Machine Learning

CSE 6363 (Fall 2016)

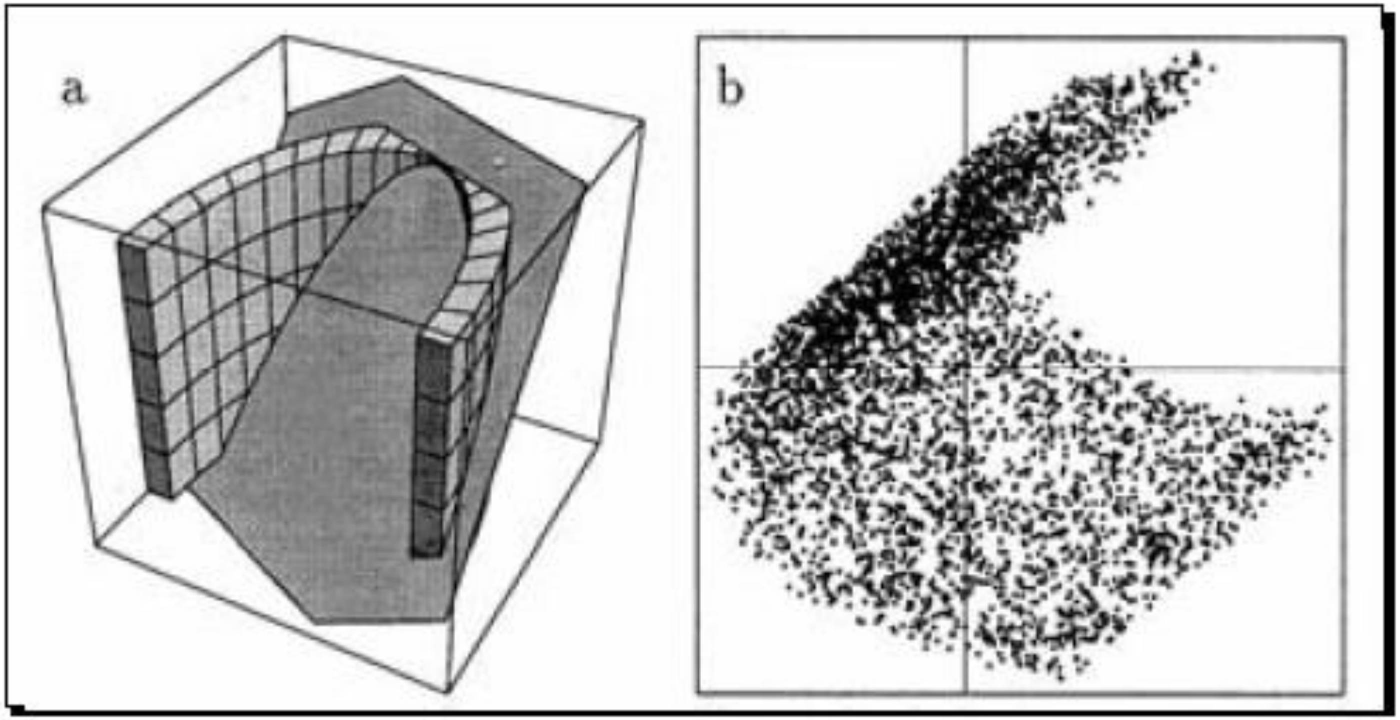
Lecture 14 Kernel PCA and Tensor

Heng Huang, Ph.D.

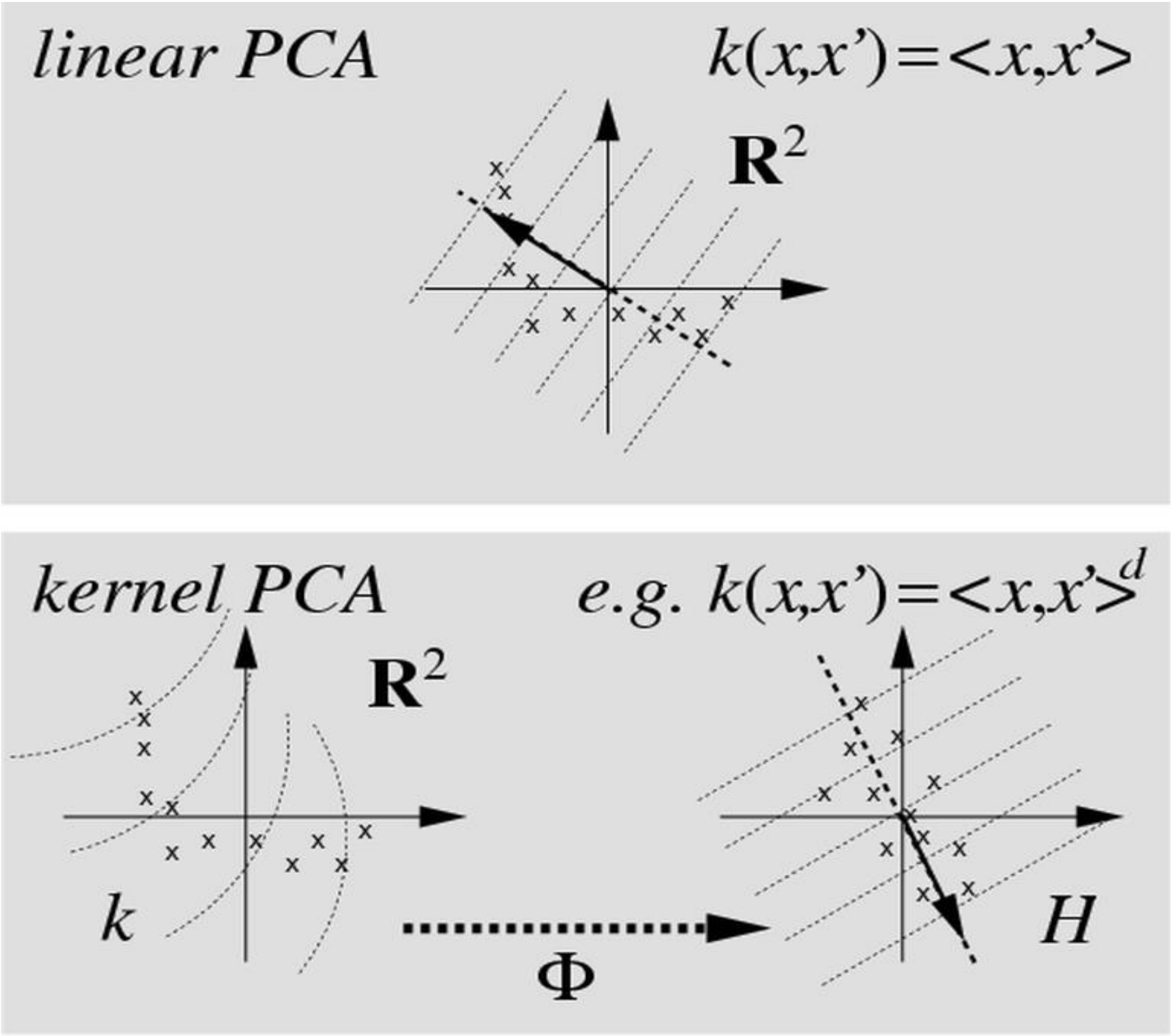
Department of Computer Science and Engineering

Limitations of Linear PCA

$\lambda_{1,2,3}$=1/3



Kernel PCA: The Main Idea



PCA Revisit

Sample covariance matrix:

$$C = \frac{1}{N}\sum_i \mathbf{x}_i \mathbf{x}_i^T$$

Linear PCA:

$$U\Lambda U^T = C \quad \Rightarrow \quad C = \sum_a \lambda_a \mathbf{u}_a \mathbf{u}_a^T$$

The projection is given by:

$$\mathbf{y}_i = U_k^T \mathbf{x}_i \quad \forall i$$

The projected data are de-correlated:

$$\frac{1}{N}\sum_i \mathbf{y}_i \mathbf{y}_i^T = \frac{1}{N}\sum_i U_k^T \mathbf{x}_i \mathbf{x}_i^T U_k = U_k^T C U_k = U_k^T U \Lambda U^T U_k^T = \Lambda_k$$

Different point of view of PCA: $\sum_i ||\mathbf{x}_i - \mathcal{P}_k \mathbf{x}_i||^2$

Kernel Method in PCA

The eigen-vectors that span the projection space must lie in the subspace spanned by the data-cases:

$$\lambda_a \mathbf{u}_a = C\mathbf{u}_a = \frac{1}{N}\sum_i \mathbf{x}_i \mathbf{x}_i^T \mathbf{u}_a = \frac{1}{N}\sum_i (\mathbf{x}_i^T \mathbf{u}_a)\mathbf{x}_i$$

$$\Rightarrow \quad \mathbf{u}_a = \sum_i \frac{(\mathbf{x}_i^T \mathbf{u}_a)}{N\lambda_a}\mathbf{x}_i = \sum_i \alpha_i^a \mathbf{x}_i$$

PCA eigenvalue function: $C\mathbf{u} = \lambda \mathbf{u}$

N equations as:

$$\mathbf{x}_i^T C \mathbf{u}_a = \lambda_a \mathbf{x}_i^T \mathbf{u}_a \quad \Rightarrow$$

$$\mathbf{x}_i^T \frac{1}{N}\sum_k \mathbf{x}_k \mathbf{x}_k^T \sum_j \alpha_j^a \mathbf{x}_j = \lambda_a \mathbf{x}_i^T \sum_j \alpha_j^a \mathbf{x}_j \quad \Rightarrow$$

$$\frac{1}{N}\sum_{j,k} \alpha_j^a [\mathbf{x}_i^T \mathbf{x}_k][\mathbf{x}_k^T \mathbf{x}_j] = \lambda_a \sum_j \alpha_j^a [\mathbf{x}_i^T \mathbf{x}_j]$$

Kernel Method in PCA

Define the matrix $[\mathbf{x}_i^T \mathbf{x}_j] = K_{ij}$

$$K^2 \boldsymbol{\alpha}^a = N\lambda_a K \boldsymbol{\alpha}^a$$

$$\Rightarrow \qquad K\boldsymbol{\alpha}^a = (\tilde{\lambda}_a)\boldsymbol{\alpha}^a \qquad\qquad \tilde{\lambda}_a = N\lambda_a$$

Orthonormal constraint:

$$\mathbf{u}_a^T \mathbf{u}_a = 1$$

$$\Rightarrow \quad \sum_{i,j} \alpha_i^a \alpha_j^a [\mathbf{x}_i^T \mathbf{x}_j] = \boldsymbol{\alpha}_a^T K \boldsymbol{\alpha}_a = N\lambda_a \boldsymbol{\alpha}_a^T \boldsymbol{\alpha}_a = 1$$

$$\Rightarrow \qquad ||\boldsymbol{\alpha}_a|| = 1/\sqrt{N\lambda_a}$$

Kernel PCA projection:

$$\mathbf{u}_a^T \mathbf{t} = \sum_i \alpha_i^a \mathbf{x}_i^T \mathbf{t} = \sum_i \alpha_i^a K(\mathbf{x}_i, \mathbf{t})$$

Center Kernel PCA

It is difficult to center the data in future space. But we can center kernel matrix.

$$K_{ij} = \Phi_i \Phi_j^T$$

$$\Phi_i = \Phi_i - \frac{1}{N}\sum_k \Phi_k$$

$$\begin{aligned} K_{ij}^c &= (\Phi_i - \frac{1}{N}\sum_k \Phi_k)(\Phi_j - \frac{1}{N}\sum_l \Phi_l)^T \\ &= \Phi_i \Phi_j^T - [\frac{1}{N}\sum_k \Phi_k]\Phi_j^T - \Phi_i[\frac{1}{N}\sum_l \Phi_l^T] + [\frac{1}{N}\sum_k \Phi_k][\frac{1}{N}\sum_l \Phi_l^T] \\ &= K_{ij} - \boldsymbol{\kappa}_i \mathbf{1}_j^T - \mathbf{1}_i \boldsymbol{\kappa}_j^T + k \mathbf{1}_i \mathbf{1}_j^T \end{aligned}$$

$$\boldsymbol{\kappa}_i = \frac{1}{N}\sum_k K_{ik}$$

$$k = \frac{1}{N^2}\sum_{ij} K_{ij}$$

Center Kernel PCA

At test-time, we need compute:

$$K_c(\mathbf{t}_i, \mathbf{x}_j) = [\Phi(\mathbf{t}_i) - \frac{1}{N}\sum_k \Phi(\mathbf{x}_k)][\Phi(\mathbf{x}_j) - \frac{1}{N}\sum_l \Phi(\mathbf{x}_l)]^T$$

Similar to the derivations in last slide, we have:

$$K_c(\mathbf{t}_i, \mathbf{x}_j) = K(\mathbf{t}_i, \mathbf{x}_j) - \boldsymbol{\kappa}(\mathbf{t}_i)\mathbf{1}_j^T - \mathbf{1}_i\boldsymbol{\kappa}(\mathbf{x}_j)^T + k\mathbf{1}_i\mathbf{1}_j^T$$

Algorithm

Input: Data $X=\{x_1, x_2, \ldots, x_l\}$ in n-dimensional space.

Process: $K_{i,j}= k(x_i,x_j)$; $i,j=1,\ldots, l$.   Kernel matrix ...

$$\hat{K} = K - \frac{1}{l} j \cdot j' \cdot K - \frac{1}{l} K \cdot j \cdot j' + \frac{1}{l^2} (j' \cdot K \cdot j) \cdot j \cdot j';$$

… for centered data

$$[W, \Lambda] = \text{eig}(\hat{K});$$

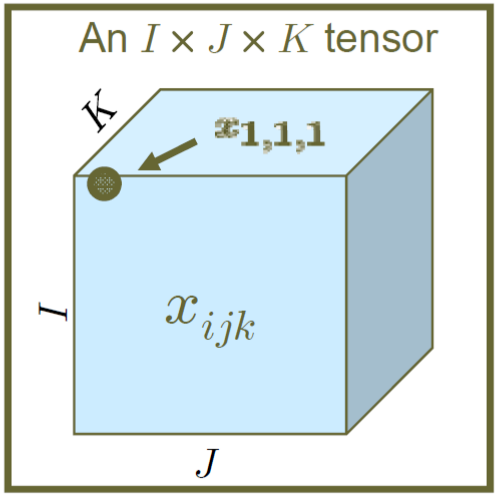
$$\alpha^{(j)} = \frac{1}{\sqrt{\lambda_j}} w_j, \quad j = 1,\ldots,l.$$

$$\tilde{x}_j = \left(\sum_{i=1}^{l} \alpha_i^{(j)} k(x_i, x) \right)_{j=1}^{k}$$

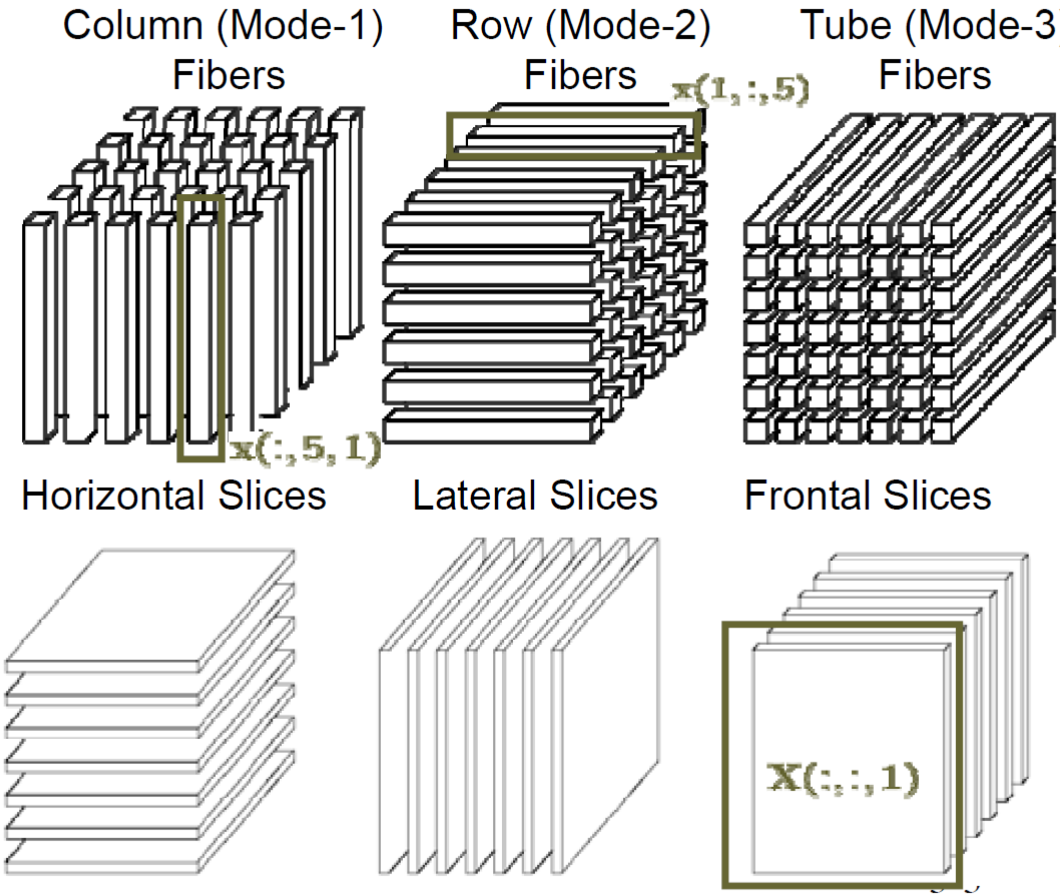
k-dimensional vector projection of new data into this subspace

Output: Transformed data

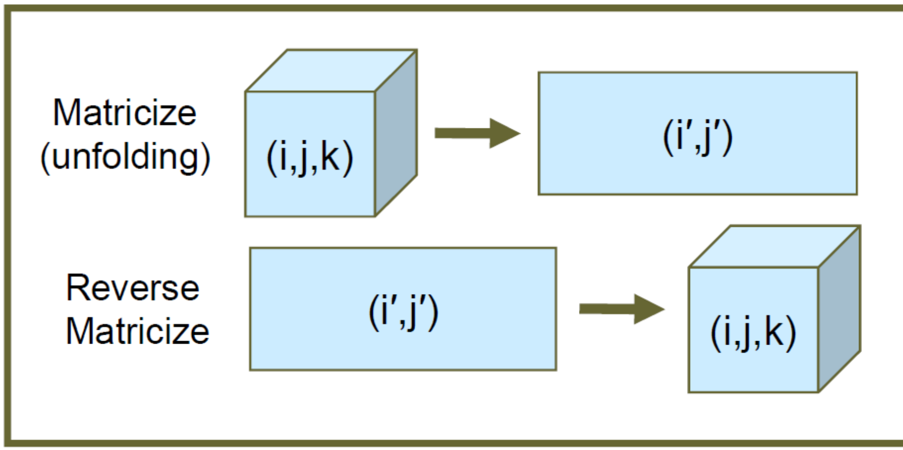
A Tensor Is A Multidimensional Array



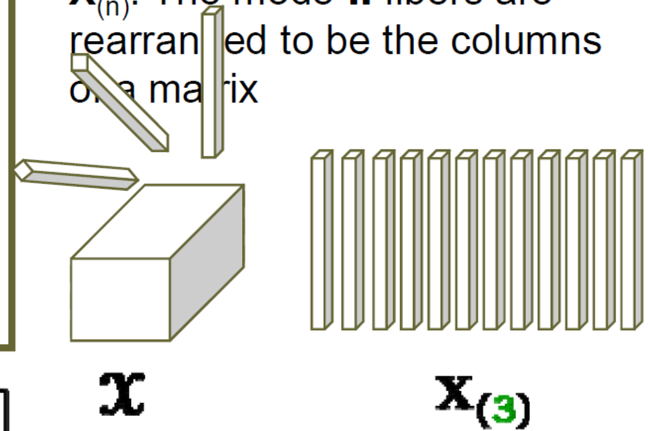
3rd order tensor
mode 1 has dimension I
mode 2 has dimension J
mode 3 has dimension K



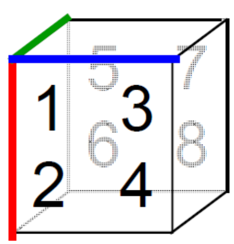
Matricization : Converting Tensor to Matrix



$\mathbf{X}_{(n)}$: The mode-**n** fibers are rearranged to be the columns of a matrix



$\mathcal{X} =$



$$\mathbf{X}_{(1)} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix}$$

$$\mathbf{X}_{(2)} = \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{bmatrix}$$

$$\mathbf{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

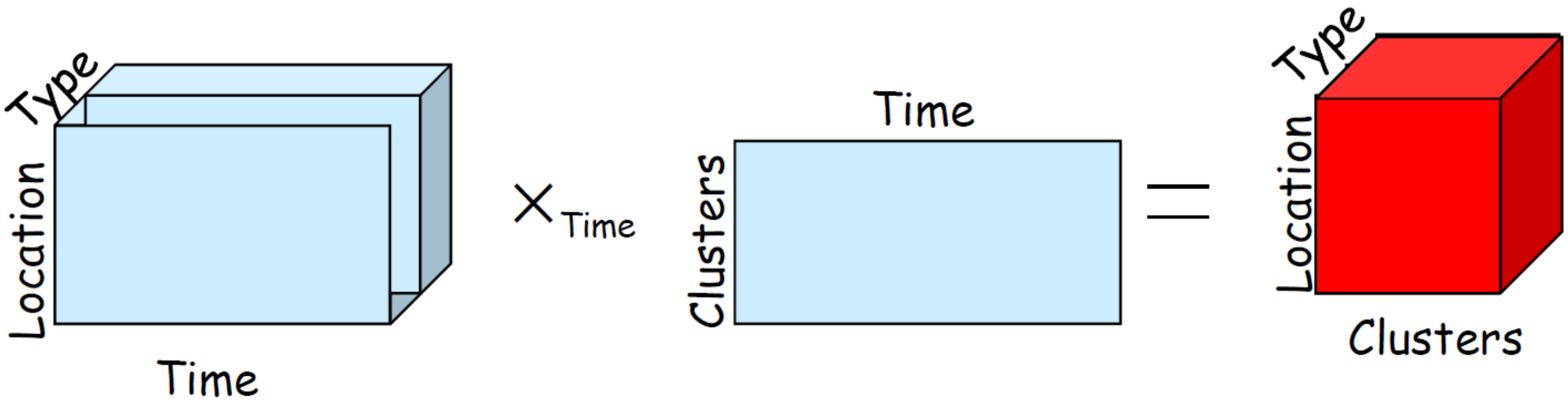
Vectorization

$$\mathrm{vec}(\mathcal{X}) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{bmatrix}$$

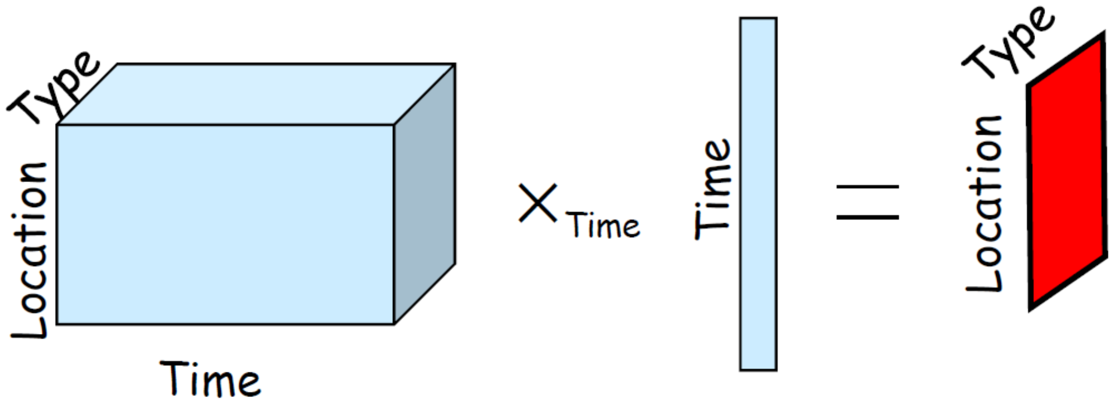
3-4

Mode-n Product Example

Tensor times a matrix



Tensor times a vector

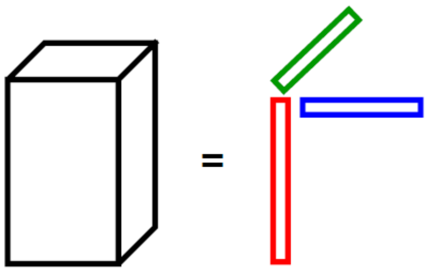


Outer, Kronecker, & Khatri-Rao Products

3-Way Outer Product

$$\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$$

$$x_{ijk} = a_i b_j c_k$$



Rank-1 Tensor

Review: Matrix Kronecker Product

$$\underset{M \times N}{\mathbf{A}} \otimes \underset{P \times Q}{\mathbf{B}} = \underset{MP \times NQ}{\begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1N}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2N}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1}\mathbf{B} & a_{M2}\mathbf{B} & \cdots & a_{MN}\mathbf{B} \end{bmatrix}}$$

$$= \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_1 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_N \otimes \mathbf{b}_Q \end{bmatrix}$$

Matrix Khatri-Rao Product

$$\underset{M \times R}{\mathbf{A}} \odot \underset{N \times R}{\mathbf{B}} = \underset{MN \times R}{\begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_R \otimes \mathbf{b}_R \end{bmatrix}}$$

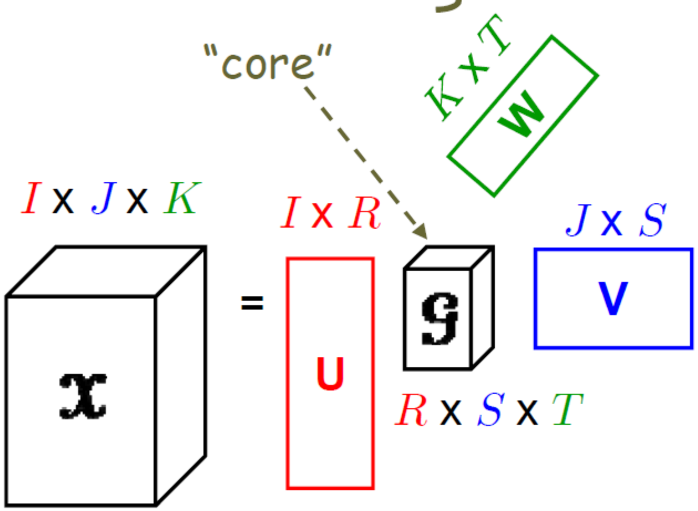
Observe: For two vectors **a** and **b**, **a** $\circ$ **b** and **a** $\otimes$ **b** have the same elements, but one is shaped into a matrix and the other into a vector.

Specially Structured Tensors

- Tucker Tensor

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$$
$$= \sum_r \sum_s \sum_t g_{rst}\, \mathbf{u}_r \circ \mathbf{v}_s \circ \mathbf{w}_t$$
$$\equiv [\![\mathcal{G} \,;\, \mathbf{U}, \mathbf{V}, \mathbf{W}]\!]$$

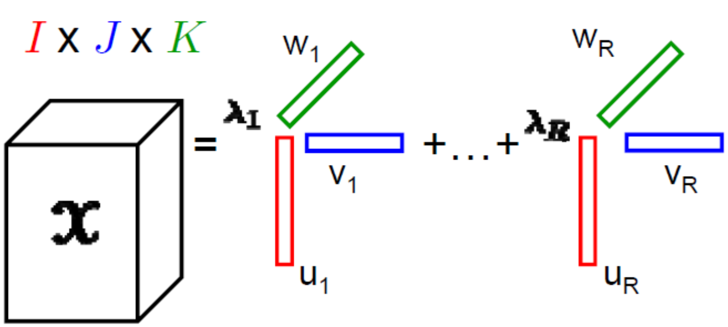
} Our Notation



- Kruskal Tensor

$$\mathcal{X} = \sum_r \lambda_r\, \mathbf{u}_r \circ \mathbf{v}_r \circ \mathbf{w}_r$$
$$\equiv [\![\lambda \,;\, \mathbf{U}, \mathbf{V}, \mathbf{W}]\!]$$

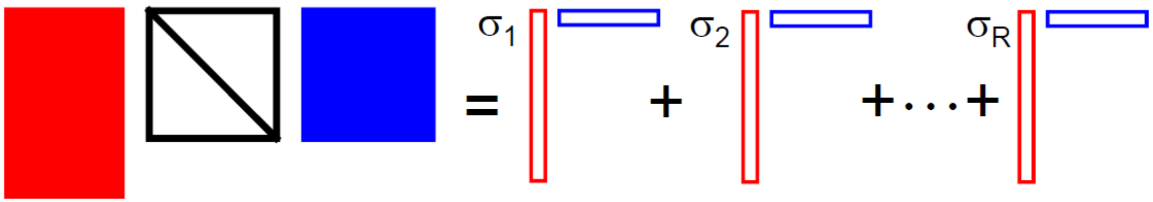
} Our Notation



What Is The HO Analogue of The Matrix SVD?

Matrix SVD:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T =$$



Tucker Tensor (finding bases for each subspace):

$$\mathbf{X} = \mathbf{\Sigma} \times_1 \mathbf{U} \times_2 \mathbf{V} = [\![\mathbf{\Sigma}\,; \mathbf{U}, \mathbf{V}]\!]$$

Kruskal Tensor (sum of rank-1 components):

$$\mathbf{X} = \sum_{r=1}^{R} \sigma_r \, \mathbf{u}_r \circ \mathbf{v}_r = [\![\sigma\,; \mathbf{U}, \mathbf{V}]\!]$$

SVD and 2D-SVD

SVD $X = (x_1, x_2, \cdots, x_n)$

Eigenvectors of XX^T and $X^T X$

$$X = U \Sigma V^T \qquad \Sigma = U^T X V$$

2D-SVD $\{A\} = \{A_1, A_2, \cdots, A_n\}$

Eigenvectors of

$$F = \sum_i (A_i - \bar{A})(A_i - \bar{A})^T$$ row-row covariance

$$G = \sum_i (A_i - \bar{A})^T (A_i - \bar{A})$$ column-column cov

$$A_i = U M_i V^T \qquad M_i = U^T A_i V$$

2D-SVD

$$\{A\} = \{A_1, A_2, \cdots, A_n\} \qquad \text{assume} \quad \overline{A} = 0$$

row-row cov: $F = \sum_i A_i A_i^T = \sum \lambda_k u_k u_k^T$

col-col cov: $G = \sum_i A_i^T A_i = \sum_{k=1} \zeta_k u_k u_k^T$

Bilinear $U = (u_1, u_2, \cdots, u_k)$

subspace $V = (v_1, v_2, \cdots, v_k)$ $\qquad M_i = U^T A_i V$

$$A_i = U M_i V^T, i = 1, \cdots, n$$

$$A_i \in \Re^{r\times c}, U \in \Re^{r\times k}, V \in \Re^{c\times k}, M_i \in \Re^{k\times k}$$