

- Conversion of a "hard" matrix problem into an easier one.

$$Ax = b \qquad \equiv \qquad Ly = Pb,\ Ux = y$$

- Low-rank approximation.

$$(10^7\text{-by-}10^5) \quad \approx \quad (10^7\text{-by-}10) \times (10\text{-by-}10^5)$$

- Reasoning about Data Re-Use via Block Factorizations

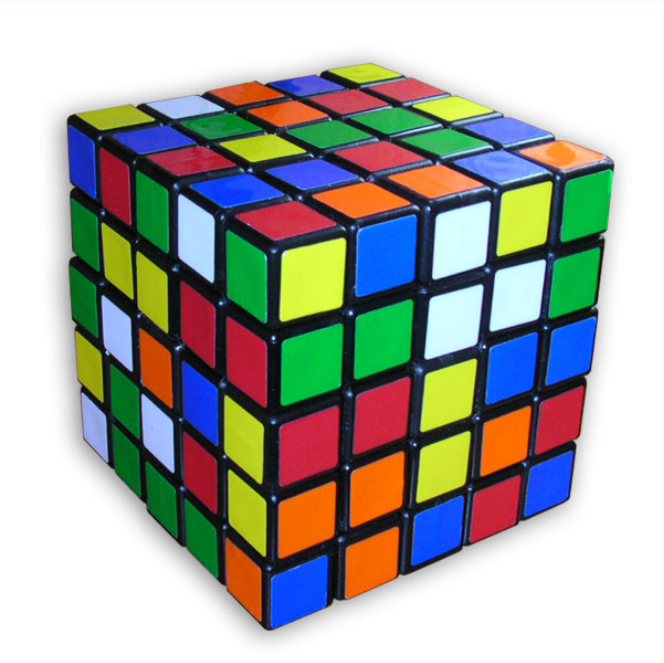
$$A_{ij} \leftarrow A_{ij} - A_{ik}A_{kj} \qquad \text{not} \qquad a_{ij} \leftarrow a_{ij} - a_{ik}a_{kj}$$

- Discovery and Insight

$$\min_{Q,x,y} \| A_1 - (A_2 + xy^T)Q \|_F \leq 10^{-4}$$

Tensor Factorizations and Decompositions

The same story is playing out with tensors:



$$= \sigma_1 \, w_1 \circ v_1 \circ u_1 \; + \; \sigma_2 \, w_2 \circ v_2 \circ u_2 + \; \ldots$$

The Big Data Theme

A Changing 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.

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

Algorithms that scale with d will induce a transition...

Matrix-Based Scientific Computation

$$\Downarrow$$

Tensor-Based Scientific Computation

What is a Tensor?

Definition

An order-d tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a real d-dimensional array $\mathcal{A}(1{:}n_1, \ldots, 1{:}n_d)$ where the index range in the k-th mode is from 1 to n_k. A tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is cubical if $n_1 = \cdots = n_d$.

Low-Order Tensors

A scalar is an order-0 tensor.

A vector is an order-1 tensor.

A matrix is an order-2 tensor.

We use calligraphic font to designate tensors that have order 3 or greater e.g., $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, etc.

Where Might They Come From?

Discretization

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

Multiway Analysis

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

You Have Seen Them Before

Images

A color picture is an m-by-n-by-3 tensor:

$$\begin{aligned} \mathcal{A}(:,:,1) &= \text{red pixel values} \\ \mathcal{A}(:,:,2) &= \text{green pixel values} \\ \mathcal{A}(:,:,3) &= \text{blue pixel values} \end{aligned}$$

Obvious extension of the "colon notation". More later.

You Have Seen them Before

Block Matrices

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

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

$$a_{45} \quad \Leftrightarrow \quad \mathcal{A}(2,3,2,1)$$

You Have Seen them Before

Kronecker Products

The $m_1 m_2 m_3$-by-$n_1 n_2 n_3$ matrix

$$A = A_1(1{:}m_1, 1{:}n_1) \otimes A_2(1{:}m_2, 1{:}n_2) \otimes A_3(1{:}m_3, 1{:}n_3)$$

is a reshaping of the m_1-by-m_2-by-m_3-by-n_1-by-n_2-by-n_3 tensor

$$\mathcal{A}(i_1, i_2, i_3, j_1, j_2, j_3) = A_1(i_1, j_1) A_2(i_2, j_2) A_3(i_3, j_3)$$

The Kronecker Product

Definition

$B \otimes C$ is a *block matrix* whose ij-th block is $b_{ij}C$.

An Example...

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \otimes C = \left[\begin{array}{c|c} b_{11}C & b_{12}C \\ \hline b_{21}C & b_{22}C \end{array}\right]$$

Kronecker products have a replicated block structure.

The Kronecker Product

There are three ways to regard $A = B \otimes C$...

If

$$A = \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & \cdots & c_{1q} \\ \vdots & \ddots & \vdots \\ c_{p1} & \cdots & c_{pq} \end{bmatrix}$$

then

(i). A is an m-by-n block matrix with p-by-q blocks.

(ii). A is an mp-by-nq matrix of scalars.

(iii). A is an unfolded 4-th order tensor $\mathcal{A} \in \mathbb{R}^{p \times q \times m \times n}$:

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

Kronecker Product: All Possible Entry-Entry Products

$4 \times 9 = 36$

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

$$=$$

$$\left[\begin{array}{ccc|ccc} b_{11}c_{11} & b_{11}c_{12} & b_{11}c_{13} & b_{12}c_{11} & b_{12}c_{12} & b_{12}c_{13} \\ b_{11}c_{21} & b_{11}c_{22} & b_{11}c_{23} & b_{12}c_{21} & b_{12}c_{22} & b_{12}c_{23} \\ b_{11}c_{31} & b_{11}c_{32} & b_{11}c_{33} & b_{12}c_{31} & b_{12}c_{32} & b_{12}c_{33} \\ \hline b_{21}c_{11} & b_{21}c_{12} & b_{21}c_{13} & b_{22}c_{11} & b_{22}c_{12} & b_{22}c_{13} \\ b_{21}c_{21} & b_{21}c_{22} & b_{21}c_{23} & b_{22}c_{21} & b_{22}c_{22} & b_{22}c_{23} \\ b_{21}c_{31} & b_{21}c_{32} & b_{21}c_{33} & b_{22}c_{31} & b_{22}c_{32} & b_{22}c_{33} \end{array}\right]$$

Kronecker Products of Kronecker Products

Hierarchical

$$B \otimes C \otimes D = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix} \otimes \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix}$$

A 2-by-2 block matrix whose entries are 4-by-4 block matrices whose entries are 3-by-3 matrices.

Bridging the Gap to Matrix Computations

Tensor computations are typically disguised matrix computations and that is because of

- The Kronecker Product

$$A = A_1 \otimes A_2 \otimes A_3 \qquad \text{an order 6 tensor}$$

- Tensor Unfoldings

$$\text{Rubik Cube} \qquad \longrightarrow \qquad 3 \times 9 \quad \text{matrix}$$

- Alternating Least Squares

  Multilinear optimization via component-wise optimization

The Decomposition Paradigm in Matrix Computations

Typical...

Convert the given problem into an equivalent easy-to-solve problem by using the "right" matrix decomposition.

$$PA = LU, \quad Ly = Pb, \quad Ux = y \quad \Longrightarrow \qquad Ax = b$$

Also Typical...

Uncover hidden relationships by computing the "right" decomposition of the data matrix.

$$A \;=\; U\Sigma V^T \quad \Longrightarrow \qquad A \approx \sum_{i=1}^{\tilde{r}} \sigma_i u_i v_i^T$$

Two Natural Questions

Question 1

Can we solve tensor problems by converting them to equivalent, easy-to-solve problems using a tensor decomposition?

Question 2

Can we uncover hidden patterns in tensor data by computing an appropriate tensor decomposition?

Let's explore the decomposition issue for 2-by-2-by-2 tensors...

The Singular Value Decomposition

The 2-by-2 case...

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

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

A reshaped presentation of the same thing...

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

The reshaped rank-1 matrices have special structure...

The Kronecker Product (KP)

The KP of a Pair of 2-vectors

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \otimes \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 y_1 \\ x_1 y_2 \\ x_2 y_1 \\ x_2 y_2 \end{bmatrix}$$

2-by-2 SVD in KP Terms...

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

Reshaping

Appreciate the Duality..

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \sigma_1 \cdot u_1 v_1^T + \sigma_2 \cdot u_2 v_2^T$$

$$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{bmatrix} = \sigma_1 \cdot (v_1 \otimes u_1) + \sigma_2 \cdot (v_2 \otimes u_1)$$

$$U = \begin{bmatrix} u_1 \mid u_2 \end{bmatrix} \qquad V = \begin{bmatrix} v_1 \mid v_2 \end{bmatrix}$$

What is a rank-1 tensor?

If $\mathcal{R} \in \mathbb{R}^{2\times 2\times 2}$ has unit rank, then there exist $f, g, h \in \mathbb{R}^2$ such that

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

This means that if $\mathcal{R} = (r_{ijk})$, then

$$r_{ijk} = h_k \cdot g_j \cdot f_i \qquad i = 1{:}2,\ j = 1{:}2,\ k = 1{:}2$$

$\mathcal{A} \in \mathbb{R}^{2\times 2\times 2}$ as a minimal sum of rank-1 tensors.

Challenge: Find thinnest possible $X, Y, Z \in \mathbb{R}^{2\times r}$ so

$$\begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} = \sum_{k=1}^{r} z_k \otimes y_k \otimes x_k$$

where $X = [\, x_1 | \cdots | x_r \,]$, $Y = [\, y_1 | \cdots | y_r \,]$, and $Z = [\, z_1 | \cdots | z_r \,]$ are column partitionings.

The minimizing r is the tensor rank.

$\mathcal{A} \in \mathbb{R}^{2\times 2\times 2}$ as a minimal sum of rank-1 tensors.

A Surprising Fact

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

This is Different from the Matrix Case...

If $A = \texttt{randn(n,n)}$, then $\text{rank}(A) = n$ with prob 100%.

What are the "full rank" 2-by-2-by-2 tensors?

The 79/21 Property

The 2-by-2 Generalized Eigenvalue Problem (GEV)

The generalized eigenvalues of $F - \lambda G$ are the zeros of

$$\det\left(\begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} - \lambda \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}\right) = 0$$

Coincidence?

If the a_{ijk} are `randn`, then

$$\det\left(\begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix} - \lambda \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix}\right) = 0$$

has real distinct roots with probability 79% and complex conjugate roots with probability 21%.

What is the connection between the 2-by-2 GEV and the 2-by-2-by-2 tensor rank problem?

Nearness Problems

The Nearest Rank-1 Matrix Problem

If $A \in \mathbb{R}^{n_1 \times n_2}$ has SVD $A = U\Sigma V^T$ then $B_{opt} = \sigma_1 u_1 v_1^T$ minimizes

$$\phi(B) = \| A - B \|_F \qquad \text{rank}(B) = 1.$$

The Nearest Rank-1 Tensor Problem for $\mathcal{A} \in \mathbb{R}^{2\times 2\times 2}$

Find unit 2-norm vectors $u, v, w \in \mathbb{R}^2$ and scalar σ so that $\| a - \sigma \cdot w \otimes v \otimes u \|_2$ is minimized where

$$a = \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix}$$

A Highly Structured Nonlinear Optimization Problem

It Depends on Four Parameters...

$$\phi(\sigma,\theta_1,\theta_2,\theta_3) = \left\| a - \sigma \begin{bmatrix} \cos(\theta_3) \\ \sin(\theta_3) \end{bmatrix} \otimes \begin{bmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{bmatrix} \otimes \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix} \right\|_2$$

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

$$c_i = \cos(\theta_i),\ s_i = \sin(\theta_i) \qquad i = 1{:}3$$

A Highly Structured Nonlinear Optimization Problem

And It Can Be Reshaped...

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

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \sigma \cdot \begin{bmatrix} c_1 \\ s_1 \end{bmatrix} = \sigma \cdot \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix}$$

Idea: Improve σ and θ_1 by minimizing with respect to x_1 and y_1, holding θ_2 and θ_3 fixed.

A Highly Structured Nonlinear Optimization Problem

And It Can Be Reshaped...

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

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \sigma \cdot \begin{bmatrix} c_2 \\ s_2 \end{bmatrix} = \sigma \cdot \begin{bmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{bmatrix}$$

Idea: Improve σ and θ_2 by minimizing with respect to x_2 and y_2, holding θ_1 and θ_3 fixed.

A Highly Structured Nonlinear Optimization Problem

And It Can Be Reshaped...

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

$$\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \sigma \cdot \begin{bmatrix} c_3 \\ s_3 \end{bmatrix} = \sigma \cdot \begin{bmatrix} \cos(\theta_3) \\ \sin(\theta_3) \end{bmatrix}$$

Idea: Improve σ and θ_3 by minimizing with respect to x_3 and y_3, holding θ_1 and θ_2 fixed.

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.

In tensor computations, the "standard matrix problem" that we end up solving is usually the linear least squares problem. In that case, the overall solution process is referred to as alternating least squares.

Bridging the Gap via Tensor Unfoldings

A Common Framework for Tensor Computations...

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

2. Through matrix computations, discover things about A.

3. Draw conclusions about tensor $\mathcal{A}$ based on what is learned about matrix A.

Tensor Unfoldings

Some Tensor Unfoldings of $\mathcal{A} \in \mathbb{R}^{4\times 3\times 2}$

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

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

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

A "tensor unfolding" is sometimes referred to as "tensor flattening" or as a "tensor matricization."

Gradients of Multilinear Forms

If $f:\mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^{n_3} \rightarrow \mathbb{R}$ is defined by

$$f(u,v,w) = \sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\sum_{i_3=1}^{n_3} \mathcal{A}(i_1,i_2,i_3)u(i_1)v(i_2)w(i_3)$$

and $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then its gradient is given by

$$\nabla f(u,v,w) = \begin{bmatrix} \mathcal{A}_{(1)}\, w \otimes v \\ \mathcal{A}_{(2)}\, w \otimes u \\ \mathcal{A}_{(3)}\, v \otimes u \end{bmatrix}$$

where $\mathcal{A}_{(1)}$, $\mathcal{A}_{(2)}$, and $\mathcal{A}_{(3)}$ are the modal unfoldings of $\mathcal{A}$.

More General Unfoldings

Example: $\mathcal{A} = \mathcal{A}(1{:}2, 1{:}3, 1{:}2, 1{:}2, 1{:}3)$

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

A Block Matrix is an Unfolded Order-4 Tensor

Some Natural Identifications

A block matrix with uniformly sized blocks

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1,c_2} \\ \vdots & \ddots & \vdots \\ A_{r_2,1} & \cdots & A_{r_2,c_2} \end{bmatrix} \qquad A_{i_2,j_2} \in \mathbb{R}^{r_1 \times c_1}$$

has several natural reshapings:

$$\begin{aligned} A_{i_1,j_1}(i_2,j_2) &\leftrightarrow \mathcal{A}(i_1,j_1,i_2,j_2) & \mathcal{A} &\in \mathbb{R}^{r_1 \times r_2 \times c_1 \times c_2} \\ A_{i_2,j_2}(i_1,j_1) &\leftrightarrow \mathcal{A}(i_1,j_1,i_2,j_2) & \mathcal{A} &\in \mathbb{R}^{r_1 \times c_1 \times r_2 \times c_2} \end{aligned}$$

Kronecker Products are Unfolded Tensors

The $m_1 m_2 m_3$-by-$n_1 n_2 n_3$ matrix

$$A = A_1(1{:}m_1, 1{:}n_1) \otimes A_2(1{:}m_2, 1{:}n_2) \otimes A_3(1{:}m_3, 1{:}n_3)$$

is a particular unfolding of the m_1-by-m_2-by-m_3-by-n_1-by-n_2-by-n_3 tensor

$$\mathcal{A}(i_1, i_2, i_3, j_1, j_2, j_3) = A_1(i_1, j_1) A_2(i_2, j_2) A_3(i_3, j_3)$$

Parts of a Tensor

Fibers

A fiber of a tensor $\mathcal{A}$ is a vector obtained by fixing all but one $\mathcal{A}$'s indices. For example, if $\mathcal{A} = \mathcal{A}(1{:}3, 1{:}5, 1{:}4, 1{:}7)$, then

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

is a fiber. We adopt the convention that a fiber is a column-vector.

The **vec** Operation

Turns matrices into vectors by stacking columns...

$$X = \begin{bmatrix} 1 & 10 \\ 2 & 20 \\ 3 & 30 \end{bmatrix} \quad \Rightarrow \quad \text{vec}(X) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 10 \\ 20 \\ 30 \end{bmatrix}$$

The **vec** Operation

Turns tensors into vectors by stacking mode-1 fibers...

$$\mathcal{A} \in \mathbb{R}^{2\times 3\times 2} \;\Rightarrow\; \text{vec}(\mathcal{A}) \;=\; \begin{bmatrix} \mathcal{A}(:,1,1) \\ \hline \mathcal{A}(:,2,1) \\ \hline \mathcal{A}(:,3,1) \\ \hline \mathcal{A}(:,1,2) \\ \hline \mathcal{A}(:,2,2) \\ \hline \mathcal{A}(:,3,2) \end{bmatrix} \;=\; \begin{bmatrix} a_{111} \\ a_{211} \\ \hline a_{121} \\ a_{221} \\ \hline a_{131} \\ a_{231} \\ \hline a_{112} \\ a_{212} \\ \hline a_{122} \\ a_{222} \\ \hline a_{132} \\ a_{232} \end{bmatrix}$$

We need to specify the order of the stacking...

Parts of a Tensor

Slices

A slice of a tensor $\mathcal{A}$ is a matrix obtained by fixing all but two of $\mathcal{A}$'s indices. For example, if $\mathcal{A} = \mathcal{A}(1{:}3, 1{:}5, 1{:}4, 1{:}7)$, then

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

is a slice. We adopt the convention that the first unfixed index in the tensor is the row index of the slice and the second unfixed index in the tensor is the column index of the slice.

Tensor Notation: Subscript Vectors

Referencing a Tensor Entry

If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\mathbf{i} = (i_1, \ldots, i_d)$ with $1 \leq i_k \leq n_k$ for $k = 1{:}d$, then

$$\mathcal{A}(\mathbf{i}) \equiv \mathcal{A}(i_1, \ldots, i_k)$$

We say that $\mathbf{i}$ is a subscript vector. Bold font will be used designate subscript vectors.

Bounding Subscript Vectors

If $\mathbf{L}$ and $\mathbf{R}$ are subscript vectors having the same dimension, then $\mathbf{L} \leq \mathbf{R}$ means that $L_k \leq R_k$ for all k.

Special Subscript Vectors

The subscript vector of all ones is denoted by $\mathbf{1}$. (Dimension clear from context.) If N is an integer, then $\mathbf{N} = N \cdot \mathbf{1}$.

Subtensors

Specification of Subtensors

Suppose $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with $\mathbf{n} = [\, n_1, \ldots, n_d \,]$. If $\mathbf{1} \leq \mathbf{L} \leq \mathbf{R} \leq \mathbf{n}$, then $\mathcal{A}(\mathbf{L}{:}\mathbf{R})$ denotes the subtensor

$$B = \mathcal{A}(L_1{:}R_1, \ldots, L_d{:}R_d)$$

The Index-Mapping Function col

Definition

If $\mathbf{i}$ and $\mathbf{n}$ are length-d subscript vectors, then the integer-valued function $\text{col}(\mathbf{i},\mathbf{n})$ is defined by

$$\text{col}(\mathbf{i},\mathbf{n}) = i_1 + (i_2-1)n_1 + (i_3-1)n_1n_2 + \cdots + (i_d-1)n_1\cdots n_{d-1}$$

The Formal Specification of Vec

If $\mathcal{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ and $v = \text{vec}(A)$, then

$$a\,(\text{col}(\mathbf{i},\mathbf{n})) = \mathcal{A}(\mathbf{i}) \qquad \mathbf{1} \le \mathbf{i} \le \mathbf{n}$$

The Importance of Notation

Contractions

$$\mathcal{A}(i_1, i_2, j_1, j_2, j_3) = \sum_{k_1} \sum_{k_2} \mathcal{B}(i_1, i_2, k_1, k_2)\mathcal{C}(k_1, k_2, j_1, j_2, j_3)$$

is better written as

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

Software: The MATLAB Tensor Toolbox

Why?

It provides an excellent environment for learning about tensor computations and for building a facility with multi-index reasoning.

Who?

Brett W. Bader and Tamara G. Kolda, Sandia Laboratories

Where?

`http://csmr.ca.sandia.gov/~tkolda/TensorToolbox/`

MATLAB Tensor Toolbox: **A Tensor is a Structure**

```
>> n = [3 5 4 7];
>> A_array = randn(n);
>> A = tensor(A_array);
>> fieldnames(A)

ans =
    'data'
    'size'
```

The `.data` field is a multi-dimensional array.

The `.size` field is an integer vector that specifies the modal dimensions.

`A.data` and `A_array` have same value.

`A.size` and `n` have the same value.

MATLAB Tensor Toolbox: **Order and Dimension**

```
>> n = [3 5 4 7];
>> A_array = randn(n);
>> A = tensor(A_array)
>> d = ndims(A)

d =
     4

>> n = size(A)
n =

     3     5     4     7
```

MATLAB Tensor Toolbox: **Tensor Operations**

```
  n = [3 5 4 7];
% Familiar initializations...
  X = tenzeros(n); Y = tenones(n);
  A = tenrand(n); B = tenrand(n);
% Extracting Fibers and Slices
  a_Fiber = A(2,:,3,6); a_Slice = A(3,:,2,:);
% These operations are legal...
  C = 3*A;   C = -A;   C = A+1;   C = A.^2;
  C = A + B; C = A./B; C = A.*B; C = A.^B;
% Frobenius norm...
  err = norm(A);
% Applying a Function to Each Entry...
  F = tenfun(@sin,A);
  G = tenfun(@(x) sin(x)./exp(x),A);
```

Problem 1. Suppose $\mathcal{A} = \mathcal{A}(1{:}n_1, 1{:}n_2, 1{:}n_3)$. We say $\mathcal{A}(\mathbf{i})$ is an *interior entry* if $\mathbf{1} < \mathbf{i} < \mathbf{n}$. Otherwise, we say $\mathcal{A}(\mathbf{i})$ is an *edge entry*. If $\mathcal{A}(i_1, i_2, i_3)$ is an interior entry, then it has six *neighbors*: $\mathcal{A}(i_1 \pm 1, i_2, i_3)$,$\mathcal{A}(i_1, i_2 \pm 1, i_3)$, and $\mathcal{A}(i_1, i_2, i_3 \pm 1)$. Implement the following function so that it performs as specified

```
function B = Smooth(A)
% A is a third order tensor
% B is a third order tensor with the property that each
% interior entry B(i) is the average of A(i)'s six
% neighbors. If B(i) is an edge entry, then B(i) = A(i).
```

Strive for an implementation that does not have any loops.

Problem 2. Formulate and solve an order-d version of Problem 1.

Modal Unfoldings Using `tenmat`

Example: A Mode-2 Unfolding of $\mathcal{A} \in \mathbb{R}^{4\times 3\times 2}$

`tenmat(A,2)` sets up $\mathcal{A}_{(2)}$:

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

$$(1,1)\ (2,1)\ (3,1)\ (4,1)\ (1,2)\ (2,2)\ (3,2)\ (4,2)$$

Notice how the fibers are ordered.

Modal Unfoldings Using `tenmat`

Precisely how are the fibers ordered?

If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, $N = n_1 \cdots n_d$, and `B = tenmat(A,k)`, then `B` is the matrix $\mathcal{A}_{(k)} \in \mathbb{R}^{n_k \times (N/n_k)}$ with

$$\mathcal{A}_{(k)}(i_k, \mathrm{col}(\tilde{\mathbf{i}}_\mathbf{k}, \tilde{\mathbf{n}})) \;=\; \mathcal{A}(\mathbf{i})$$

where

$$\begin{aligned} \tilde{\mathbf{i}}_\mathbf{k} &= [i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_d] \\ \tilde{\mathbf{n}}_\mathbf{k} &= [n_1, \ldots, n_{k-1}, n_{k+1}, \ldots, m_d] \end{aligned}$$

Recall that the col function maps multi-indices to integers. For example, if $\mathbf{n} = [2\ 3\ 2]$ *then*

$\mathbf{i}$	$(1,1,1)$	$(2,1,1)$	$1,2,1)$	$(2,2,1)$	$(1,3,1)$	$(2,3,1)$	$(1,1,2)$	$(2,1,2)$	...
$col(\mathbf{i}, \mathbf{n})$	1	2	3	4	5	6	7	8	...

MATLAB Tensor Toolbox: **A Tenmat Is a Structure**

```
>> n = [3 5 4 7];
>> A = tenrand(n);
>> A2 = tenmat(A,2);
>> fieldnames(A2)

ans =

    'tsize'
    'rindices'
    'cindices'
    'data'
```

A2.tsize = size of the unfolded tensor = [3 5 4 7]
A2.rindices = mode indices that define A2's rows = [2]
A2.cindices = mode indices that define A2's columns = [1 3 4]
A2.data = the matrix that is the unfolded tensor.

Other Modal Unfoldings Using `tenmat`

Mode-k Unfoldings of $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$

A particular mode-k unfolding is defined by a permutation $\mathbf{v}$ of $\left[\ 1{:}k-1 \ \middle|\ k+1{:}d\ \right]$. Tensor entries get mapped to matrix entries as follows

$$\mathcal{A}(\mathbf{i}) \ \rightarrow \ A(i_k, \mathrm{col}(\mathbf{i}(\mathbf{v}), \mathbf{n}(\mathbf{v})))$$

Name	**v**	**How to get it...**
$\mathcal{A}_{(k)}$ (Default)	$[1{:}k{-}1 \ \ k{+}1{:}d]$	`tenmat(A,k)`
Forward Cyclic	$[k+1{:}d \ \ 1{:}k{-}1]$	`tenmat(A,k,'fc')`
Backward Cyclic	$[k-1{:}-1{:}1 \ \ d{:}-1{:}k{+}1]$	`tenmat(A,k,'bc')`
Arbitrary	v	`tenmat(A,k,v)`

Using `tenmat`

Problem 3. Write a MATLAB function `alpha = MaxFnorm(A)` that returns the maximum value of $\| B \|_F$ where B is a slice of the input tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$.

Summary

Key Words

- Reshaping is about turning a matrix in to a differently sized matrix or into a tensor or into a vector.
- The Kronecker product is an operation between two matrices that produces a highly structured block matrix.
- A Rank-1 tensor can be reshaped into a multiple Kronecker product of vectors.
- A tensor decomposition expresses a given tensor in terms of simpler tensors.
- The alternating least squares approach to multilinear LS is based on solving a sequence of linear LS problems, each obtained by freezing all but a subset of the unknowns.

Summary

Key Words

- A block matrix is a matrix whose entries are matrices.
- A fiber of a tensor is a column vector defined by fixing all but one index and varying what's left. A slice of a tensor is a matrix defined by fixing all but two indices and varying what's left.
- A mode-k unfolding of a tensor is obtained by assembling all the mode-k fibers into a matrix.
- `tensor` is a Tensor Toolbox command used to construct tensors.
- `tenmat` is a Tensor Toolbox command that is used to construct unfoldings of a tensor.

Summary

Recurring themes...

Given tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, there are many ways to unfold a tensor into a matrix $A \in \mathbb{R}^{N_1 \times N_2}$ where $N_1 N_2 = n_1 \cdots n_d$.

For example, A could be a block matrix whose entries are $\mathcal{A}$-slices.

A facility with block matrices and tensor indexing is required to understand the layout possibilities.

Computations with the unfolded tensor frequently involve the Kronecker product.

References

L. De Lathauwer and B. De Moor (1998). "From Matrix to Tensor: Multilinear Algebra and Signal Processing," in *Mathematics in Signal Processing IV,* J. McWhirter and I. Proudler, eds., Clarendon Press, Oxford, 1–15.

G.H. Golub and C. Van Loan (2013). *Matrix Computations, 4th Edition*, Johns Hopkins University Press, Baltimore, MD.

T.G. Kolda and B.W. Bader (2009). "Tensor Decompositions and Applications," *SIAM Review* 51, 455–500.

P.M. Kroonenberg (2008). *Applied Multiway Data Analysis*, Wiley.

A. Smilde, R. Bro, and P. Geladi (2004). *Multi-Way Analysis: Applications in the Chemical Sciences,* Wiley, West Sussex, England.

B.W. Bader and T.G. Kolda (2006). "Algorithm 862: MATLAB Tensor Classes for Fast Algorithm Prototyping," *ACM Transactions on Mathematical Software*, 32, 635–653.