From Matrix to Tensor:
The Transition to Numerical Multilinear Algebra

Lecture 2. Tensor Unfoldings

Charles F. Van Loan

Cornell University

The Gene Golub SIAM Summer School 2010
Selva di Fasano, Brindisi, Italy

What is this Lecture About?

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.

Let us begin to get ready for this...

What is this Lecture About?

Recurring themes...

Given tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, there are many ways to assemble its entries 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. A portion of Lecture 3 is devoted to this important "bridging the gap" matrix operation.

What is this Lecture About?

Example: Gradients of Multilinear Forms

If $f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^{n_3} \to \mathbb{R}$ 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 matrix unfoldings of $\mathcal{A}$.

What is this Lecture About?

Tensor Unfoldings of $\mathcal{A} \in \mathbb{R}^{4\times3\times2}$

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

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

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

Where We Are

Block Matrices

Definition

A block matrix is a matrix whose entries are matrices.

A 3-by-2 Block Matrix...

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix} \qquad A_{ij} \in \mathbb{R}^{50\times 70}$$

Regarded as a matrix of scalars, A is 150-by-140.

Block Matrices

The blocks in a block matrix do not have to have uniform size.

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \\ \hline A_{31} & A_{32} \end{array}\right] = \left[\begin{array}{c|c} \text{50-by-70} & \text{50-by-10} \\ \hline \text{50-by-70} & \text{50-by-10} \\ \hline \text{20-by-70} & \text{20-by-10} \end{array}\right]$$

Block Matrices: Special Cases

A Block Column Vector...

$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} \qquad A_i \in \mathbb{R}^{100 \times 50}$$

A Block Row Vector...

$$A = \begin{bmatrix} A_1 & A_2 & A_3 \end{bmatrix} \qquad A_i \in \mathbb{R}^{100 \times 50}$$

Block Matrices: Special Cases

A Column-Partitioned Matrix...

$$A = \left[\begin{array}{c|c|c} a_1 & a_2 & a_3 \end{array}\right] \qquad a_i \in \mathbb{R}^{100}$$

A Row-Partitioned Matrix

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix} \qquad a_i \in \mathbb{R}^{100}$$

Block Matrices: Addition

A 3-by-2 Example...

$$\left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \\ \hline B_{31} & B_{32} \end{array}\right] + \left[\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \\ \hline C_{31} & C_{32} \end{array}\right]$$

$$=$$

$$\left[\begin{array}{c|c} B_{11} + C_{11} & B_{12} + C_{12} \\ \hline B_{21} + C_{21} & B_{22} + C_{22} \\ \hline b_{31} + C_{31} & B_{32} + C_{32} \end{array}\right]$$

The matrices must be partitioned conformably.

Block Matrices: Multiplication

An Example...

$$
\left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \\ \hline B_{31} & B_{32} \end{array}\right]
\left[\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right]
$$

$$=$$

$$
\left[\begin{array}{c|c} B_{11}C_{11} + B_{12}C_{21} & B_{11}C_{12} + B_{12}C_{22} \\ \hline B_{21}C_{11} + B_{22}C_{21} & B_{21}C_{12} + B_{22}C_{22} \\ \hline B_{31}C_{11} + B_{32}C_{21} & B_{31}C_{12} + B_{32}C_{22} \end{array}\right]
$$

The matrices must be partitioned conformably.

Block Matrices: Transposition

An Example...

$$\left[\begin{array}{cc} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \end{array}\right]^{T} = \left[\begin{array}{ccc} B_{11}^{T} & B_{21}^{T} & B_{31}^{T} \\ B_{12}^{T} & B_{22}^{T} & B_{32}^{T} \end{array}\right]$$

Different from Block Transposition...

$$\left[\begin{array}{cc} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \end{array}\right]^{[T]} = \left[\begin{array}{ccc} B_{11} & B_{21} & B_{31} \\ B_{12} & B_{22} & B_{32} \end{array}\right]$$

(*More on this later.*)

Recursive Block Structure

A block matrix can have block matrix entries.

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

$$A_{11} = \begin{bmatrix} A_{1111} & A_{1112} \\ A_{1121} & A_{1122} \end{bmatrix} \qquad A_{12} = \begin{bmatrix} A_{1211} & A_{1212} \\ A_{1221} & A_{1222} \end{bmatrix}$$

$$A_{21} = \begin{bmatrix} A_{2111} & A_{2112} \\ A_{2121} & A_{2122} \end{bmatrix} \qquad A_{22} = \begin{bmatrix} A_{2211} & A_{2212} \\ A_{2221} & A_{2222} \end{bmatrix}$$

Example: The Discrete Fourier Transform Matrix F_n

$$F_{2m}P = \begin{bmatrix} F_m & \Omega \cdot F_m \\ F_m & -\Omega \cdot F_m \end{bmatrix} \qquad P = \text{permutation}, \ \Omega = \text{diagonal}$$

The vec Operation

Turns matrices into vectors by stacking columns...

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

MATLAB: **Vec of a matrix using the `reshape` function.**

```
% If A is a matrix, then the following
% assigns vec(A) to a ...
[n1,n2] = size(A);
a = reshape(A,n1*n2,1);
```

It can be shown that

$$a(i_1 + (i_2 - 1)n_1)$$

$$=$$

$$A(i_1, i_2)$$

The **vec** Operation

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

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

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

Tensor Notation: Subscript Vectors

Reference

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.

Bounds

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 Cases

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

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

$$\text{col}(\mathbf{i}, \mathbf{n}) \;=\; i_1 + (i_2 - 1)n_1 + (i_3 - 1)n_1 n_2 + \cdots + (i_d - 1)n_1 \cdots n_{d-1}$$

The Formal Specification of Vec

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

$$a\left(\text{col}(\mathbf{i}, \mathbf{n})\right) \;=\; \mathcal{A}(\mathbf{i}) \qquad \mathbf{1} \leq \mathbf{i} \leq \mathbf{n}$$

MATLAB: **Vec of a tensor using the `reshape` function**

```
% If A is a n(1) x...x n(d), then the
% following assigns vec(A) to a ...
n = size(A);
a = reshape(A,prod(n),1);
```

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

$$a(i_1 + (i_2 - 1)n_1 + (i_3 - 1)n_1 n_2 + \ldots + (i_d - 1)n_1 \cdots \ldots \cdots n_{d-1})$$

$$=$$

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

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.

Problem 2.1. How many fibers are there in the tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$?

MATLAB: **Fiber Extraction**

```
n = [3 5 4 7];
A = randn(n);

v0 = A(2,:,4,6);          % v0(1,j,1,1) = A(2,j,4,6)
v1 = squeeze(v0);         % v1(j) = A(2,j,4,6)
[r,c] = size(v1);
if r==1
   v = v1';
else
   v = v1;
end
```

The function `squeeze` removes "singleton dimensions". In the above, `v0` is a fourth-order tensor that has dimension 1 in modes 1, 3, and 4. Sometimes `squeeze` produces a row vector. The `if-else` is necessary because (by convention) we insist that fibers are column vectors.

Problem 2.2. Write a recursive MATLAB function `alpha = Tensor1norm(A)` that returns the maximum value of $\| f \|_1$ where f is a fiber of the input tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$.

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.

Problem 2.3. How many slices are there in the tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ if $n_1 = \cdots = n_d = N$?

MATLAB: **Slice Extraction**

```
n = [3 5 4 7];
A = randn(n);

C0 = A(2,:,4,:);      % C0(2,i,4,j) = A(2,i,4,j)
C = squeeze(C0);      % C(i,j) = A(2,i,4,j)
```

In the above, C0 is a fourth-order tensor that has dimension 1 in modes 1 and 3. C is a 5-by-7 matrix, a.k.a., 2nd order tensor.

Problem 2.4. Write a recursive MATLAB function `alpha = MaxFnorm(A)` that returns the maximum value of $\parallel B \parallel_F$ where B is a slice of the input tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$.

Subtensors

Subscript Vectors

If $\mathbf{i} = [\, i_1, \ldots, i_d \,]$ and $\mathbf{j} = [\, j_1, \ldots, j_d \,]$ are integer vectors, then $\mathbf{i} \le \mathbf{j}$ means that $i_k \le j_k$ for $k = 1{:}d$.

Suppose $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with $\mathbf{n} = [\, n_1, \ldots, n_d \,]$. If $\mathbf{1} \le \mathbf{i} \le \mathbf{n}$, then

$$\mathcal{A}(\mathbf{i}) = \mathcal{A}(i_1, \ldots, i_d).$$

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 $\mathbf{1} \le \mathbf{L} \le \mathbf{R} \le \mathbf{n}$, then $\mathcal{A}(\mathbf{L}{:}\mathbf{R})$ denotes the subtensor

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

MATLAB: Assignment to Fibers, Slices, and Subtensors

```
n = [3 5 4 7];

A = zeros(n);

A(2,5,:,6) = ones(4,1);

A(1,:,3,:) = rand(5,7);

A(2:3,4:5,1:2,5:7) = zeros(2,2,2,3);
```

Problem 2.5. Write a MATLAB function `A = Sym4(N)` that returns a random 4th order tensor $\mathcal{A} \in \mathbb{R}^{N \times N \times N \times N}$ with the property that $\mathcal{A}(\mathbf{i}) = \mathcal{A}(\mathbf{j})$ whenever $\mathbf{j}(1{:}2)$ is a permutation of $\mathbf{i}(1{:}2)$ and $\mathbf{j}(3{:}4)$ is a permutation of $\mathbf{i}(3{:}4)$.

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?

Bruce W. Bader and Tammy G. Kolda, Sandia Laboratories

Where?

`http://csmr.ca.sandia.gov/ tilde 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. In the above, `A.data` and `A_array` have same value and `A.size` and `n` have the same value.

A First Example

The Frobenius norm of a 3-tensor

```
function sigma = NormFro3(A)
n = A.size;
s = 0;
for i1=1:n(1)
   for i2=1:n(2)
      for i3=1:n(3)
         s = s + A(i1,i2,i3)^2;
      end
   end
end
s = sqrt(s);
```

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
```

Problem 2.6. If `A` is a d-dimensional array in `Matlab`, then what is `sum(A)`? Explain why the following function returns the Frobenius norm of the input tensor.

```
  function sigma = NormFro(A)
% A is a tensor.
% sigma is the square root of the sum of the
% squares of its entries.
  B = A.data.^2;
  for k=1:ndims(A)
     B = sum(B);
  end
  sigma = sqrt(B);
```

MATLAB Tensor Toolbox: **Tensor Operations**

```
n = [3 5 4 7];
% Extracting Fibers and Slices
a_Fiber = A(2,:,3,6); a_Slice = A(3,:,2,:);

% Familiar initializations...
X = tenzeros(n); Y = tenones(n);
A = tenrandn(n); B = tenrandn(n);

% 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;

% Applying a Function to Each Entry...
F = tenfun(@sin,A);
G = tenfun(@(x) sin(x)./exp(x),A);
```

Problem 2.7. 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 2.8. Formulate and solve an order-d version of Problem 2.7.

All Tensors Secretly Wish that They Were Matrices!

Example 1. $\mathcal{A} = \mathcal{A}(1{:}n_1, 1{:}n_2, 1{:}n_3)$

$$\text{ima\_matrix} = \left[\; \mathcal{A}(1, :, :) \;\middle|\; \cdots \;\middle|\; \mathcal{A}(n_1, :, :) \;\right]$$

$$\text{ima\_matrix} = \left[\; \mathcal{A}(:, 1, :) \;\middle|\; \cdots \;\middle|\; \mathcal{A}(:, n_2, :) \;\right]$$

$$\text{ima\_matrix} = \left[\; \mathcal{A}(:, :, 1) \;\middle|\; \cdots \;\middle|\; \mathcal{A}(:, :, n_3) \;\right]$$

All Tensors Secretly Wish that They Were Matrices!

Example 2. $\mathcal{A} = \mathcal{A}(1{:}n_1, 1{:}n_2, 1{:}2, 1{:}2, 1{:}2)$

$$\text{ima\_matrix} = \begin{bmatrix} \mathcal{A}(:,:,1,1,1) \\ \mathcal{A}(:,:,2,1,1) \\ \mathcal{A}(:,:,1,2,1) \\ \mathcal{A}(:,:,2,2,1) \\ \mathcal{A}(:,:,1,1,2) \\ \mathcal{A}(:,:,2,1,2) \\ \mathcal{A}(:,:,1,2,2) \\ \mathcal{A}(:,:,2,2,2) \end{bmatrix}$$

Example 3. $\mathcal{A} = \mathcal{A}(1{:}n_1, 1{:}n_2, 1{:}n_3, 1{:}n_4)$

$$\text{ima\_matrix} = \begin{bmatrix} \mathcal{A}(\,:\,,\,:\,,1,1) & \mathcal{A}(\,:\,,\,:\,,1,2) & \cdots & \mathcal{A}(\,:\,,\,:\,,1,n_4) \\ \mathcal{A}(\,:\,,\,:\,,2,1) & \mathcal{A}(\,:\,,\,:\,,2,2) & \cdots & \mathcal{A}(\,:\,,\,:\,,2,n_4) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}(\,:\,,\,:\,,n_3,1) & \mathcal{A}(\,:\,,\,:\,,n_3,2) & \cdots & \mathcal{A}(\,:\,,\,:\,,n_3,n_4) \end{bmatrix}$$

$\mathcal{A}(i,j,k,\ell)$ is the (i,j) entry of block (k,ℓ)

What are they?

A tensor unfolding of $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is obtained by assembling $\mathcal{A}$'s entries into a matrix $A \in \mathbb{R}^{N_1 \times N_2}$ where $N_1 N_2 = n_1 \cdots n_d$.

Obviously, there are many ways to unfold a tensor.

An important family of tensor unfoldings are the mode-k unfoldings.

In a mode-k unfolding, the mode-k fibers are assembled to produce an n_k-by-(N/n_k) matrix where $N = n_1 \cdots n_d$.

The tensor toolbox function `tenmat` can be used to produce modal unfoldings and other, more general unfoldings.

Modal Unfoldings Using `tenmat`

Example: A Mode-1 Unfolding of $\mathcal{A} \in \mathbb{R}^{4\times 3\times 2}$

`tenmat(A,1)` sets up

$$\begin{array}{c} \begin{bmatrix} a_{111} & a_{121} & a_{131} & a_{112} & a_{122} & a_{132} \\ a_{211} & a_{221} & a_{231} & a_{212} & a_{222} & a_{232} \\ a_{311} & a_{321} & a_{331} & a_{312} & a_{322} & a_{332} \\ a_{411} & a_{421} & a_{431} & a_{412} & a_{422} & a_{432} \end{bmatrix} \\ \begin{array}{cccccc} (1,1) & (2,1) & (3,1) & (1,2) & (2,2) & (3,2) \end{array} \end{array}$$

Notice how the fibers are ordered.

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

$$\begin{array}{cccccccc} \left[\begin{array}{cccccccc} a_{111} & a_{211} & a_{311} & a_{411} & a_{112} & a_{212} & a_{312} & a_{412} \\ a_{121} & a_{221} & a_{321} & a_{421} & a_{122} & a_{222} & a_{322} & a_{422} \\ a_{131} & a_{231} & a_{331} & a_{431} & a_{132} & a_{232} & a_{332} & a_{432} \end{array}\right] \\ (1,1) \quad (2,1) \quad (3,1) \quad (4,1) \quad (1,2) \quad (2,2) \quad (3,2) \quad (4,2) \end{array}$$

Notice how the fibers are ordered.

Modal Unfoldings Using `tenmat`

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

`tenmat(A,3)` sets up

$$\begin{array}{cccccccccccc} \left[\begin{array}{cccccccccccc} a_{111} & a_{211} & a_{311} & a_{411} & a_{121} & a_{221} & a_{321} & a_{421} & a_{131} & a_{231} & a_{331} & a_{431} \\ a_{112} & a_{212} & a_{312} & a_{412} & a_{122} & a_{222} & a_{322} & a_{422} & a_{132} & a_{232} & a_{332} & a_{432} \end{array}\right] \\ (1,1) \; (2,1) \; (3,1) \; (4,1) \; (1,2) \; (2,2) \; (3,2) \; (4,2) \; (1,3) \; (2,3) \; (3,3) \; (4,3) \end{array}$$

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

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

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

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

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 $[1{:}k-1 \;\; k+1{:}d]$. 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	$\mathbf{v}$	**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)`

Problem 2.9. Given that $\mathcal{A} = \mathcal{A}(1{:}n_1, 1{:}n_2, 1{:}n_3)$, show how `tenmat` can be used to compute the following unfoldings:

$$A_1 = \left[\ \mathcal{A}(1, :, :)\ \middle|\ \cdots\ \middle|\ \mathcal{A}(n_1, :, :)\ \right]$$

$$A_2 = \left[\ \mathcal{A}(:, 1, :)\ \middle|\ \cdots\ \middle|\ \mathcal{A}(:, n_2, :)\ \right]$$

$$A_3 = \left[\ \mathcal{A}(:, :, 1)\ \middle|\ \cdots\ \middle|\ \mathcal{A}(:, :, n_3)\ \right]$$

Problem 2.10. How can `tenmat` be used to compute the vec of a tensor?

A Block Matrix is an Unfolded 4th Order 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:

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

The function `tenmat` *can be used to set up these unfoldings...*

A 4th Order Tensor is a Block Matrix

Example 1.

If $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$, then `tenmat(A,[1 2],[3 4]` sets up

$$\begin{bmatrix} \mathcal{A}(1,1,:,:) & \cdots & \mathcal{A}(1,n_2,:,:) \\ \vdots & \ddots & \vdots \\ \mathcal{A}(n_1,1,:,:) & \cdots & \mathcal{A}(n_1,n_2,:,:) \end{bmatrix}$$

Example 2.

If $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$, then `tenmat(A,[1 3],[2 4]` sets up

$$\begin{bmatrix} \mathcal{A}(1,:,1,:) & \cdots & \mathcal{A}(1,:,n_3,:) \\ \vdots & \ddots & \vdots \\ \mathcal{A}(n_1,:,1,:) & \cdots & \mathcal{A}(n_1,:,n_3,:) \end{bmatrix}$$

Even More General Unfoldings

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

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

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

MATLAB Tensor Toolbox: **General Unfoldings Using** `tenmat`

```
   function A_unfolded = Unfold(A,rIdx,cIdx)
 % A = A(1:n(1),...,1:n(d)) is a tensor.
 % rIdx and cIdx are integer vectors with the
 %    property that [rIdx cIdx] is a
%     permutation of 1:d.
 %
 % A_unfolded is an nRows-by-nCols matrix where
 %        nRows = prod(n(rIdx))
 %        nCols = prod(n(cIdx))
 % if 1<=i<=n, then
 %        A(i) = A_unfolded(r,c)
 % with
 %        r = col(i(rIdx),n(rIdx))
 %        c = col(i(cIdx),n(cIdx))
   A_unfolded = tenmat(A,rIdx,cIdx)
```

Summary of Lecture 2.

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.