Linear Systems, SVD, PCA, Multilinear Models

COMPSCI 369

Georgy Gimel'farb

COMPSCI 369 Computational Science

Recommended reading:

- G. Strang, *Computational Science and Engineering*. Wellesley-Cambridge Press, 2007: Sections 1.5-7
- C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006: Sections 12.1-3
- W. H. Press et al., *Numerical Recipes: The Art of Scientific Computing*. Cambridge Univ. Press, 2007: Sec. 2.6; Chapter 11

Optional reading:

- M. Turk and A. Pentland, "Face recognition using eigenfaces", in *Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition (CVPR'91), Lahaiana, Maui, Hawaii, June 3-6, 1991*. Los Alamitos: IEEE Computer Society Press, pp. 586–591, 1991.
- M. Turk and A. Pentland, "Eigenfaces for recognition", Journal of Cognitive Neuroscience, Vol. 3 (1), pp. 71–86, 1991.

Acknowledgements: Face images are from the MIT-CBCL face recognition database:

- B. Weyrauch, J. Huang, B. Heisele, and V. Blanz: "Component-based face recognition with 3d morphable models", in *Proc. of CVPR Workshop on Face Processing in Video (FPIV'04), Washington DC, 2004*.

Learning Outcomes

Understand methods for solving **systems of linear equations**

- Understand standard matrix decompositions
  - $\Rightarrow$ Singular Value Decomposition (SVD) of an arbitrary rectangular $m \times n$ matrix

$$\mathbf{A} \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix} = \mathbf{UDV}^{\mathsf{T}}$$

  - $\Rightarrow$ QR and LU decompositions of square $n \times n$ matrices
- Be familiar with eigen-vectors $\mathbf{e}$ and eigen-values λ of an arbitrary square $n \times n$ matrix $\mathbf{A}$: i.e. $\mathbf{Ae}_i = \lambda_i \mathbf{e}_i$; $i = 1, \dots, n$
- Be familiar with Principal Component Analysis (PCA)

[*Optional*] Know about tensors and multilinear models

Solving Linear Equation Systems: Recall What You Know

Most common problem in numerical modelling and analysis:

$$\text{System of } m \text{ equations} \begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ \ldots & \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{cases}$$

$$\Leftrightarrow \underbrace{\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}}_{\mathbf{b}} \Leftrightarrow \mathbf{Ax} = \mathbf{b}$$

Given the known matrix $\mathbf{A}$ of coefficients and the known vector $\mathbf{b}$, find the unknown n-component vector $\mathbf{x}$

Solving Linear Equation Systems $\mathbf{Ax} = \mathbf{b}$

If the matrix $\mathbf{A}$ is square, $m = n$, and non-singular, $\det A \neq 0$, then in principle it can be inverted and $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$

- In practice, mostly $m \neq n$ and are of very large orders
  - E.g., m or $n \geq 10,000 \ldots 1,000,000$
- Even if $m = n$, testing for singularity and inverting a very large matrix present huge computational difficulties
- **Easily solvable systems** (diagonal, orthonormal, and triangilar):
  - Diagonal $n \times n$ matrices with nonzero components:

$$\mathbf{A} = \begin{bmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{bmatrix} \Rightarrow \mathbf{A}^{-1} = \begin{bmatrix} \frac{1}{a_1} & & \\ & \ddots & \\ & & \frac{1}{a_n} \end{bmatrix}$$

$$\Rightarrow x_i = \frac{b_i}{a_i} \text{ for all } i = 1, \ldots, n$$

    - Simplifying notation: $\mathbf{A} = \mathrm{diag}\{a_1, \ldots, a_n\}$
  - Orthonormal (or orthogonal) and triangular $n \times n$ matrices

Easily Solvable System: Orthonormal Matrix

$$\begin{aligned} 0.48x_1 + 0.64x_2 + 0.60x_3 &= 3.56 \\ 0.36x_1 + 0.48x_2 - 0.80x_3 &= -1.08 \\ 0.80x_1 - 0.60x_2 &= -0.40 \end{aligned}$$

or

$$x_1 \overbrace{\begin{bmatrix} 0.48 \\ 0.36 \\ 0.80 \end{bmatrix}}^{\mathbf{a}_1} + x_2 \overbrace{\begin{bmatrix} 0.64 \\ 0.48 \\ -0.60 \end{bmatrix}}^{\mathbf{a}_2} + x_3 \overbrace{\begin{bmatrix} 0.60 \\ -0.80 \\ 0.00 \end{bmatrix}}^{\mathbf{a}_3} = \begin{bmatrix} 3.56 \\ -1.08 \\ -0.40 \end{bmatrix}$$

Solution ($\mathbf{A}^{-1} = \mathbf{A}^\mathsf{T}$, as $\mathbf{a}_i^\mathsf{T}\mathbf{a}_j = 1$ if $i = j$ and 0 otherwise; $i, j \in \{1, 2, 3\}$):

$$\mathbf{x} = \mathbf{A}^\mathsf{T}\mathbf{b} \Rightarrow \begin{cases} x_1 &= 0.48 \cdot 3.56 - 0.36 \cdot 1.08 - 0.80 \cdot 0.40 \\ &\equiv 1.7088 - 0.3888 - 0.3200 \equiv 1.0 \\ x_2 &= 0.64 \cdot 3.56 - 0.48 \cdot 1.08 + 0.60 \cdot 0.40 \\ &\equiv 2.2784 - 0.5184 + 0.2400 \equiv 2.0 \\ x_3 &= 0.60 \cdot 3.56 + 0.80 \cdot 1.08 \\ &\equiv 2.136 + 0.864 \equiv 3.0 \end{cases}$$

Orthonormal Matrices

$\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \ldots\ \mathbf{a}_n]$ where $\mathbf{a}_i = [a_{i1}\ a_{i2}\ \ldots\ a_{in}]^\mathsf{T}$ are mutually orthogonal unit column vectors:

$$\mathbf{a}_i^\mathsf{T}\mathbf{a}_j \equiv \sum_{k=1}^{n} a_{ik}a_{jk} = \delta_{i-j} \equiv \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases}$$

Inversion by transposition: $\mathbf{A}^{-1} = \mathbf{A}^\mathsf{T}$, so $\mathbf{x} = \mathbf{A}^\mathsf{T}\mathbf{b}$

$$\mathbf{A}^\mathsf{T}\mathbf{A} \equiv \begin{bmatrix} \mathbf{a}_1^\mathsf{T} \\ \vdots \\ \mathbf{a}_n^\mathsf{T} \end{bmatrix} [\mathbf{a}_1\ \mathbf{a}_2\ \ldots\ \mathbf{a}_n] = \mathbf{I}_n \equiv \mathrm{diag}\{1, 1, \ldots, 1\}$$

Additional matrix property: $\mathbf{A}\mathbf{A}^\mathsf{T} = \mathbf{I}_n$, too

- Proof: $\mathbf{A}\mathbf{A}^\mathsf{T}\mathbf{A} \equiv \mathbf{A}\underbrace{(\mathbf{A}^\mathsf{T}\mathbf{A})}_{\mathbf{I}_n} = \mathbf{A}$ and $\mathbf{A}\mathbf{A}^\mathsf{T}\mathbf{A} \equiv (\mathbf{A}\mathbf{A}^\mathsf{T})\mathbf{A}$

Orthonormal Matrix: An Example

$$\mathbf{A} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \end{bmatrix}; \quad \mathbf{A}^\mathsf{T} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \end{bmatrix}$$

$$\mathbf{A}^\mathsf{T}\mathbf{A} \equiv \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \end{bmatrix} = \mathrm{diag}(1,1,1)$$

$$\mathbf{A}\mathbf{A}^\mathsf{T} \equiv \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \end{bmatrix} = \mathrm{diag}(1,1,1)$$

$$\mathbf{A}\mathbf{x} = \mathbf{b} \left(\text{i.e. } b_i = \sum_{j=1}^{3} a_{ij} x_j \right) \Rightarrow x_j = \sum_{i=1}^{3} a_{ij}^\mathsf{T} b_i = \sum_{i=1}^{3} a_{ji} b_i$$

Triangular Matrices

Lower and upper triangular matrices:

$$\mathbf{A} = \begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \text{ or } \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ & a_{22} & \dots & a_{2n} \\ & & \ddots & \vdots \\ & & & a_{nn} \end{bmatrix}$$

Simple sequential solution of the system $\mathbf{Ax} = \mathbf{b}$ without explicit inversion of $\mathbf{A}$:

- For the lower triangular matrix $x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j \right)$:

$$x_1 = \frac{b_1}{a_{11}};\ x_2 = \frac{b_2 - a_{21}x_1}{a_{22}};\ \dots;\ x_n = \frac{b_n - a_{n1}x_1 - \dots - a_{n-1,n}x_{n-1}}{a_{nn}}$$

Three Essential Factorisations

- **Elimination** (LU decomposition): $\mathbf{A} = \mathbf{LU}$
  - Lower triangular matrix  $\times$  Upper triangular matrix
- **Orthogonalisation** (QR decomposition): $\mathbf{A} = \mathbf{QR}$
  - Orthogonal matrix (columns)  $\times$ 
- **Singular Value Decomposition** (SVD): $\mathbf{A} = \mathbf{UDV}^{\mathsf{T}}$
  -  $\times$  diag(singular values) $\times$  Orthogonal (rows)
  - Orthonormal columns in $\mathbf{U}$ and $\mathbf{V}$: the left and right **singular vectors**, respectively
  - Left singular vector: an **eigenvector** of the square $m \times m$ matrix $\mathbf{AA}^{\mathsf{T}}$
  - Right singular vector: an **eigenvector** of the square $n \times n$ matrix $\mathbf{A}^{\mathsf{T}}\mathbf{A}$

Eigenvectors and Eigenvalues

Definition of an eigenvector $\mathbf{e}$ with an eigenvalue λ

$$\mathbf{A}\mathbf{e} = \lambda\mathbf{e}, \text{ i.e. } (\mathbf{A} - \lambda\mathbf{I})\mathbf{e} = \mathbf{0}$$

- λ is an eigenvalue of $\mathbf{A}$ if determinant $|\mathbf{A} - \lambda\mathbf{I}| = 0$
- This determinant is a polynomial in λ of degree n: so it has n roots $\lambda_1, \lambda_2, \ldots, \lambda_n$
- Every symmetric matrix $\mathbf{A}$ has a full set (basis) of n orthogonal unit eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$

Example of deriving eigenvalues and eigenvectors

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \Rightarrow |\mathbf{A} - \lambda\mathbf{I}| \equiv \begin{vmatrix} 2-\lambda & -1 \\ -1 & 2-\lambda \end{vmatrix} = \lambda^2 - 4\lambda + 3 = 0$$

$$\Rightarrow \lambda_1 = 1; \lambda_2 = 3 \Rightarrow \mathbf{e}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}; \mathbf{e}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Eigenvectors and Eigenvalues: A Few Properties

- No algebraic formula for the polynomial roots for $n > 4$ (Galois' Theorem)
  - Thus, the eigenvalue problem needs own special algorithms
  - Solving the eigenvalue problem is harder than to solve $\mathbf{Ax} = \mathbf{b}$
- Determinant $|\mathbf{A}| = \lambda_1 \lambda_2 \cdots \lambda_n$ (the product of eigenvalues)
- $\text{trace}(\mathbf{A}) = a_{11} + a_{22} + \ldots + a_{nn} = \lambda_1 + \lambda_2 + \ldots + \lambda_n$ (the sum of eigenvalues)
- $\mathbf{A}^k = \underbrace{\mathbf{A} \cdots \mathbf{A}}_{k \text{ times}}$ has the same eigenvectors as $\mathbf{A}$: e.g. for $\mathbf{A}^2$

$$\mathbf{Ae} = \lambda \mathbf{e} \quad \Rightarrow \quad \mathbf{AAe} = \lambda \mathbf{Ae} = \lambda^2 \mathbf{e}$$

  - Eigenvalues of $\mathbf{A}^k$ are $\lambda_1^k, \ldots, \lambda_n^k$
  - Eigenvalues of $\mathbf{A}^{-1}$ are $\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_n}$

Properties of Eigenvectors and Eigenvalues: An Example

2×2 matrix $\mathbf{A} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$:

- Eigenvectors and eigenvalues:
  $\mathbf{e}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$; $\lambda_1 = 1$, and $\mathbf{e}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$; $\lambda_2 = 3$
- Determinant $\det \mathbf{A} \equiv |\mathbf{A}| = 4 - 1 = 3 \Longleftrightarrow \lambda_1 \cdot \lambda_2 \equiv 1 \cdot 3 = 3$
- $\mathrm{trace}(\mathbf{A}) = 2 + 2 = 4 \Longleftrightarrow \lambda_1 + \lambda_2 \equiv 1 + 3 = 4$
- Inverse matrix $\mathbf{A}^{-1} = \frac{1}{3} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$: eigenvalues $\lambda_1 = \frac{1}{3}$ and $\lambda_2 = 1$
- Matrix $\mathbf{A}^2 = \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix}$: eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 9$
- Matrix $\mathbf{A}^3 = \begin{bmatrix} 14 & -13 \\ -13 & 14 \end{bmatrix}$: eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 27$

Eigenvectors and Eigenvalues: Matrix Diagonalisation

Diagonalisation of a square $n \times n$ matrix $\mathbf{A}$ with n linearly independent eigenvectors $\mathbf{e}$

- Eigenvector matrix $\mathbf{S} = [\mathbf{e}_1 \, \mathbf{e}_2 \, \ldots \, \mathbf{e}_n] \Rightarrow$
  Diagonalisation $\mathbf{S}^{-1}\mathbf{A}\mathbf{S} = \mathbf{\Lambda} \equiv \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$
- With no repeated eigenvalues, $\mathbf{A}$ always has n linearly independent eigenvectors
- **Real symmetric matrices**:
  real eigenvalues and orthonormal eigenvectors, so $\mathbf{S}^{-1} = \mathbf{S}^{\mathsf{T}}$
- **Symmetric diagonalisation**:

$$\mathbf{A} \;\Rightarrow\; \mathbf{\Lambda} = \mathbf{S}^{-1}\mathbf{A}\mathbf{S} = \mathbf{S}^{\mathsf{T}}\mathbf{A}\mathbf{S} \;\Rightarrow\; \mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{\mathsf{T}}$$

- Yet another useful representation of $\mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{\mathsf{T}}$:

$$\mathbf{A} = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^{\mathsf{T}} + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^{\mathsf{T}} + \ldots + \lambda_n \mathbf{e}_n \mathbf{e}_n^{\mathsf{T}}$$

Matrix Diagonalisation: An Example

2×2 matrix $\mathbf{A} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$: Eigenvector matrix $\mathbf{S} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$

- Diagonalisation:

$$\underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}}_{\mathbf{S}^\mathsf{T}} \underbrace{\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}}_{\mathbf{S}} =$$

$$\underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} & -\frac{3}{\sqrt{2}} \end{bmatrix}}_{\mathbf{S}^\mathsf{T}\mathbf{A}} \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}}_{\mathbf{S}} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}}_{\mathbf{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2)}$$

- Useful representation

$$\mathbf{A} = \underbrace{1}_{\lambda_1} \cdot \underbrace{\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}}_{\mathbf{e}_1\mathbf{e}_1^\mathsf{T}} + \underbrace{3}_{\lambda_2} \cdot \underbrace{\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}}_{\mathbf{e}_2\mathbf{e}_2^\mathsf{T}} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

Singular Value Decomposition

- Factorisation of any ordinary (real or complex) rectangular $m \times n$ matrix (most frequently when $m \geq n$)
- Applications in signal processing, pattern recognition, and statistics
  - Least squares data fitting, regularised inverse problems, pseudoinverse, principal component analysis (PCA), etc.
    - Computational tomography, seismology, weather forecast, image compression, image denoising, etc.
  - If sets of linear equations are singular or very close to singular then conventional solutions (e.g. by LU decomposition) give unsatisfactory results
  - SVD diagnoses and in some cases solves the problem giving an useful numerical answer (though not necessarily the **expected** one!)
- SVD is a method of choice for solving most of the linear systems that appear in the least squares problems

Singular Value Decomposition

SVD represents an ordinary $m \times n$ matrix $\mathbf{A}$ as $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\mathsf{T}$

- $\mathbf{U}$: an $m \times n$ column-orthogonal matrix; its n columns – the top n eigenvectors $\mathbf{u}$ of the $m \times m$ matrix $\mathbf{A}\mathbf{A}^\mathsf{T}$
- $\mathbf{V}$: an $n \times n$ orthogonal matrix; its n columns – the eigenvectors $\mathbf{v}$ of the $n \times n$ matrix $\mathbf{A}^\mathsf{T}\mathbf{A}$
- $\mathbf{D}$: an $n \times n$ diagonal matrix of non-negative (≥ 0) singular values $\sigma_1, \ldots, \sigma_n$ such that $\mathbf{A}\mathbf{v}_j = \sigma_j \mathbf{u}_j$; $j = 1, \ldots, r$
  - $\sigma_j = \sqrt{\lambda_j}$ where λ_j is the eigenvalue for $\mathbf{v}_j$

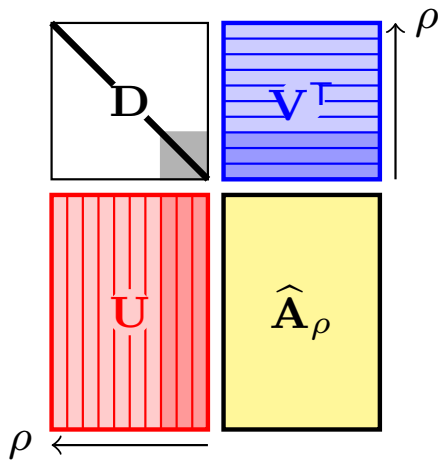
Useful representation (r – the rank of $\mathbf{A}$, i.e the number of its nonzero singular values):

$$\mathbf{A} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^\mathsf{T} + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^\mathsf{T} + \ldots + \sigma_r \mathbf{u}_r \mathbf{v}_r^\mathsf{T};$$

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$$

Approximate (truncated) representation:

$$\widehat{\mathbf{A}}_\rho = \sum_{i=1}^{\rho} \sigma_i \mathbf{u}_i \mathbf{v}_i^\mathsf{T};\ \rho < r$$



SVD: an Example

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow \mathbf{A}\mathbf{A}^{\mathsf{T}} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\Rightarrow \underbrace{\lambda_1 = 3; \mathbf{u}_1 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}; \quad \lambda_2 = 1; \mathbf{u}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}}$$

Top n=2 eigenvalues and eigenvectors for $\mathbf{A}\mathbf{A}^{\mathsf{T}}$

$$\mathbf{A}^{\mathsf{T}}\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\Rightarrow \underbrace{\lambda_1 = 3; \mathbf{v}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \quad \lambda_2 = 1; \mathbf{v}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}}$$

Eigenvalues and eigenvectors for $\mathbf{A}^{\mathsf{T}}\mathbf{A}$

SVD: an Example (cont.)

Singular values: $\mathbf{A}\mathbf{v}_j = \sigma_j \mathbf{u}_j; j = 1, 2$

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \sigma_1 \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \Rightarrow \sigma_1 = \sqrt{3}$$

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \sigma_2 \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \Rightarrow \sigma_2 = 1$$

$$\mathbf{A} \equiv \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} = \underbrace{[\mathbf{u}_1\ \mathbf{u}_2]}_{\mathbf{U}} \underbrace{\mathrm{diag}(\sigma_1, \sigma_2)}_{\mathbf{D}} \underbrace{\begin{bmatrix} \mathbf{v}_1^\mathsf{T} \\ \mathbf{v}_2^\mathsf{T} \end{bmatrix}}_{\mathbf{V}^\mathsf{T}}$$

$$\equiv \underbrace{\begin{bmatrix} 1/\sqrt{6} & 1/\sqrt{2} \\ 2/\sqrt{6} & 0 \\ 1/\sqrt{6} & -1/\sqrt{2} \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}}_{\mathbf{V}^\mathsf{T}}$$

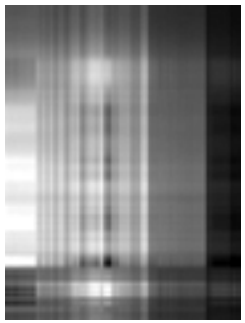
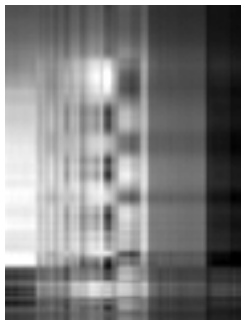
SVD: an Example (cont.)

Matrix representation:

$$\begin{aligned}
\widehat{\mathbf{A}}_1 &= \sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top \\
&\equiv \sqrt{3} \cdot \tfrac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix} \equiv \tfrac{1}{2} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \equiv \begin{bmatrix} 0.5 & 0.5 \\ 1 & 1 \\ 0.5 & 0.5 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
\widehat{\mathbf{A}}_2 &= \widehat{\mathbf{A}}_1 + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^\top \\
&\equiv \begin{bmatrix} 0.5 & 0.5 \\ 1 & 1 \\ 0.5 & 0.5 \end{bmatrix} + 1 \cdot \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \tfrac{1}{\sqrt{2}} \begin{bmatrix} -1 & 1 \end{bmatrix} \\
&\equiv \begin{bmatrix} 0.5 & 0.5 \\ 1 & 1 \\ 0.5 & 0.5 \end{bmatrix} + \begin{bmatrix} -0.5 & 0.5 \\ 0 & 0 \\ 0.5 & -0.5 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}}_{\mathbf{A}}
\end{aligned}$$

Image Compression with SVD

					
$\mathbf{A}_{230\times322}$	$37,948\mathbf{u}_1\mathbf{v}_1^\mathsf{T}$	$+5,050\mathbf{u}_2\mathbf{v}_2^\mathsf{T}$	$+4,331\mathbf{u}_3\mathbf{v}_3^\mathsf{T}$	$+3,428\mathbf{u}_4\mathbf{v}_4^\mathsf{T}$	$+2.916\mathbf{u}_5\mathbf{v}_5^\mathsf{T}$
$u = 74.060$	$c_1 = 553$	$c_2 = 1,106$	$c_3 = 1,659$	$c_4 = 2,212$	$c_5 = 2,765$
ε_ρ:	$\varepsilon_1 = 27.3$	$\varepsilon_2 = 24.3$	$\varepsilon_3 = 21.6$	$\varepsilon_4 = 20.1$	$\varepsilon_5 = 18.8$
					
$+3,045\mathbf{u}_6\mathbf{v}_6^\mathsf{T}$	$+2,718\mathbf{u}_7\mathbf{v}_7^\mathsf{T}$	$+2,532\mathbf{u}_8\mathbf{v}_8^\mathsf{T}$	$+2,417\mathbf{u}_9\mathbf{v}_9^\mathsf{T}$	$+1,975\mathbf{u}_{10}\mathbf{v}_{10}^\mathsf{T}$	$\ldots + 947\mathbf{u}_{20}\mathbf{v}_{20}^\mathsf{T}$
$c_6 = 3,318$	$c_7 = 3,871$	$\varepsilon_8 = 4,424$	$c_9 = 4,977$	$c_{10} = 5,530$	$c_{20} = 11,060$
$\varepsilon_6 = 15.9$	$\varepsilon_7 = 14.7$	$\varepsilon_8 = 13.7$	$\varepsilon_9 = 13.0$	$\varepsilon_{10} = 12.4$	$\varepsilon_{20} = 7.9$

- $u \equiv mn$ – number of pixels; c_ρ – compressed data volume ($c_\rho = \rho(1 + n + m)$)
- $\varepsilon_\rho = \frac{1}{mn}\sum_{i,j=1,1}^{m.n} |A_{ij} - \widehat{A}_{\rho:ij}|$ – mean absolute reconstruction error per pixel

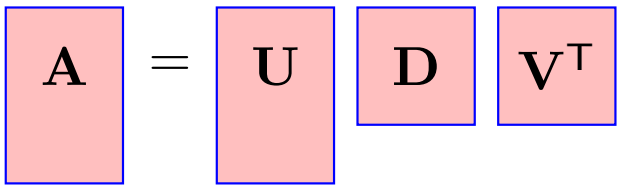
Image Compression with SVD

					
σ_ρ:	$\sigma_{30} = 552$	$\sigma_{40} = 450$	$\sigma_{50} = 360$	$\sigma_{60} = 296$	$\sigma_{70} = 260$
$u = 74.060$	$c_{30} = 16,590$	$c_{40} = 22,120$	$c_{50} = 27,650$	$c_{60} = 33,180$	$c_{70} = 38,710$
ε_ρ:	$\varepsilon_{30} = 6.3$	$\varepsilon_{40} = 4.9$	$\varepsilon_{50} = 4.3$	$\varepsilon_{60} = 4.1$	$\varepsilon_{70} = 3.5$
					
$\sigma_{80} = 226$	$\sigma_{90} = 196$	$\sigma_{100} = 177$	$\sigma_{110} = 155$	$\sigma_{120} = 141$	$\sigma_{130} = 122$
$c_{80} = 44,240$	$c_{90} = 49,770$	$c_{100} = 55,300$	$c_{110} = 60,830$	$c_{120} = 66,360$	$c_{130} = 71,890$
$\varepsilon_{80} = 3.1$	$\varepsilon_{90} = 2.8$	$\varepsilon_{100} = 2.5$	$\varepsilon_{110} = 2.2$	$\varepsilon_{120} = 1.9$	$\varepsilon_{130} = 1.6$

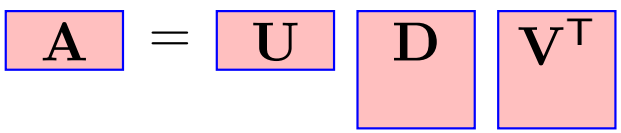
- $u \equiv mn$ – number of pixels; c_ρ – compressed data volume ($c_\rho = \rho(1 + n + m)$)
- $\varepsilon_\rho = \frac{1}{mn} \sum_{i,j=1,1}^{m.n} |A_{ij} - \widehat{A}_{\rho:ij}|$ – mean absolute reconstruction error per pixel

Structure of SVD

- Overdetermined case: $m > n$ (more equations than unknowns)



- Underdetermined case: $m < n$ (fewer equations than unknowns)

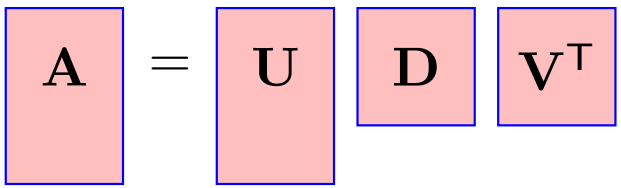


- Matrix $\mathbf{V}$ is orthogonal (orthonormal): $\mathbf{V}\mathbf{V}^\mathsf{T} = \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}_n$ (the unit $n \times n$ matrix)
- Matrix $\mathbf{U}$ is column-orthogonal (orthonormal):

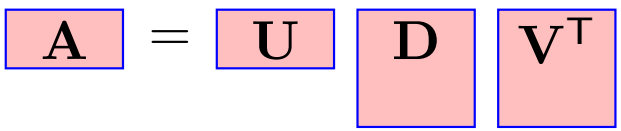
$$\mathbf{U}^\mathsf{T}\mathbf{U} = \left[\sum_{i=1}^{m} u_{i\alpha}u_{i\beta} = \delta_{\alpha\beta}\right];\ 1 \le \alpha, \beta \le n$$

Structure of SVD

- Overdetermined case: $m > n$ (more equations than unknowns)



- Underdetermined case: $m < n$ (fewer equations than unknowns)



  - If $m \geq n$, $\delta_{\alpha\beta} = 1$ if $\alpha = \beta$ and $\delta_{\alpha\beta} = 0$ otherwise:

$$\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}_n$$

  - If $m < n$, the singular values $\sigma_j = 0$ for $j = m+1, \ldots, n$ and the corresponding columns in $\mathbf{U}$ are also zero
    - So the above relationships for $\delta_{\alpha\beta}$ in $\mathbf{U}^\mathsf{T}\mathbf{U}$ hold only for $1 \leq \alpha, \beta \leq m$

Basic Properties of SVD

- Singular Value Decomposition (SVD) can always be done, no matter how singular the matrix is!
- SVD is almost "unique" up to:
  1. The same permutation of

     the columns of $\mathbf{U}$,

     the elements of $\mathbf{D}$, and

     the columns of $\mathbf{V}$ (or rows of $\mathbf{V}^\mathsf{T}$); or
  2. An orthogonal rotation on any set of columns of $\mathbf{U}$ and $\mathbf{V}$ whose corresponding elements of $\mathbf{D}$ are exactly equal

Basic Properties of SVD

Due to the permutation freedom:

- If $m < n$, the numerical SVD algorithm need not return zero σ_j's in their canonical positions $j = m + 1, \ldots, n$
- Actually, the $n - m$ zero singular values can be scattered among all n positions $j = 1, \ldots, n$
- Thus, all the singular values should be to sorted into a canonical order
- It is convenient to sort into descending order $\sigma_1 \geq \sigma_2 \geq \ldots$

Solutions of Systems of Linear Equations (optional: basic notions)

System $\mathbf{Ax} = \mathbf{b}$ is a linear mapping of an n-dimensional vector space $\mathcal{X}$ to (generally) an m-dimensional vector space $\mathcal{B}$

- The mapping may also reach only a lesser-dimensional subspace of the full m-dimensional one
  - The latter subspace is called **the range** of $\mathbf{A}$
- **Rank** r of $\mathbf{A}$:
  - $1 \leq r \leq \min(m, n)$
  - Rank r of $\mathbf{A}$ is the dimension r of the range of $\mathbf{A}$
  - Rank r is equal to the number of linearly independent columns (also – the number of linearly independent rows) of $\mathbf{A}$

Simple Examples

1. Overdetermined system – 2D-to-3D mapping; the matrix of rank 2:

$$\overbrace{\begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}}^{\mathbf{A}} \overbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}^{\mathbf{x}} = \overbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}^{\mathbf{b}}; \quad \mathbf{Ax} = \mathbf{0} \Rightarrow \text{0D point } \mathbf{x} = \mathbf{0}$$

2. Underdetermined system – 3D-to-2D mapping; the matrix of rank 2:

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \quad \mathbf{Ax} = \mathbf{0} \Rightarrow \text{1D line } x_1 = -x_2 = x_3$$

3. Underdetermined system – 3D-to-2D mapping; the matrix of rank 1

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \quad \mathbf{Ax} = \mathbf{0} \Rightarrow \text{2D plane } x_1+x_2+x_3 = 0$$

Solutions of Systems of Linear Equations (optional: basic notions)

Nullspace $\text{null}(\mathbf{A})$ of $\mathbf{A}$:

the space of the vectors $\mathbf{x}$ such that $\mathbf{Ax} = \mathbf{0}$

- **Nullity** – the dimension of the nullspace
- **Rank – nullity theorem**:

  $\text{rank}(\mathbf{A}) + \text{nullity } \mathbf{A} = n$ (the number of columns)

  - Example 1 (Slide 28): rank 2; nullity 0 (point in 2D); $n = 2$
  - Example 2 (Slide 28): rank 2; nullity 1 (line in 3D) ; $n = 3$
  - Example 3 (Slide 28): rank 1; nullity 2 (plane in 3D); $n = 3$

If $m = n$ and $r = n$: $\mathbf{A}$ is square, nonsingular and invertible

- $\mathbf{Ax} = \mathbf{b}$ has a unique solution for any $\mathbf{b}$, and only zero vector is mapped to zero

$\mathbf{Ax} = \mathbf{b}$: Why SVD?

Most favourable case: the matrix $\mathbf{A}$ is square and invertible:

- Only in this case the **LU decomposition** $\blacksquare = \blacksquare\blacksquare$ of $\mathbf{A}$ into the product $\mathbf{LU}$ of the lower and upper triangular matrices is the preferred solution method for $\mathbf{x}$

In a host of practical cases: $\mathbf{A}$ has rank $r < n$ (i.e. nullity > 0):

- Most right-hand side vectors $\mathbf{b}$ yield no solution: e.g.

$$\text{there is no such } \mathbf{x} \text{ that } \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

- But some $\mathbf{b}$ have multiple solutions (actually: the whole subspace of them): e.g. for all $\mathbf{x}$ such that $x_1 + x_2 + x_3 = 1$

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$\mathbf{Ax} = \mathbf{b}$: Why SVD?

Practical example: a linear 1D predictor

Time series of $m = 1000$ measurements $(s_i \ : \ i = 1, \ldots, 1000)$ – an unknown predictor $\widehat{s}_i = a_1 s_{i-1} + a_2 s_{i-2} + a_3 s_{i-3}$; $n = 3 \ll m$:

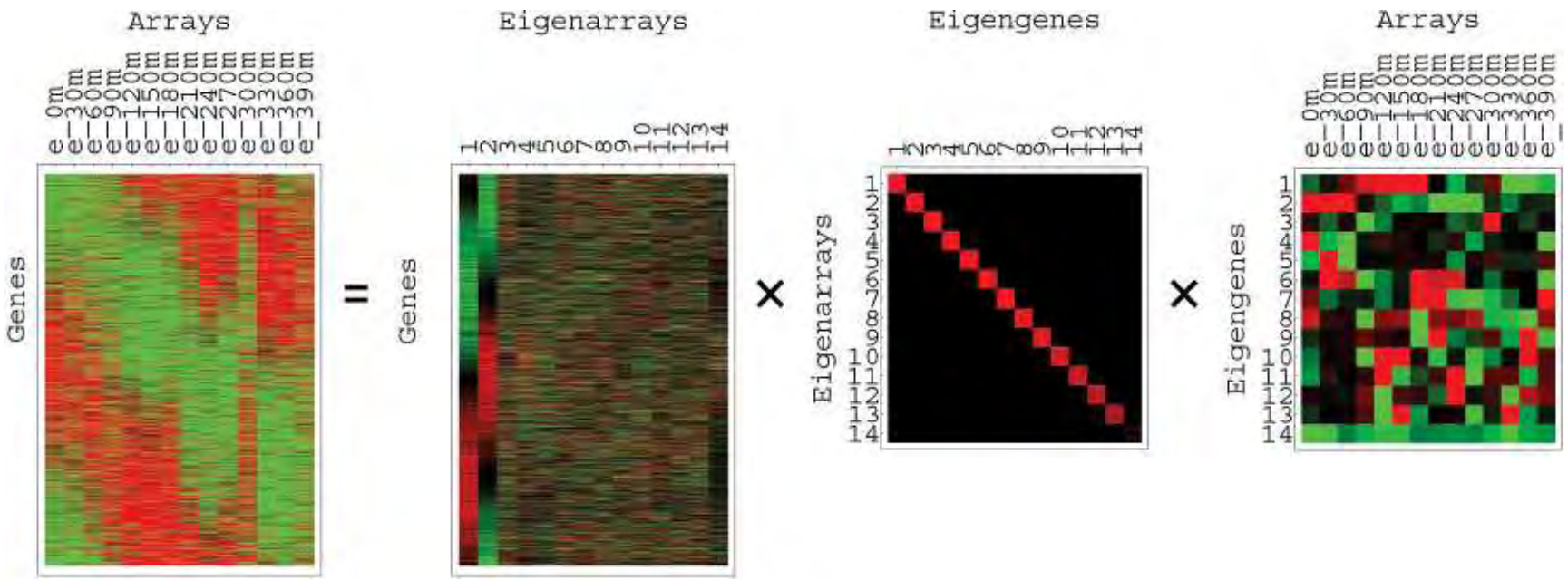
$$\begin{bmatrix} s_3 & s_2 & s_1 \\ s_4 & s_3 & s_2 \\ \vdots & \vdots & \vdots \\ s_{999} & s_{998} & s_{997} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} s_4 \\ s_5 \\ \vdots \\ s_{1000} \end{bmatrix}$$

SVD $\mathbf{A} = \mathbf{UDV}^\mathsf{T}$ explicitly constructs orthonormal bases for both the nullspace and the range of a matrix!

- Columns of $\mathbf{U}$ whose same-numbered singular values $\sigma_j \neq 0$ in $\mathbf{D}$: an orthonormal set of basis vectors that span the range
- Columns of $\mathbf{V}$ whose same-numbered singular values $\sigma_j = 0$ in $\mathbf{D}$: an orthonormal basis for the nullspace

Stanford University: SVD of Gene Expression Matrix

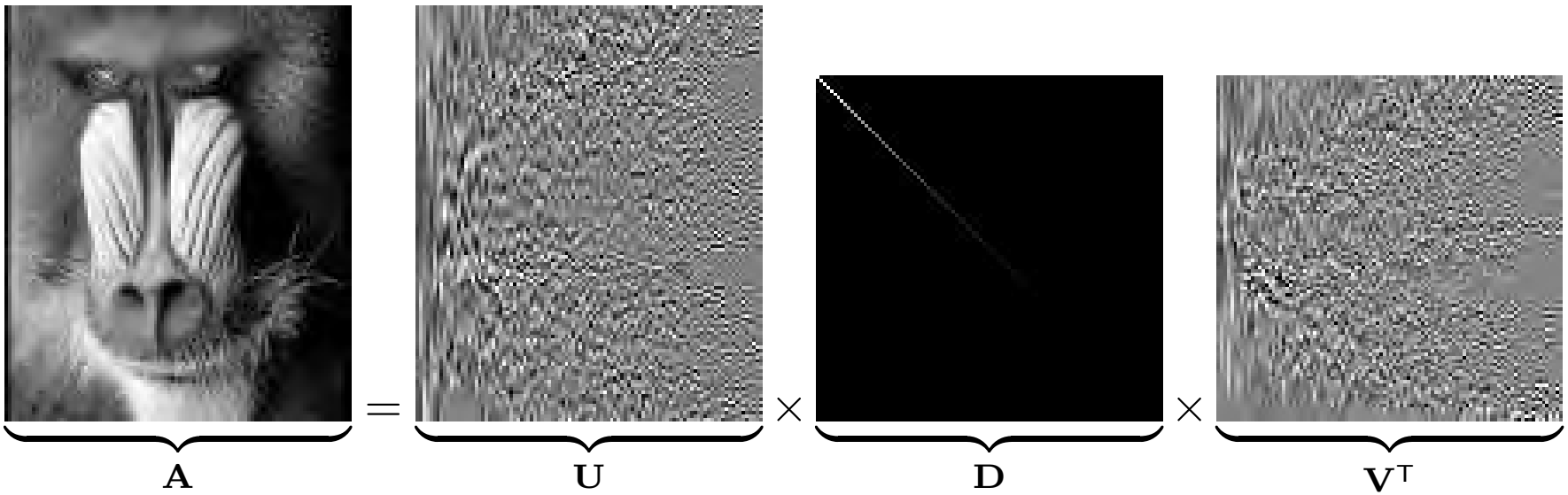
http://smd.stanford.edu/images/help/svd_matrices.gif



SVD of Yeast Cell Cycle Data

The $m \times 14$ matrix $\mathbf{A}$ of gene expression values (m genes; 14 arrays) is decomposed into three matrices $\mathbf{A} = \mathbf{UDV}^{\mathsf{T}}$: the $m \times 14$ eigenarrays matrix $\mathbf{U}$, the 14×14 eigenexpression levels matrix $\mathbf{D}$, and the 14×14 eigengenes matrix $\mathbf{V}^{\mathsf{T}}$

SVD of "Baboon"



The 120×100 matrix $\mathbf{A}$ of grey values in image points is decomposed into three matrices $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\mathsf{T}$: the 120×100 column-orthogonal matrix $\mathbf{U}$, the 100×100 diagonal matrix of singular values $\mathbf{D}$, and the 100×100 orthogonal matrix $\mathbf{V}^\mathsf{T}$

Ranges of the matrix elements:

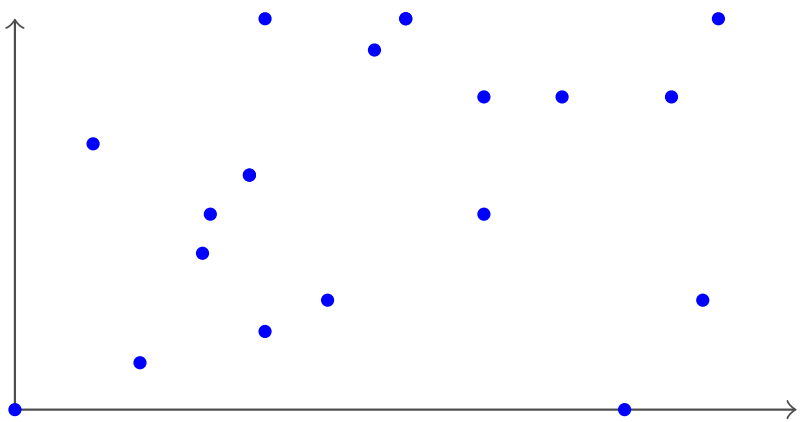
$\mathbf{A}$	$\mathbf{U}$	$\mathbf{D}$	$\mathbf{V}$
[0;255]	[-0.38;0.41]	[1.7; 12,326.7]	[-0.55; 0.59]

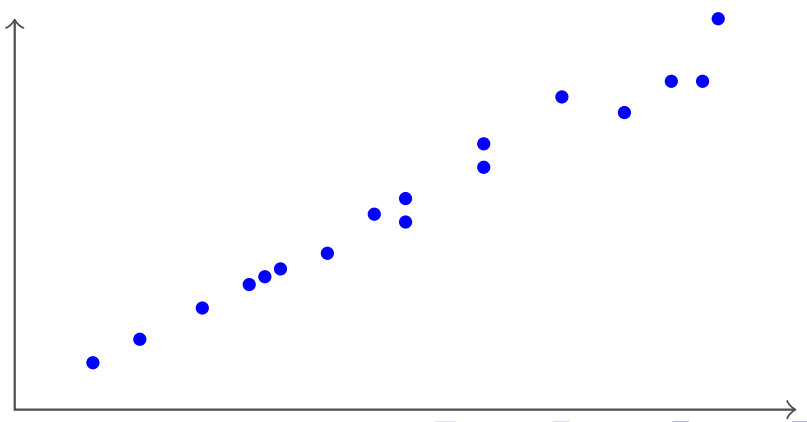
Principal Component Analysis (PCA)

Goal of PCA: to identify most important properties revealed by an $m \times n$ matrix $\mathbf{A}$ of measurement data:

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix} \equiv \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

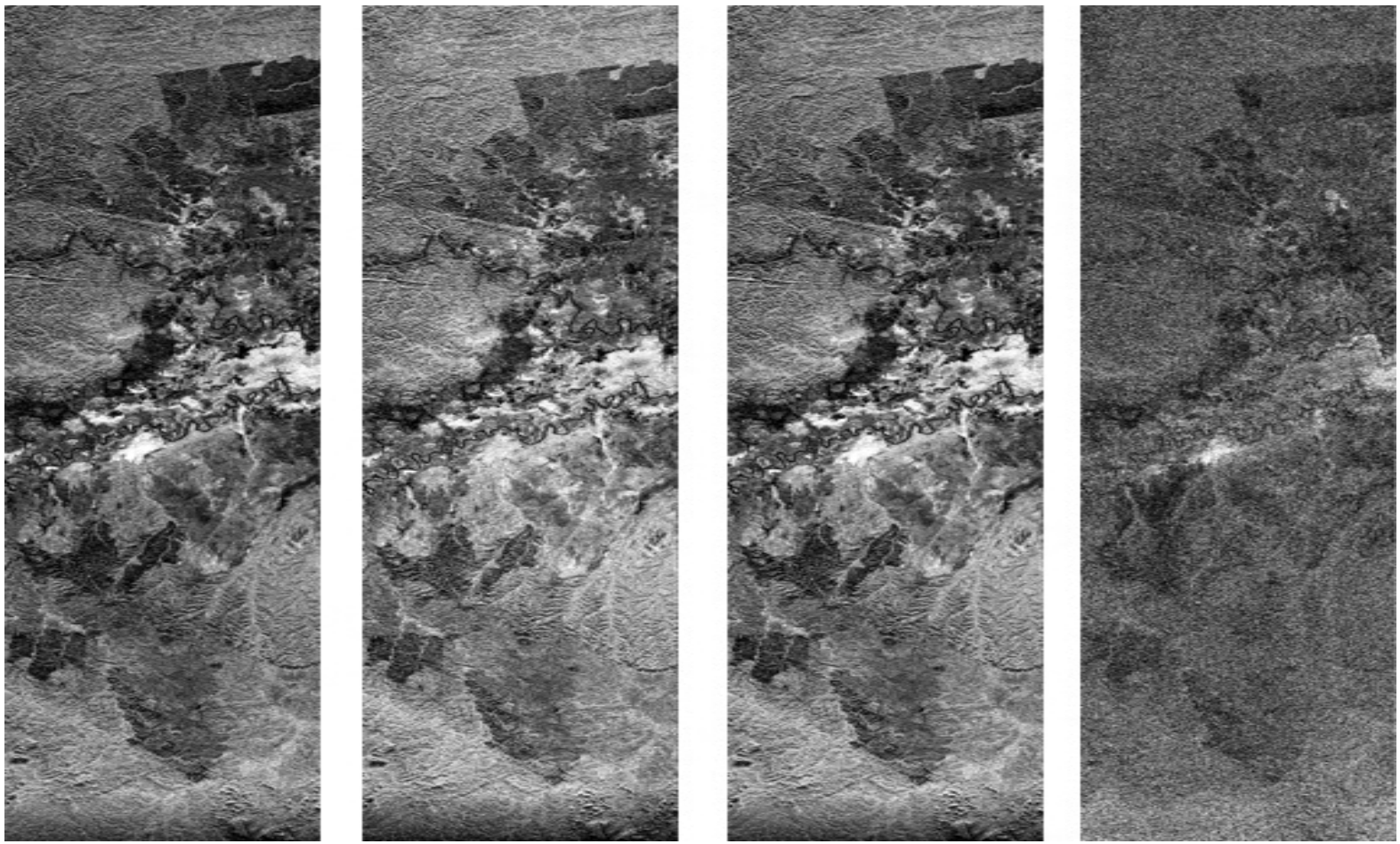
PCA is also called the **Karhunen-Loève transform**





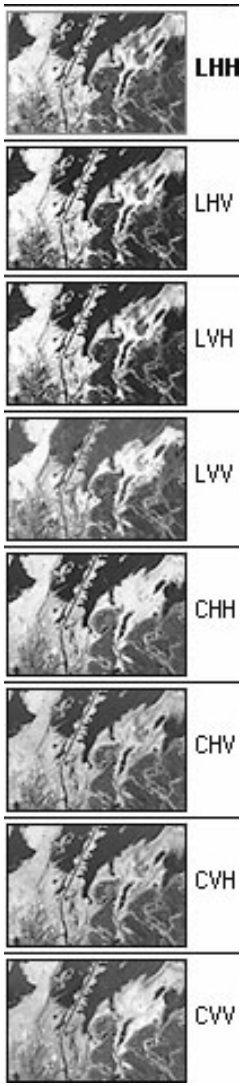
Practical Example 1

SAR image1 SAR Image 2 PCA comp. 1 PCA comp. 2

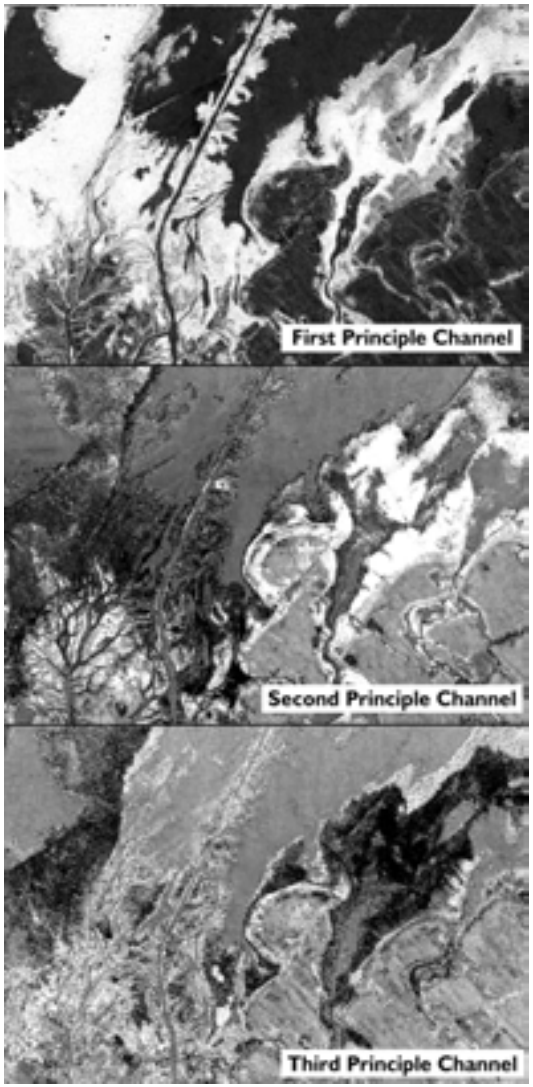


M. Shimada, M. Minamisawa, O. Isoguchi: A Study on Estimating the Forest Fire Scars in East Kalimantan Using JERS-1 SAR Data. http://www.eorc.jaxa.jp/INSAR-WS/meeting/paper/g2/g2.html

Practical Example 2: http://www.cacr.caltech.edu/SDA/images/

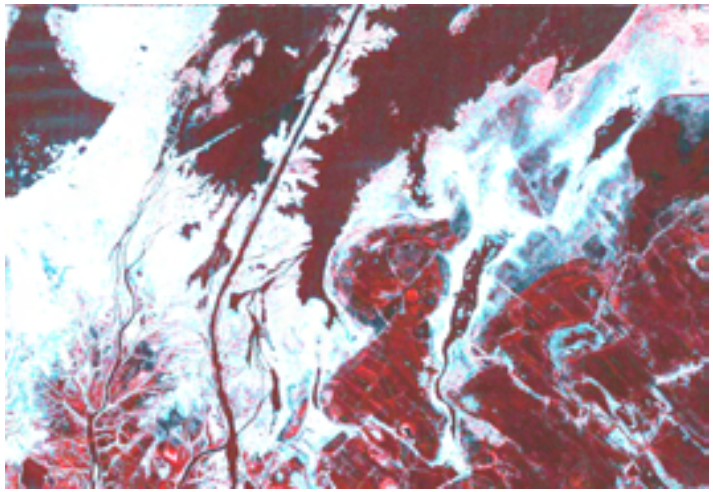


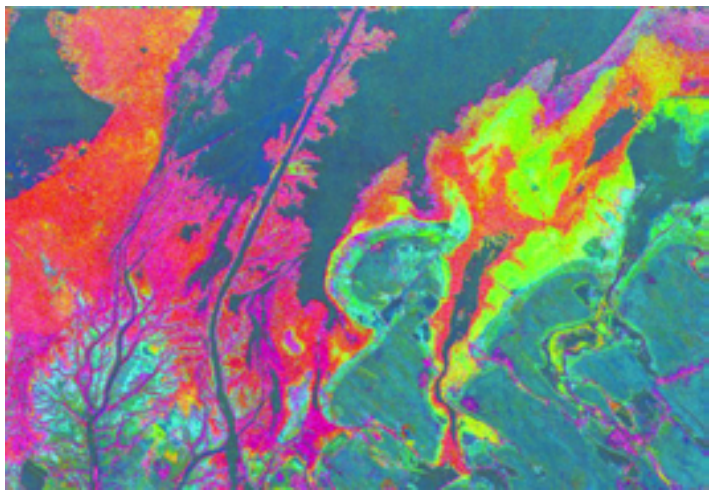
SAR channels



3 top-rank PC

Colour-coded 3 original SAR channels





Colour-coded 3 top-rank PC

Principal Component Analysis (PCA)

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix} \equiv \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Measurements are considered as linear combinations of the original properties (e.g. principal components):

$$\mathbf{a} = \alpha_1 \mathbf{u}_1 + \ldots + \alpha_k \mathbf{u}_k$$

- Weights α in the combination are called **loadings**
- All loadings are nonzero

General assumption: **zero mean value** for this data, so the **variance** is the critical indicator of importance, large or small

Data Projection by PCA

$m \times m$ **covariance matrix** $\mathbf{\Sigma}_n$ for n samples of m-dimensional data:

$$\mathbf{\Sigma}_n = \frac{1}{n-1}\left(\mathbf{a}_1\mathbf{a}_1^\mathsf{T} + \ldots + \mathbf{a}_n\mathbf{a}_n^\mathsf{T}\right) = \frac{1}{n-1}\mathbf{A}\mathbf{A}^\mathsf{T}$$

- Best basis in the property space $\mathcal{R}_m$: eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots$ of $\mathbf{A}\mathbf{A}^\mathsf{T}$
- Best basis in the sample space $\mathcal{R}_n$: eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots$ of $\mathbf{A}^\mathsf{T}\mathbf{A}$

Data Projection by PCA

PCA: the orthogonal projection of data onto a lower-dimensional ($k < m$) linear subspace (the principal subspace)

- Variance of the projected data in the principal subspace is maximal (with respect to any other subspace of the same dimension k)
- **Principal subspace**: the k eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ of $\mathbf{A}\mathbf{A}^\mathsf{T}$ (i.e. of $\mathbf{\Sigma}_n$, too) with the largest eigenvalues:

$$\begin{aligned} \mathbf{a}_{[k]} &= \alpha_1 \mathbf{u}_1 + \ldots + \alpha_k \mathbf{u}_k \\ \alpha_i &= \mathbf{u}_i^\mathsf{T} \mathbf{a} = u_{i,1} a_1 + \ldots + u_{i,m} a_m \\ &\equiv \mathbf{u}_i^\mathsf{T} \left(\alpha_1 \mathbf{u}_1 + \ldots + \alpha_m \mathbf{u}_m \right) \end{aligned}$$

Data Projection by PCA

- Column-orthogonal matrix of k principal components: $\mathbf{U}_k = [\mathbf{u}_1 \ \ldots \ \mathbf{u}_k]$
- Projection matrix (projection to the principal subspace):

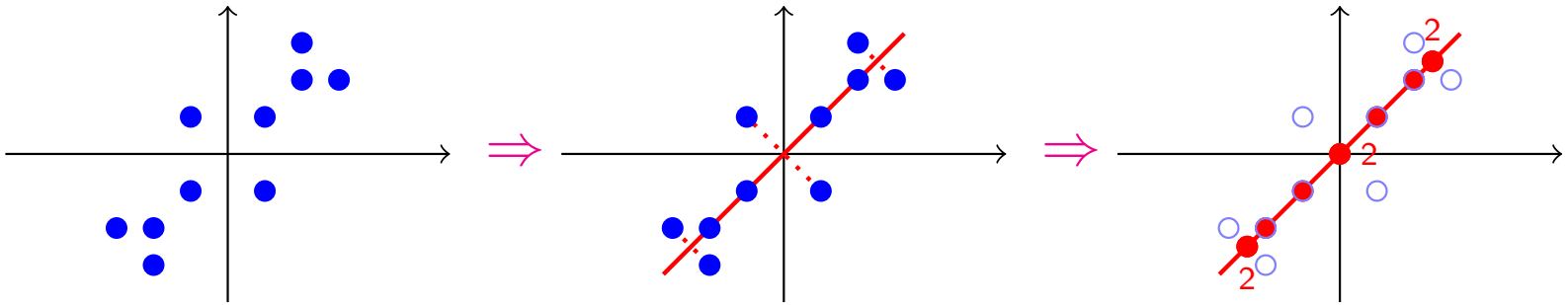
$$\mathbf{P}_k = \mathbf{U}_k \mathbf{U}_k^\mathsf{T} \equiv \left[\begin{array}{cccc} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_k \end{array} \right] \left[\begin{array}{c} \mathbf{u}_1^\mathsf{T} \\ \mathbf{u}_2^\mathsf{T} \\ \vdots \\ \mathbf{u}_k^\mathsf{T} \end{array} \right]$$

- Original measurement vector $\mathbf{a} \rightarrow$ Projected vector $\mathbf{P}_k \mathbf{a}$

Another interpretation of the PCA:

- Projection $\mathbf{P}_k$ to the principal subspace minimises the projection error $\sum_{j=1}^{n} \| \mathbf{a}_j - \mathbf{P}_k \mathbf{a}_j \|^2$

PCA: An Example

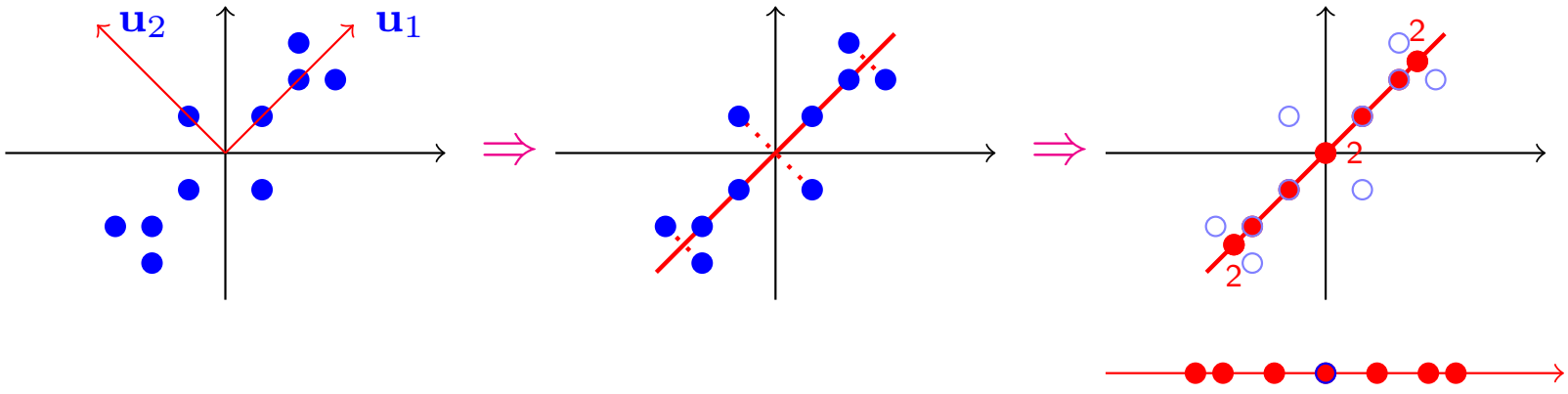


$$\mathbf{A} = [\mathbf{a}_1 \dots \mathbf{a}_{10}] = \begin{bmatrix} -3 & -2 & -2 & -1 & -1 & 1 & 1 & 2 & 2 & 3 \\ -2 & -3 & -2 & -1 & 1 & -1 & 1 & 2 & 3 & 2 \end{bmatrix}$$

Variances $(\sigma_{11}; \sigma_{22})$ and covariances $(\sigma_{12} \equiv \sigma_{21})$:

$$\begin{aligned} \sigma_{11} &= \frac{1}{9}\left(a_{1,1}^2 + \dots + a_{1,10}^2\right) \\ &= \frac{1}{9}\left((-3)^2 + (-2)^2 + (-2)^2 + \dots + 2^2 + 3^2\right) = \frac{38}{9} \\ \sigma_{22} &= \frac{1}{9}\left(a_{2,1}^2 + \dots + a_{2,10}^2\right) \\ &= \frac{1}{9}\left((-2)^2 + (-3)^2 + (-2)^2 + \dots + 3^2 + 2^2\right) = \frac{38}{9} \\ \sigma_{12} &= \frac{1}{9}\left(a_{1,1}a_{2,1} + \dots + a_{1,10}a_{2,10}\right) \\ &= \frac{1}{9}\left((-3)(-2) + (-2)(-3) + (-2)(-2) + \dots + 2 \cdot 3 + 3 \cdot 2\right) = \frac{32}{9} \end{aligned}$$

PCA: An Example (cont.)



Covariance matrix $\mathbf{\Sigma}_{10} = \frac{1}{9}\begin{bmatrix} 38 & 32 \\ 32 & 38 \end{bmatrix} \Rightarrow \begin{vmatrix} 38-\lambda & 32 \\ 32 & 38-\lambda \end{vmatrix} = 0$

$\Rightarrow (38-\lambda)^2 - 32^2 \equiv (70-\lambda)(6-\lambda) = 0 \Rightarrow$ Eigenvalues and vectors:

$$\lambda_1 = 70;\ \mathbf{u}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}; \quad \lambda_2 = 6;\ \mathbf{u}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$\Rightarrow$ Projection matrix $\mathbf{P}_1 = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \Rightarrow$

$$\mathbf{P}_1\mathbf{A} = \begin{bmatrix} -2.5 & -2.5 & -2 & -1 & 0 & 0 & 1 & 2 & 2.5 & 2.5 \\ -2.5 & -2.5 & -2 & -1 & 0 & 0 & 1 & 2 & 2.5 & 2.5 \end{bmatrix}$$

Linear Image Models [optional]

Challenge of large data dimension

Even a "small" image contains many individual signals: e.g. $200 \times 200 = \mathbf{40,000}$ or $75 \times 75 = 5,625$ pixels

- Computational image modelling is a very complex problem
- Large covariance matrices, e.g. $40,000 \times 40,000$ or $5,625 \times 5,625$
  - Mostly, computationally unfeasible
  - Much less images to process (typically, from dozens to thousands) than their pixel numbers
- Facial images: varying 3D viewing poses and directions, non-uniform illumination, varying facial expression, background and appearance, etc.

Image Sets in a p-D Vectorial Signal Space [optional]

Image dimension p is much larger than the number n of images

- Set $\mathbf{G}$ of n images of p pixels each, $p \gg n$:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1^\mathsf{T} \\ \mathbf{g}_2^\mathsf{T} \\ \vdots \\ \mathbf{g}_n^\mathsf{T} \end{bmatrix} \equiv \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1p} \\ g_{21} & g_{22} & \cdots & g_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{np} \end{bmatrix}$$

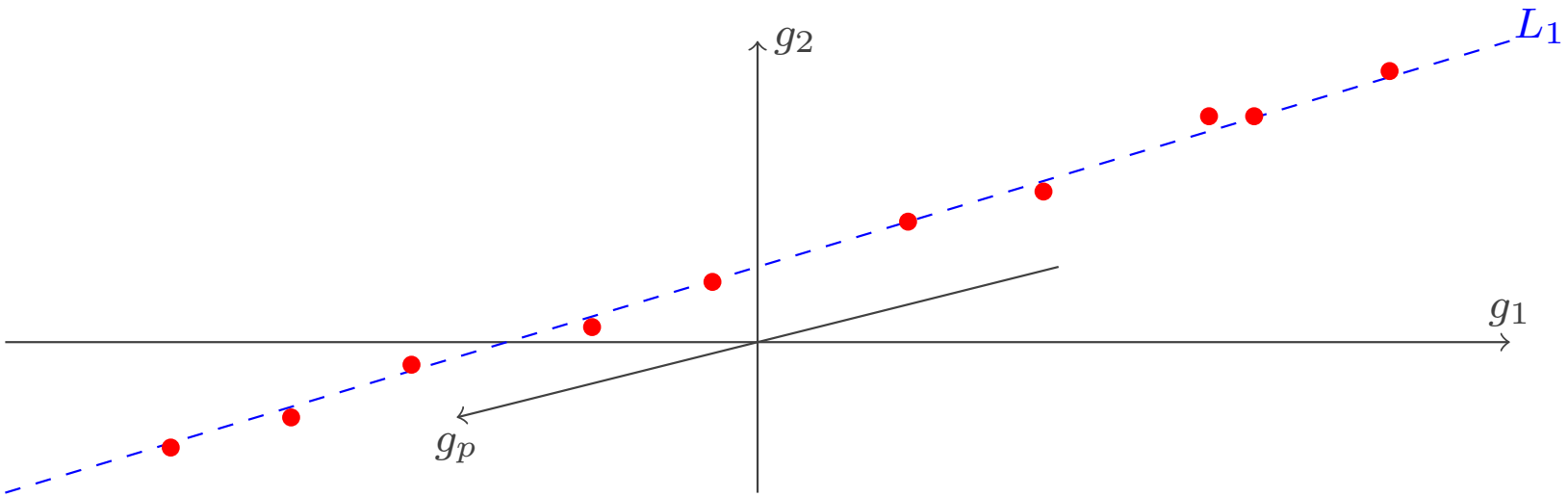
  - For instance, a set of 20 images of size 200×200 pixels: $m = 20 \ll p = 40,000$

- Centring $\mathbf{G}$ around its mean or centroid $\bar{\mathbf{g}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{g}_i$:

$$\mathbf{g}_i \leftarrow \mathbf{g}_i - \bar{\mathbf{g}}, \text{ i.e. } g_{ij} \leftarrow g_{ij} - \bar{g}_j;\ i = 1, \ldots, n;\ j = 1, \ldots, p$$

Subspace Model of Images [optional]

- Centred images $\mathbf{g}$ (points in a p-dimensional signal space) are scattered within or in a close vicinity of an m-dimensional linear subspace L_m; $m \leq n-1 \ll p$
  - In the latter case, the scatter is considerably larger within rather than outside the subspace L_m



Subspace Model of Images [optional]

- Scatter of a set $\mathbf{G}$ of centred images is described by the signal covariance $p \times p$ matrix $\widehat{\mathbf{C}} = \frac{1}{p-1}\mathbf{G}^\mathsf{T}\mathbf{G}$:

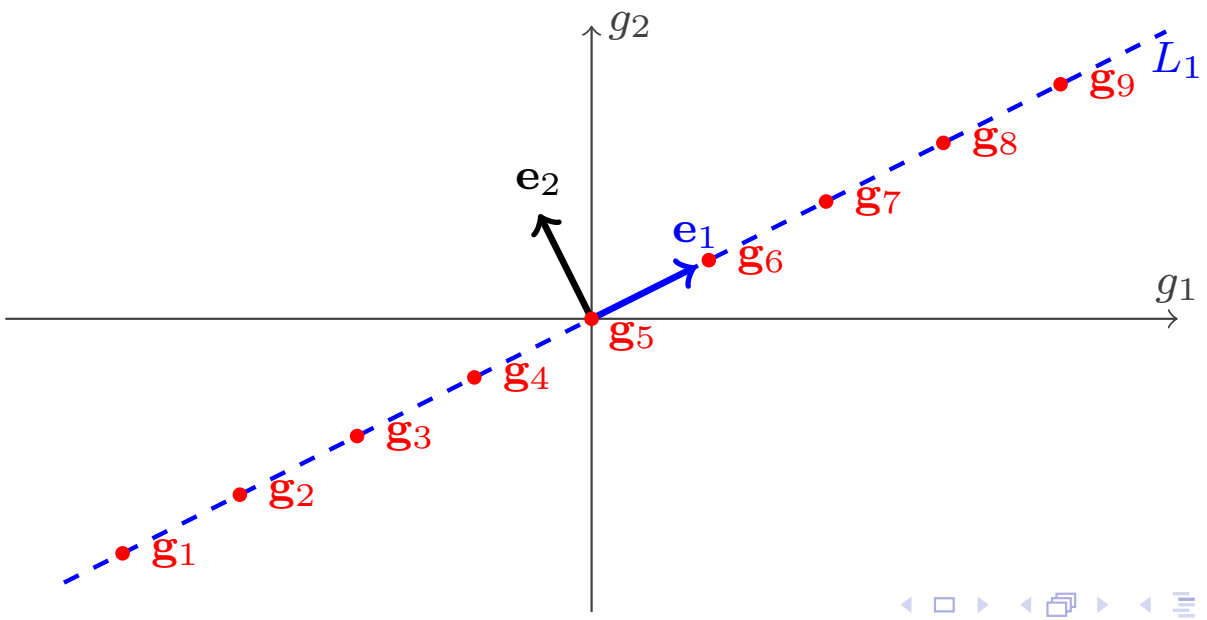
$$\widehat{\mathbf{C}} = \frac{1}{p-1}\begin{bmatrix} \sum_{i=1}^{n} g_{i1}^2 & \sum_{i=1}^{n} g_{i1}g_{i2} & \cdots & \sum_{i=1}^{n} g_{i1}g_{ip} \\ \sum_{i=1}^{n} g_{i2}g_{i1} & \sum_{i=1}^{n} g_{i2}^2 & \cdots & \sum_{i=1}^{n} g_{i2}g_{ip} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} g_{ip}g_{i1} & \sum_{i=1}^{n} g_{ip}g_{i2} & \cdots & \sum_{i=1}^{n} g_{ip}^2 \end{bmatrix}$$

- Orthonormal basis of the subspace L_m:
  - m top orthonormal eigenvectors $\mathbf{e}_i$ of the covariance matrix $\widehat{\mathbf{C}}$; $i = 1, \ldots, m$, with eigenvalues $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_m$
  - Orthonormal basis: $\mathbf{e}_\alpha^\mathsf{T}\mathbf{e}_\beta = \begin{cases} 1 & \alpha = \beta \\ 0 & \alpha \neq \beta \end{cases}$

Simplistic Example 1: $p = 2$; $n = 9$ [optional]

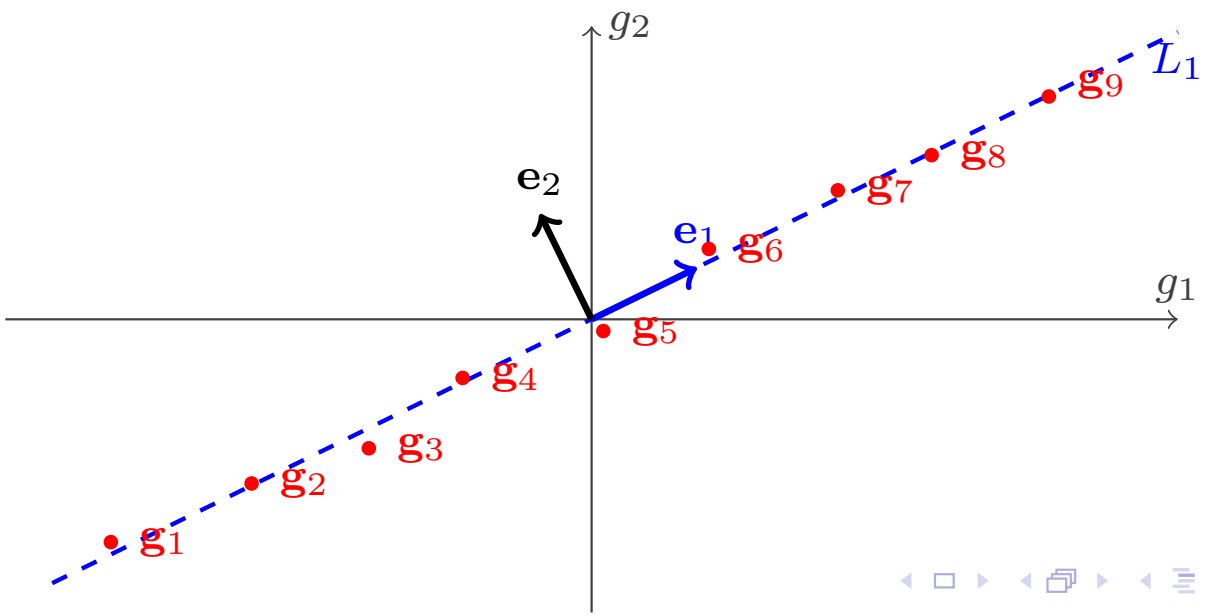
- $\mathbf{G}^\mathsf{T} = \begin{bmatrix} -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \\ -2 & -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5 & 2 \end{bmatrix}$
- $\widehat{\mathbf{C}} = \begin{bmatrix} 60 & 30 \\ 30 & 15 \end{bmatrix} \Rightarrow$ PCA: $\begin{vmatrix} 60-\lambda & 30 \\ 30 & 15-\lambda \end{vmatrix} = 0 \Rightarrow \lambda^2 - 75\lambda = 0$

  $\Rightarrow \lambda_1 = 75;\ \lambda_2 = 0 \Rightarrow \mathbf{e}_1 = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix};\ \mathbf{e}_2 = \frac{1}{\sqrt{5}} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \Rightarrow m = 1$



Simplistic Example 2: $p = 2$; $n = 9$ [optional]

- $\mathbf{G}^\mathsf{T} = \begin{bmatrix} -4.1 & -2.9 & -1.9 & -1.1 & 0.1 & 1 & 2.1 & 2.9 & 3.9 \\ -1.9 & -1.4 & -1.1 & -0.5 & -0.1 & 0.6 & 1.1 & 1.4 & 1.9 \end{bmatrix}$
- $\widehat{\mathbf{C}} = \begin{bmatrix} 59.08 & 28.86 \\ 28.86 & 14.18 \end{bmatrix} \Rightarrow$ PCA: $\begin{vmatrix} 59.08 - \lambda & 28.86 \\ 28.86 & 14.18 - \lambda \end{vmatrix} = 0 \Rightarrow$
  $\lambda^2 - 73.26\lambda + 4.8548 = 0 \Rightarrow \lambda_1 = 73.26;\ \lambda_2 = 0.066 \Rightarrow$
  $\mathbf{e}_1 = \begin{bmatrix} 0.898 \\ 0.439 \end{bmatrix}; \mathbf{e}_2 = \begin{bmatrix} 0.439 \\ -0.898 \end{bmatrix} \Rightarrow m \approx 1$



Subset of MIT-CBCL Face Recognition Database

(10 subjects, 2 directions of illumination: $n = 20$; $p = 200 \times 200 = 40,000$)

Centroid

PCA of an Image Set

- Principlal components: eigenvectors $\widehat{\mathbf{e}}$ of the intractable $p \times p$ covariance matrix $\widehat{\mathbf{C}} = \frac{1}{p-1}\mathbf{G}^{\mathsf{T}}\mathbf{G}$ of the centred images $\mathbf{G}$
  - For the MIT-CBCL set of 20 images, the $40,000 \times 40,000$ matrix cannot be processed using in a straightforward way any existing PCA software
  - Eigenvectors of $\widehat{\mathbf{C}} \Rightarrow \widehat{\mathbf{C}}\widehat{\mathbf{e}}_i = \widehat{\lambda}_i\widehat{\mathbf{e}}_i$; $\widehat{\mathbf{e}}_i^{\mathsf{T}}\widehat{\mathbf{e}}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$, are needed for finding an appropriate subspace model of images if it exists
  - But only $n-1$ top principal components out of $p-1$ ones could have possibly non-zero eigenvalues, i.e. 19 top eigenvectors out of 39,999 for the MIT-CBCL image set

Turk – Pentland's Tractable PCA of an Image Set

- Compute top $n-1$ eigenvectors $\widehat{\mathbf{e}}_i$ and eigenvalues $\widehat{\lambda}_i$ of $\widehat{\mathbf{C}}$ from eigenvectors $\mathbf{e}_i$ and eigenvalues λ_i of the $n \times n$ matrix $\mathbf{C} = \frac{1}{n-1}\mathbf{G}\mathbf{G}^\mathsf{T}$
- The latter matrix is tractable
- Its eigenvectors are linearly related to the goal eigenvectors:

$$\frac{1}{n-1}\mathbf{G}\mathbf{G}^\mathsf{T}\mathbf{e}_i = \lambda_i\mathbf{e}_i \Rightarrow \mathbf{G}^\mathsf{T}\left(\frac{1}{n-1}\mathbf{G}\mathbf{G}^\mathsf{T}\mathbf{e}_i\right) = \mathbf{G}^\mathsf{T}\left(\lambda_i\mathbf{e}_i\right) \Rightarrow$$
$$\frac{1}{p-1}\mathbf{G}^\mathsf{T}\mathbf{G}\left(\mathbf{G}^\mathsf{T}\mathbf{e}_i\right) = \frac{n-1}{p-1}\lambda_i\left(\mathbf{G}^\mathsf{T}\mathbf{e}_i\right)$$

- Eigenvectors $\frac{1}{p-1}\mathbf{G}^\mathsf{T}\mathbf{G}\widehat{\mathbf{e}}_i = \widehat{\lambda}_i\widehat{\mathbf{e}}_i \Rightarrow \widehat{\mathbf{e}}_i = \frac{1}{\sqrt{(n-1)\lambda_i}}\mathbf{G}^\mathsf{T}\mathbf{e}_i$;

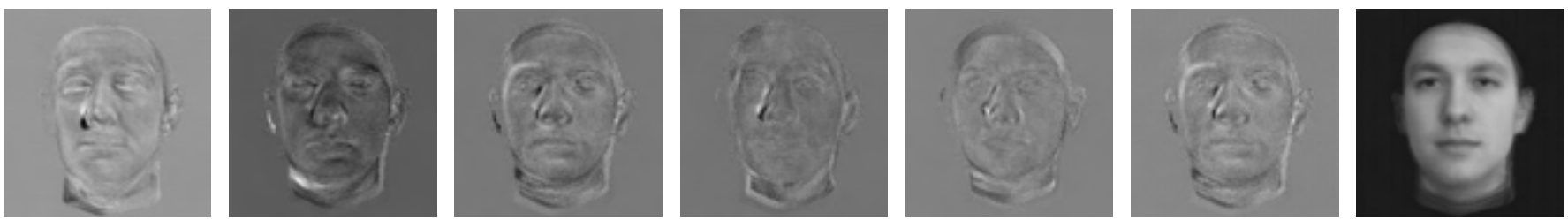
$$\widehat{\lambda}_i = \frac{n-1}{p-1}\lambda_i;\ \widehat{\mathbf{e}}_i^\mathsf{T}\widehat{\mathbf{e}}_k = \frac{1}{(n-1)\lambda_i}\mathbf{e}_i^\mathsf{T}\underbrace{\mathbf{G}\mathbf{G}^\mathsf{T}\mathbf{e}_k}_{(n-1)\lambda_k\mathbf{e}_k} = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases}$$

Eigenfaces for the MIT-CBCL Image Subset

$(n = 20;\ p = 200 \times 200 = 40,000;\ \lambda_i \times 10^6)$

$\lambda_1 = 45$ $\lambda_2 = 14$ $\lambda_3 = 7.9$ $\lambda_4 = 6.4$ $\lambda_5 = 4.0$ $\lambda_6 = 3.3$ $\lambda_7 = 2.7$

$\lambda_8 = 2.2$ $\lambda_9 = 1.5$ $\lambda_{10} = 1.3$ $\lambda_{11} = 0.2$ $\lambda_{12} = 0.09$ $\lambda_{13} = 0.03$ $\lambda_{14} = 0.02$



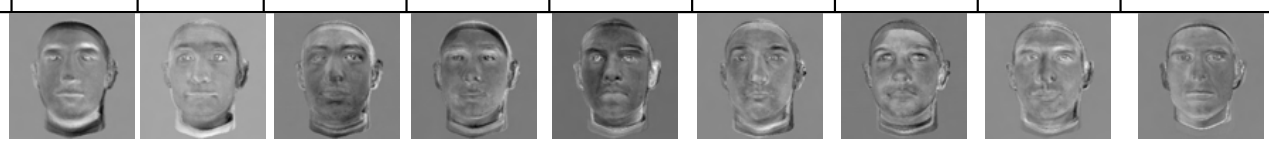
$\lambda_{15} = 0.02$ $\lambda_{16} = 0.006$ $\lambda_{17} = 0.004$ $\lambda_{18} = 0.004$ $\lambda_{19} = 0.003$ $\lambda_{20} = 0$ **Centroid**

Eigenfaces in Image Representation and Restoration

- Choosing a representative subspace L_m for a given set $\mathbf{G}$ of n images
- **Criterion**: by the relative variance of image signals $\nu_i = \frac{\sum_{k=1}^{i} \lambda_k}{\sum_{k=1}^{19} \lambda_k}$

  in the subspace L_m spanned over the top m eigenfaces:

	$\widehat{\mathbf{e}}_1$	$\widehat{\mathbf{e}}_2$	$\widehat{\mathbf{e}}_3$	$\widehat{\mathbf{e}}_4$	$\widehat{\mathbf{e}}_5$	$\widehat{\mathbf{e}}_6$	$\widehat{\mathbf{e}}_7$	$\widehat{\mathbf{e}}_8$	$\widehat{\mathbf{e}}_9$
$\lambda_m \times 10^6$	45	14	7.9	6.4	4.0	3.3	2.7	2.2	1.5
ν_m,%	50	67	76	83	87	91	94	96	98



Eigenfaces in Image Representation and Restoration

- The chosen subspace L_m ensures a desired fraction of the total variance, e.g. $\geq 80\%$, or $\geq 90\%$, etc.

PCA-based restoration of an image $\mathbf{g}$:

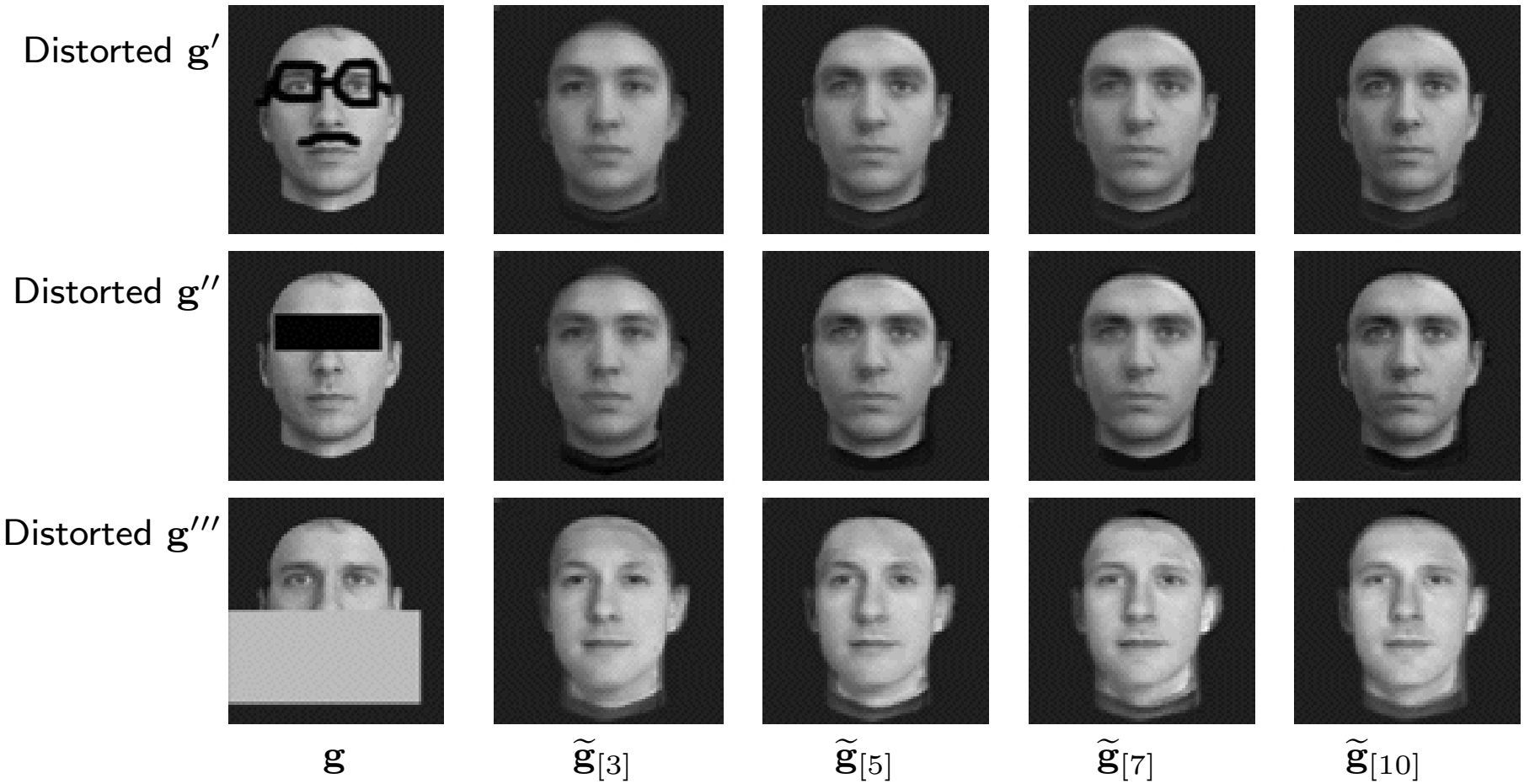
- Image projection $\widetilde{\mathbf{g}}_{[m]}$ onto L_m built for a set of "noiseless" (ideal) images:

$$\widetilde{\mathbf{g}}_{[m]} = \sum_{k=1}^{m} \alpha_k \widehat{\mathbf{e}}_k \quad \text{where} \quad \alpha_k = \mathbf{g}^{\mathsf{T}} \widehat{\mathbf{e}}_k$$

$\mathbf{g} \in \mathbf{G}$

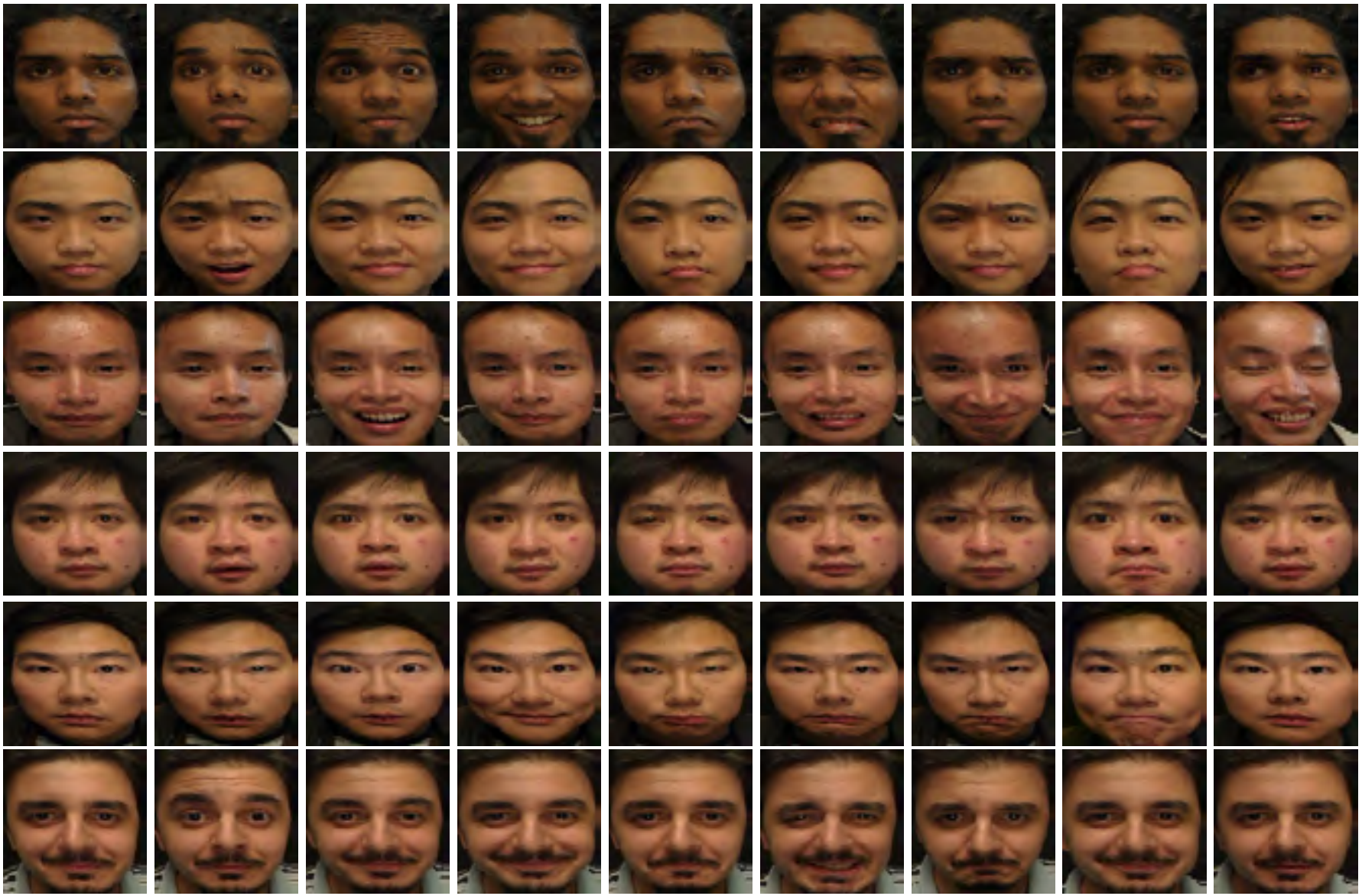
$\mathbf{g}$ $\quad \widetilde{\mathbf{g}}_{[3]}$ $\quad \widetilde{\mathbf{g}}_{[5]}$ $\quad \widetilde{\mathbf{g}}_{[7]}$ $\quad \widetilde{\mathbf{g}}_{[10]}$

Eigenfaces in Image Representation and Restoration



Under large distortions, the projection onto L_m cannot restore the original image...

54 faces with different expression (6 subjects; 9 expressions)



Mean Image and Top-rank Eigenface

Mean $\boldsymbol{\mu} = \frac{1}{54} \sum_{i=1}^{54} \mathbf{g}_i$

Maximal eigen-value 0.316 (31.6% of the total variance)

Fraction of the total variance for i top-rank eigenfaces: $c_i = \sum_{j=1}^{i} \lambda_i$

i	1	2	3	4	5	6
λ_i	0.316	0.211	0.094	0.084	0.057	0.032
c_i	0.316	0.527	0.621	**0.705**	0.762	**0.794**
$\mathbf{e}_i$						

i	7	8	9	10	11	12
c_i	0.816	0.831	0.846	0.861	0.873	0.893
						

i	13	14	15	16	17	18
c_i	**0.902**	0.909	0.916	0.923	0.929	0.935
						

Limitations of SVD and PCA

- Matrices $\mathbf{U}$ and $\mathbf{V}$ in SVD are not at all **sparse**
  - Thus: significant cost of computing and using them
    - Sparse matrices yield lower computational costs!
  - Sparseness of very large matrices expected in computational science reflects typical "local" behaviour of practical problems
    - A large world with small neighbourhoods
  - **But**: orthogonal eigenvectors and singular vectors are not local...
- SVD and PCA work only with 2D data (i.e. **matrices** $\mathbf{A} = [a_{ij}]_{1,1}^{m,n}$ with two indices)
  - Eigenvectors, PCA, and SVD have no perfect extension to data with larger number of indices
    - E.g. **tensors** with three ($\mathbf{A} = [a_{ijk}]_{1,1,1}^{m,n,q}$) or four indices ($\mathbf{A} = [a_{ijkl}]_{1,1,1,1}^{m,n,q,s}$) being used frequently in physics and engineering

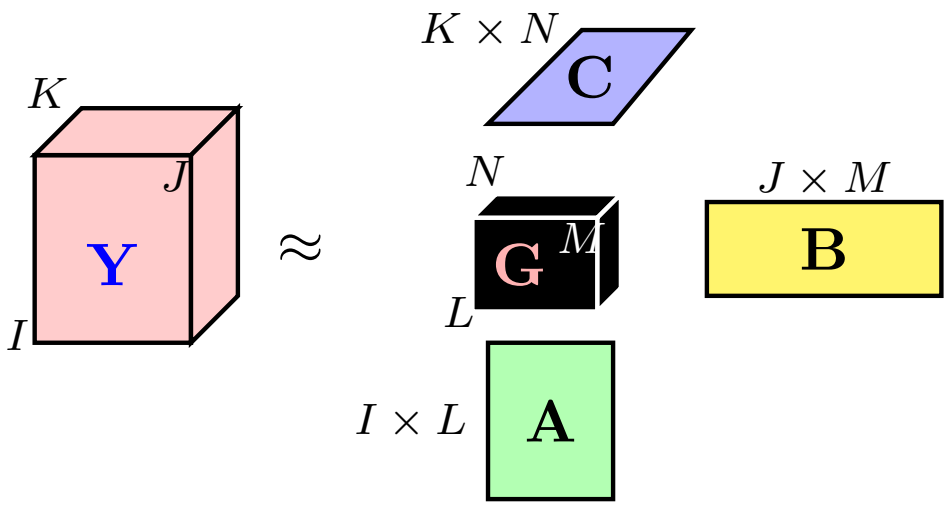
Tucker Model

Multiway array: data indexed with 3 or more indices:

$$\mathbf{Y}^{I\times J\times K} = \{y_{ijk} : \ i = 1,\ldots,I;\ j = 1,\ldots,J;\ k = 1,\ldots,K\}$$

- 1 index – a vector; 2 indices – a matrix; $\geq$3 indices – a tensor
- 3 **ways** (or **modes**) in a tensor: row ($\rightarrow$), column($\downarrow$), tube($\nearrow$)

Model of a 3[rd]-order tensor $\mathbf{Y}$: $y_{ijk} \approx \sum_{l=1}^{L}\sum_{m=1}^{M}\sum_{n=1}^{N} a_{il}b_{jm}c_{kn}g_{lmn}$



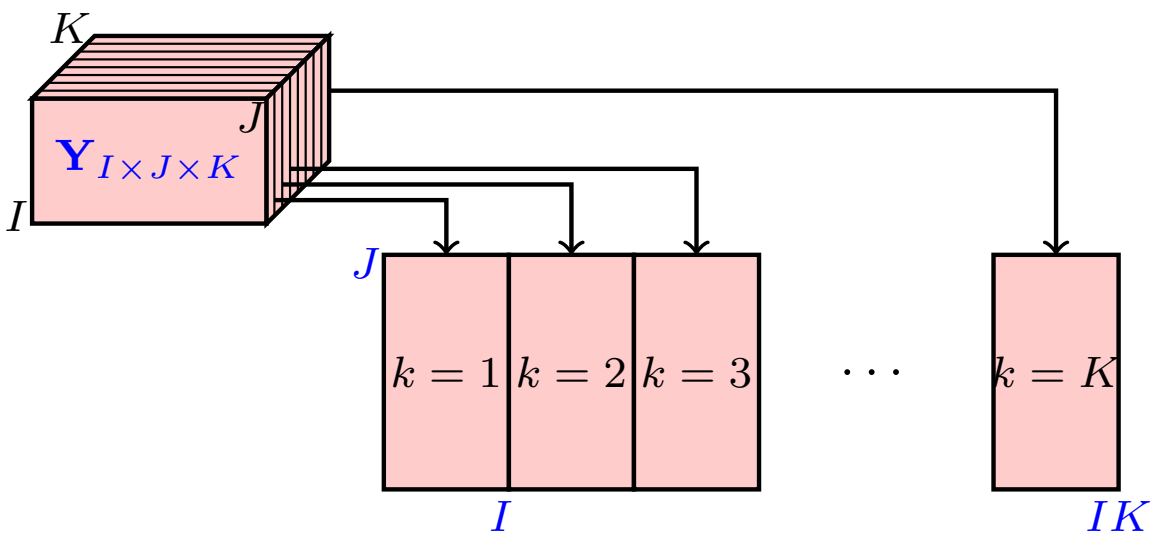
Forming a Tucker Model

Decomposition of a tensor $\mathbf{Y}$:

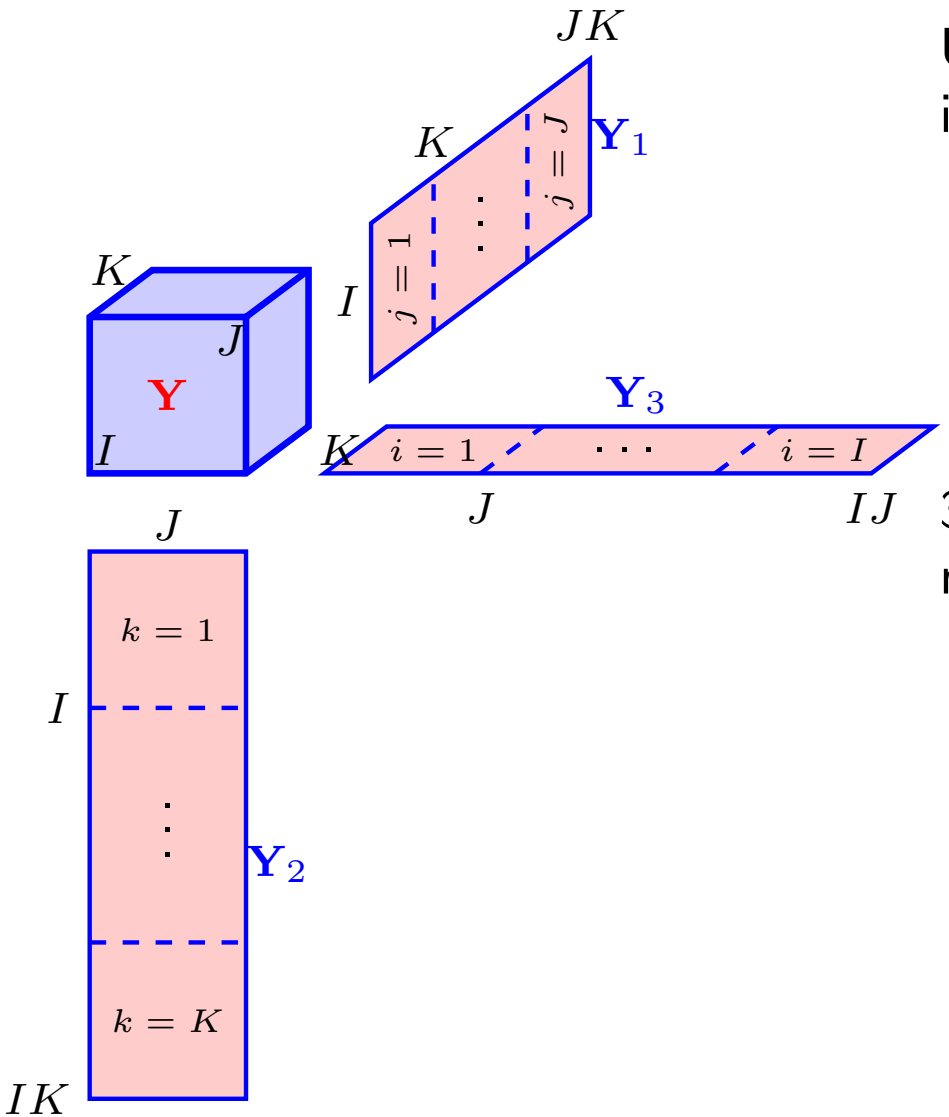
- 3 orthogonal **mode matrices** $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, weighted by the **core** 3-way array $\mathbf{G}$ of size $L \times M \times N$
- The core array $\mathbf{G}$ is analogous to the diagonal matrix $\mathbf{D}$ in SVD

Unfolding of a *tensor* – its rearrangement into a *matrix*

- Possible unfolding of a 3-way array $\mathbf{Y}_{I \times J \times K}$ into a $J \times IK$ matrix:



Forming a Tucker Model: 3–Mode SVD



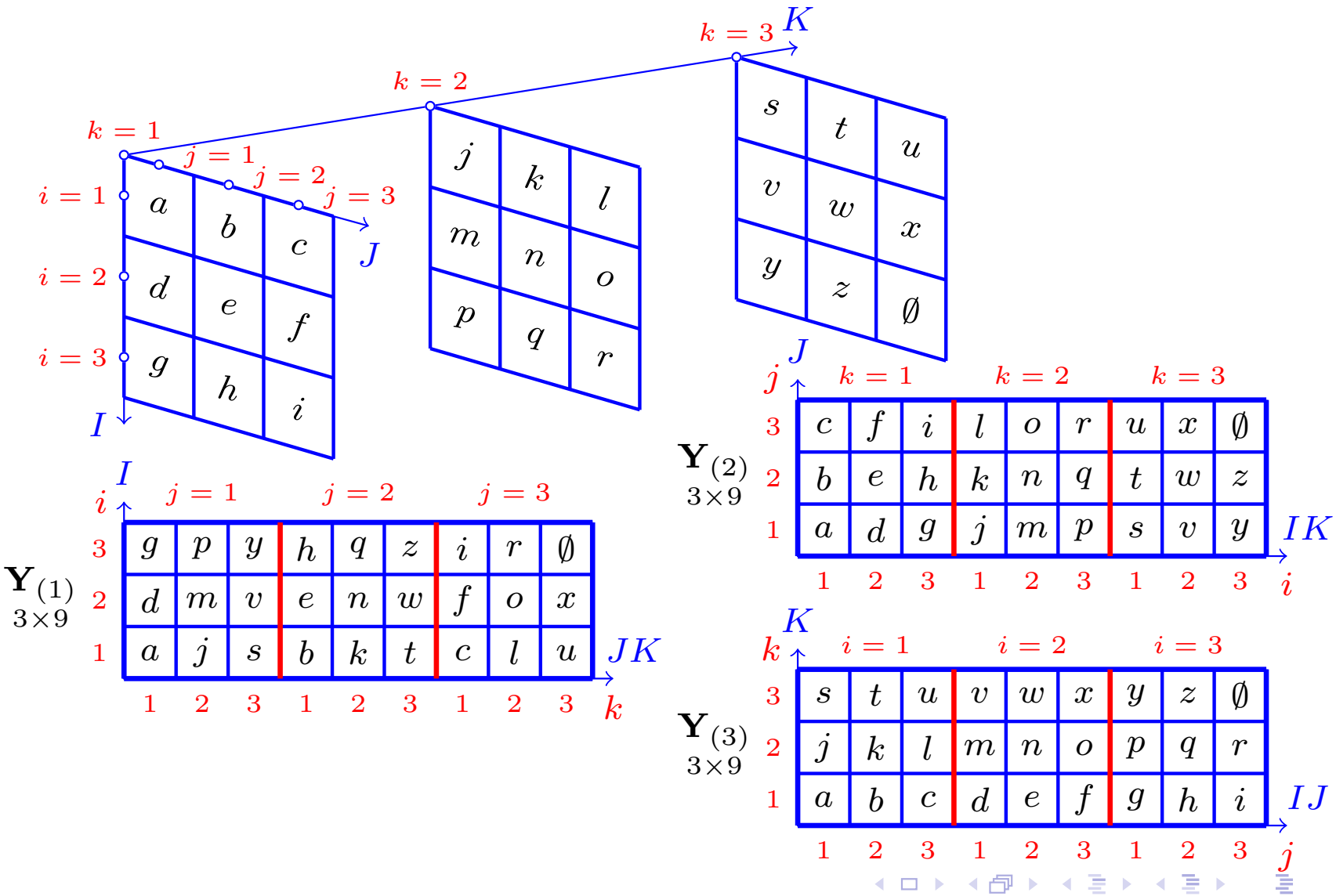
Unfolding a 3-way array $\mathbf{Y}_{I\times J\times K}$ into the three matrices $\mathbf{Y}_{(\ldots)}$:

- $I \times JK$ matrix $\mathbf{Y}_{(1)}$
- $J \times IK$ matrix $\mathbf{Y}_{(2)}$
- $K \times IJ$ matrix $\mathbf{Y}_{(3)}$

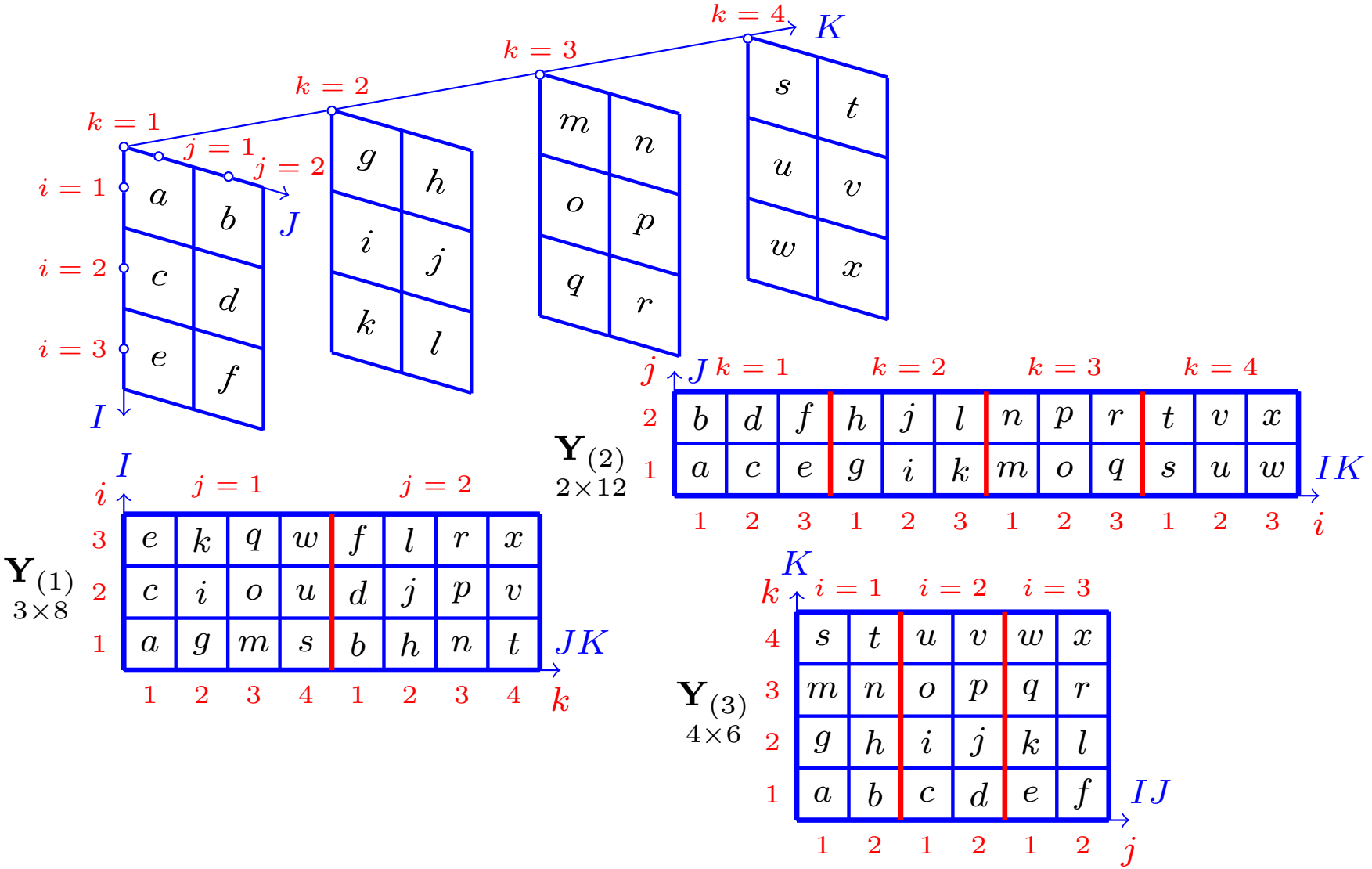
3-mode SVD to form truncated matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$:

- $\mathbf{Y}_{(1)} \Longrightarrow I \times L$ matrix $\mathbf{A}$; $L \leq JK$
- $\mathbf{Y}_{(2)} \Longrightarrow J \times M$ matrix $\mathbf{B}$; $M \leq IK$
- $\mathbf{Y}_{(3)} \Longrightarrow K \times N$ matrix $\mathbf{C}$; $N \leq IJ$

An Example: Unfolding $\mathbf{Y}_{3\times3\times3}$ into Three Matrices

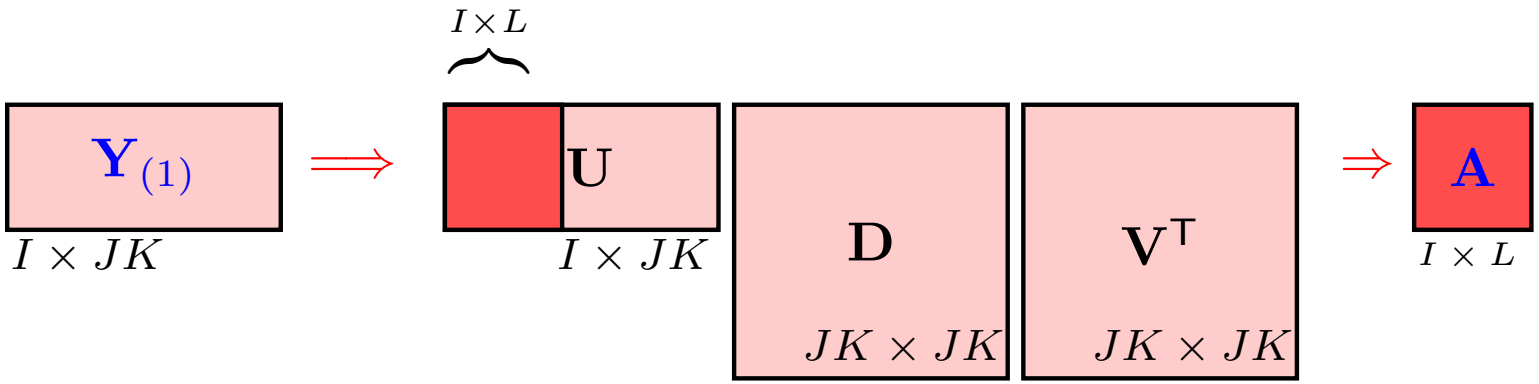


An Example: Unfolding $\mathbf{Y}_{3\times2\times4}$ into Three Matrices

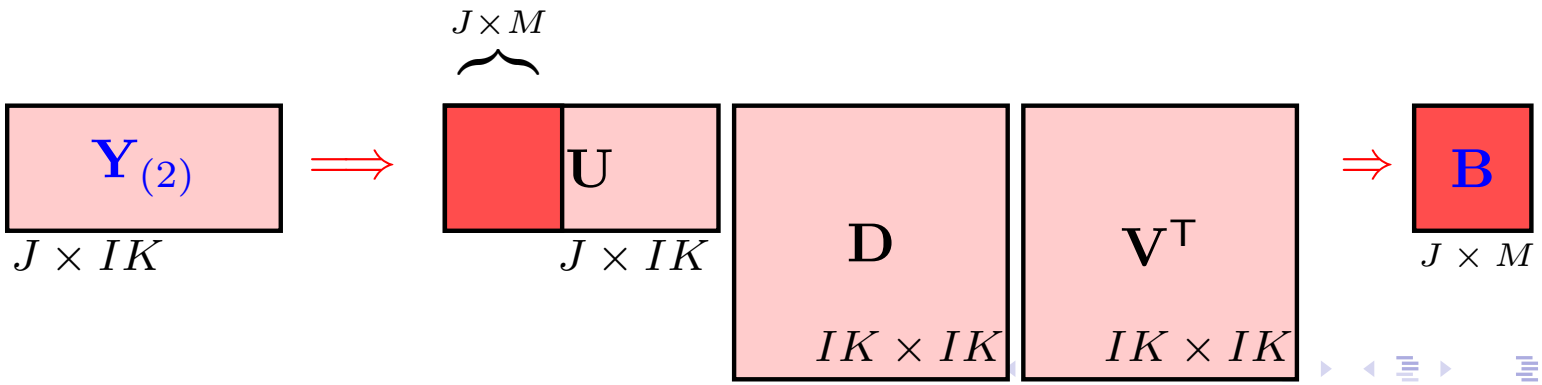


Forming a Tucker Model: 3–Mode SVD

1. Find the $I \times L$ matrix $\mathbf{A}$: SVD of the unfolded $I \times JK$ matrix $\mathbf{Y}_{(1)}$; taking $\mathbf{A}$ from the left matrix of SVD: $L \leq JK$

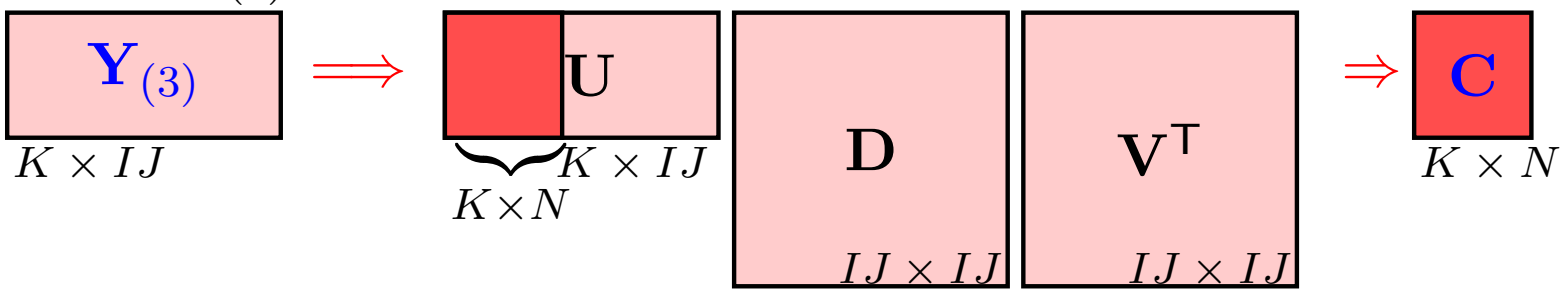


2. Find the $J \times M$ matrix $\mathbf{B}$: SVD of the unfolded $J \times IK$ matrix $\mathbf{Y}_{(2)}$; taking $\mathbf{B}$ from the left matrix of SVD: $M \leq IK$



Forming a Tucker Model: 3–Mode SVD

3. Find the $K \times N$ matrix $\mathbf{C}$: SVD of the unfolded $K \times IJ$ matrix $\mathbf{Y}_{(3)}$; taking $\mathbf{C}$ from the left matrix of SVD: $N \leq IJ$

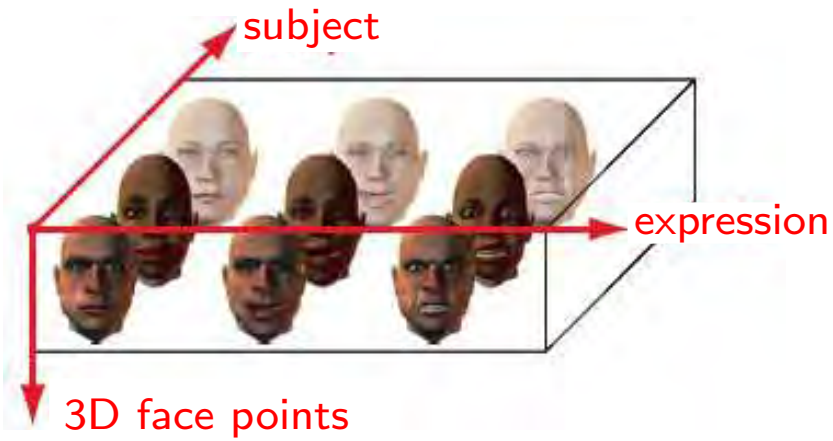


Compute the core tensor $\mathbf{G}$:

$$g_{lmn} = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} a_{il} b_{jm} c_{kn} y_{ijk}$$

MIT CSAI Lab: Multilinear Models of 3D Face Surfaces

D.Vlasic e.a., Face Transfer with Multilinear Models: ACM Trans. on Graphics, vol.24 (3), 426-433, 2005



Bilinear data tensor: 3 subjects $\times$ 3 expressions $\times$ $30,000$ 3D face points

Using a trilinear model to combine pose and identity from V_1 with expression from V_2 and mouth articulation from V_3 in order to blend these attributes back to V_1