Singular Value Decomposition (SVD)

March 30, 2021

- One of the most important decompositions in the linear algebra is the singular value decomposition (SVD).
- The SVD is extremely useful in all areas of science, engineering, and statistics, such as signal processing, least squares fitting of data, pattern recognition, statistics,image compression, image denoising and process control.
- SVD, has also many applications in the mathematics, such as computing the pseudo inverse, matrix approximation, and determining the rank, range, and null space of a matrix.

Eigenvectors and Eigenvalues

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

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

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

Example of deriving eigenvalues and eigenvectors

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

$$\rightarrow \lambda_1 = 1,\ \lambda_2 = 3,\ \mathbf{u}_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix},\ \mathbf{u}_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Singular Value

- Consider the matrix $\mathbf{F} \in \mathbb{C}^{I_1 \times I_2}$.
- Set $\mathbf{G} = \mathbf{F}^H \mathbf{F}$.
- $\mathbf{G}$ is an hermitian positive semi-definite matrix (means $\forall \mathbf{x} \in \mathbb{C}^{I_1}, \mathbf{x}^H \mathbf{G} \mathbf{x} \geq 0$.
- So the eigenvalues of $\mathbf{G}$ are non-negative.
- Suppose that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{I_2} \geq 0$ are the eigenvalues of $\mathbf{G}$.
- Set $\sigma_i = \sqrt{\lambda_i},\ i = 1, 2, \cdots, I_2$.
- $\{\sigma_i\}$ are called the *singular values* of $\mathbf{F}$.

Matrix SVD:

Every matrix $\mathbf{F} \in \mathbb{C}^{I_1 \times I_2}$ can be written as the product

$$\mathbf{F} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^{\mathrm{H}}, \tag{1}$$

where

- $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{I_1}] \in \mathbb{C}^{I_1 \times I_1}$ is an unitary matrix.
- $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{I_2}] \in \mathbb{C}^{I_2 \times I_2}$ is an unitary matrix.

$$(\mathbf{U}^H\mathbf{U} = \mathbf{U}\mathbf{U}^H = \mathbf{I}_{I_1}, \qquad \mathbf{V}^H\mathbf{V} = \mathbf{V}\mathbf{V}^H = \mathbf{I}_{I_2})$$

Matrix SVD:

- $\mathbf{S} \in \mathbb{C}^{I_1 \times I_2}$ have the following properties:
  - **Pseudodiagonality:**

$$\mathbf{S} = \operatorname{diag}\left(\sigma_1, \sigma_2, \ldots, \sigma_{\min(I_1, I_2)}\right).$$

  - **Ordering:**

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(I_1, I_2)} \geq 0.$$

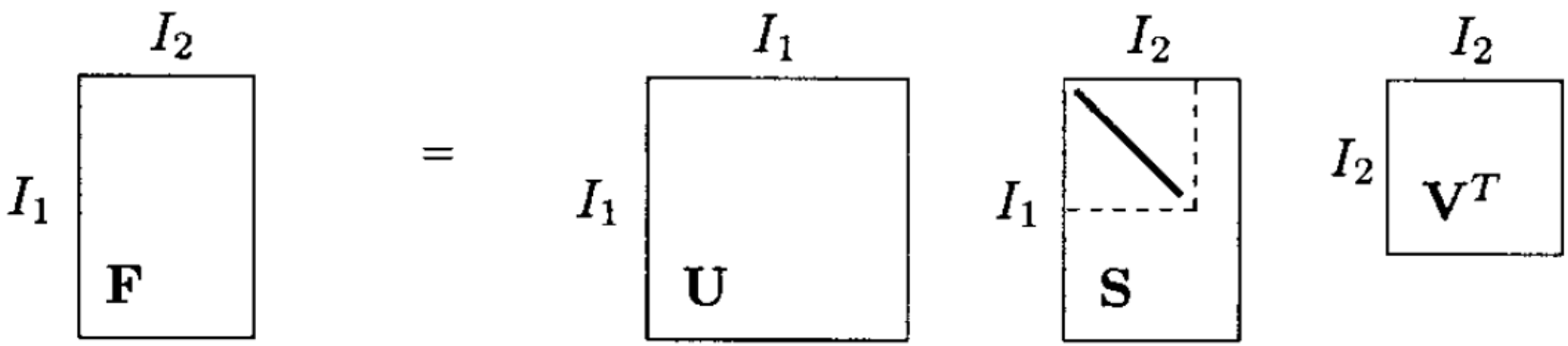


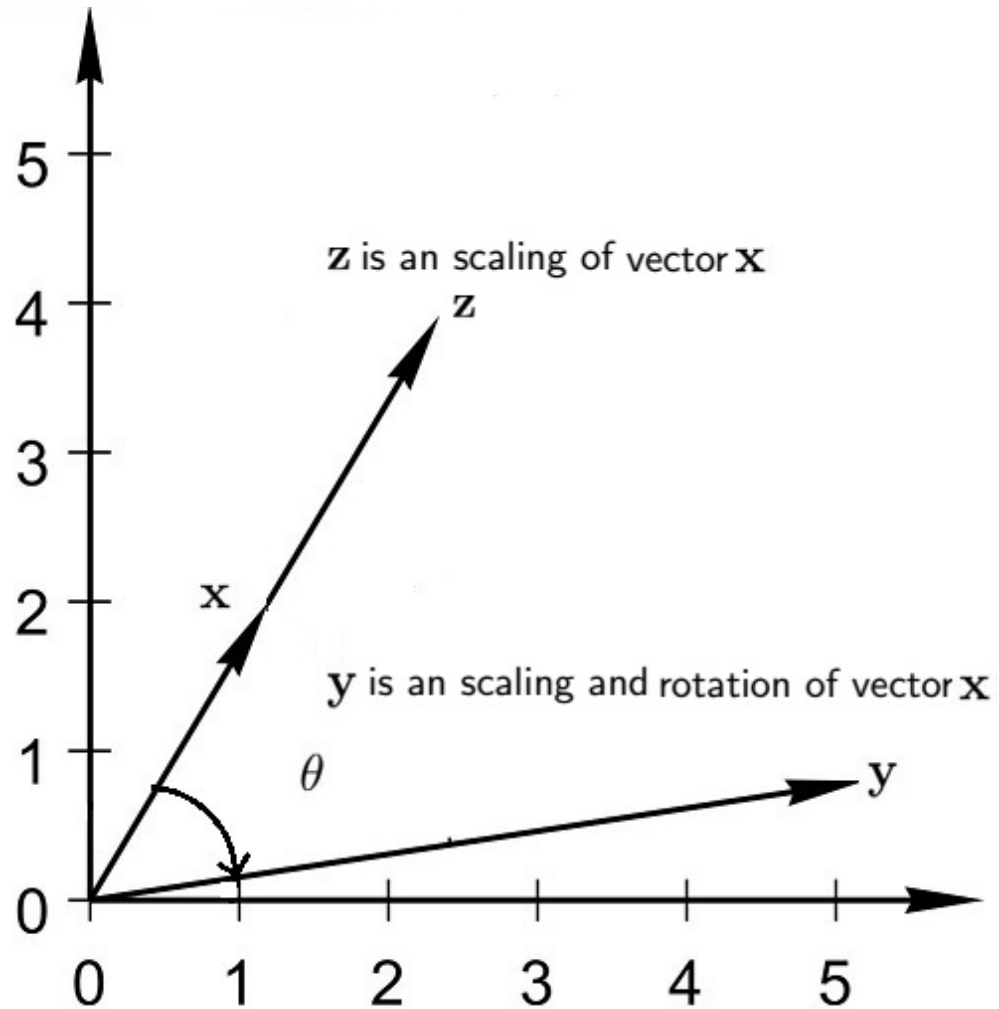
Figure: Visualization of the matrix SVD.

Points

- The diagonal entries σ_j are the singular values of $\mathbf{F}$.
- The columns $\{\mathbf{u}_i\}$ of $\mathbf{U}$ and the columns $\{\mathbf{v}_i\}$ of $\mathbf{V}$ are the left and right singular vectors of $\mathbf{F}$, respectively.
- The number r of non-zero σ_r is exactly the *rank* of matrix $\mathbf{F}$.
- The singular values of $\mathbf{F}$ are the same as those of $\mathbf{F}^H$.

What does a matrix do to a vector?

- Set $\mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $\mathbf{A} = \begin{pmatrix} 1 & 2 \\ -3 & 2 \end{pmatrix}$,
- $\mathbf{z} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$,
- $\mathbf{y} = \mathbf{A}\mathbf{x} = \begin{pmatrix} 1 & 2 \\ -3 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \end{pmatrix}$.



What happens to a circle under a matrix multiplication?

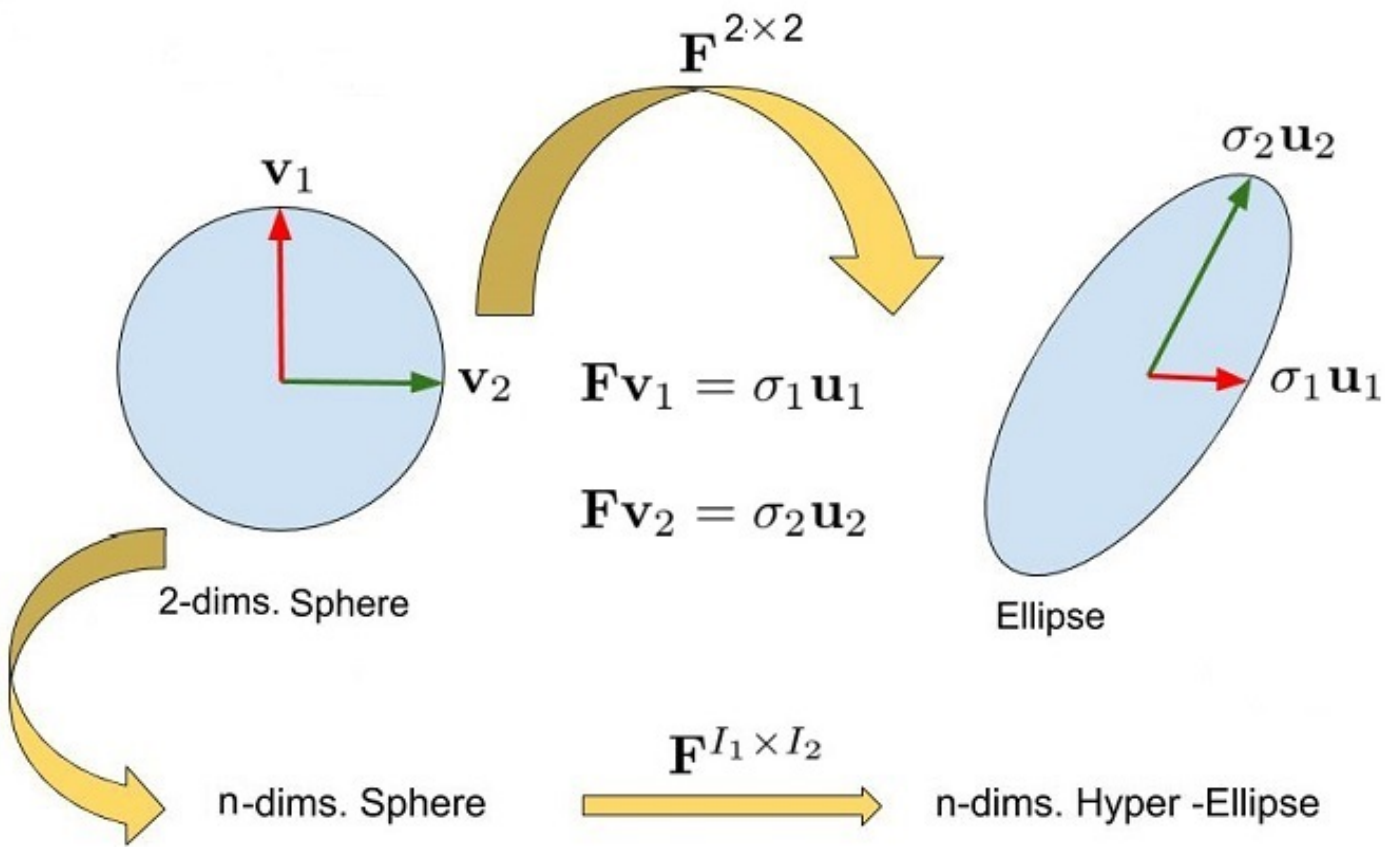


Figure: Geometric interpretation of the operation of a 2×2 matrix indicated by its effect on a 2-dimensional sphere (circle) and the two orthogonal vectors $\mathbf{v}_1$ and $\mathbf{v}_2$.

The interpretation of above figure

- For every linear map $\mathbf{F} : \mathbb{C}^{I_2} \to \mathbb{C}^{I_1}$, one can find orthonormal bases of $\mathbb{C}^{I_1}$ and $\mathbb{C}^{I_2}$ such that $\mathbf{F}$ maps the i-th basis vector of $\mathbb{C}^{I_2}$ to a non-negative multiple of the i-th basis vector of $\mathbb{C}^{I_1}$, and sends the left-over basis vectors to zero.

The effect of matrix multiplication on a n-dims sphere

- Consider $\mathbf{F} \in \mathbb{C}^{I_1 \times I_2}$ and suppose that $\min\{I_1, I_2\} = I_2$,
- $\mathbf{F}\mathbf{v}_i = \sigma_i \mathbf{u}_i,\ i = 1, \cdots, I_2$, and $\mathbf{F}\mathbf{v}_i = 0,\ \textit{for } i > I_2$,
- So, $\mathbf{F}[\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{I_2}] = [\sigma_1 \mathbf{u}_1, \sigma_2 \mathbf{u}_2, \cdots, \sigma_{I_2} \mathbf{u}_{I_2}]$,
- Then $\mathbf{FV} = \mathbf{US}$, where
  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{I_1}]$, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{I_2}]$ and

$$\mathbf{S} = \begin{pmatrix} \sigma_1 & \cdots & 0 \\ 0 & \ddots & 0 \\ \vdots & \cdots & \sigma_{I_2} \\ 0 & 0 & 0 \end{pmatrix},$$

- Since $\mathbf{V}^{-1} = \mathbf{V}^H$, we have

$$\mathbf{FVV}^H = \mathbf{USV}^H \rightarrow \mathbf{F} = \mathbf{USV}^H, \text{ (SVD of } \mathbf{F}).$$

Geometrical interpretation of SVD

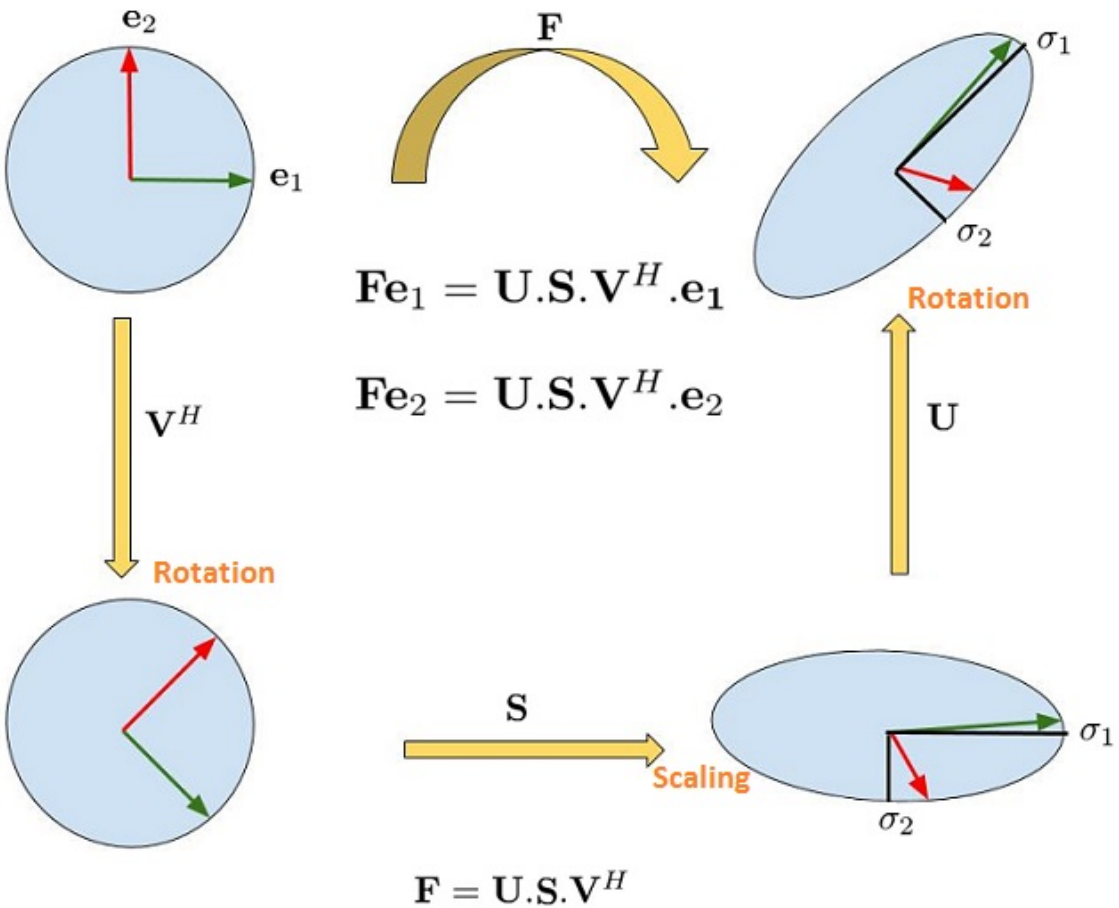


Figure: Geometric interpretation of the SVD of a 2×2 matrix indicated by its effect on the unit circle and two canonical unit vectors $\mathbf{e}_1$ and $\mathbf{e}_2$.

Geometrical Interpretations of SVD

Rotation, coordinate scaling, and reflection

- The SVD represents the essential geometry of a linear transformation.
- It tells us that every linear transformation is a composition of three fundamental actions:
  - **V** represents a rotation or reflection of vectors in the l_2-dimensional space.
  - **S** represents coordinate-by-coordinate scaling by the factor σ_i.
  - **U** represents a rotation or reflection of vectors in the l_1-dimensional space.
- The diagram shows that every linear transformation is fundamentally a rotation or reflection, followed by a scaling, followed by another rotation or reflection.

Geometrical interpretation of SVD

Rotation, coordinate scaling, and reflection

- Consider the circle of radius one in $\mathbb{C}^{I_2}$. The linear map $\mathbf{F}$ maps this circle onto an ellipsoid in $\mathbb{C}^{I_1}$.
- The lengths of the semi-axes of the ellipse are the singular values of $\mathbf{F}$, namely σ_1, σ_2.
- Singular values encode magnitude of the semiaxis, while singular vectors encode direction.

Example.

Let $\mathbf{F} \in \mathbb{C}^{4\times 3}$. Then

$$\mathbf{F} = (\mathbf{u}_1\ \mathbf{u}_2\ \mathbf{u}_3\ \mathbf{u}_4) \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \\ 0 & 0 & 0 \end{pmatrix} (\mathbf{v}_1\ \mathbf{v}_2\ \mathbf{v}_3)\,.$$

Example.

Compute the SVD in MATLAB:

$>>$ $\mathbf{F} = [1 \ -2 \ 3; \ -2 \ 3 \ 1; \ 2 \ -4 \ 6; \ -1 \ 2 \ -3];$

$>>$ $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = svd(\mathbf{F});$

$>>$ $disp(\mathbf{S})$

$$\mathbf{S} = \begin{pmatrix} 9.2780 & 0 & 0 \\ 0 & 3.4524 & 0 \\ 0 & 0 & 0.0000 \\ 0 & 0 & 0 \end{pmatrix}.$$

Note. The matrix $\mathbf{F}$ has rank 2. $\mathbf{U}$ is 4×4 and $\mathbf{V}$ is 3×3.

Proposition.

Let $\mathbf{F} = \mathbf{U}\mathbf{S}\mathbf{V}^H$, and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = 0$.
Then

- $\mathbf{F}\mathbf{v}_i = \sigma_i \mathbf{u}_i,\ \mathbf{F}^H\mathbf{u}_i = \sigma_i \mathbf{v}_i,\ 1 \leq i \leq \min\{l_1, l_2\}$.
- The columns of matrix $\mathbf{U}$ are the eigenvectors of $\mathbf{F}\mathbf{F}^H$ with corresponding eigenvalues
  $\lambda_1 = \sigma_1^2,\ \lambda_2 = \sigma_2^2,\ \cdots, \lambda_r = \sigma_r^2, \lambda_{r+1} = 0, \cdots, \lambda_{l_1} = 0$.
- The columns of matrix $\mathbf{V}$ are the eigenvectors of $\mathbf{F}^H\mathbf{F}$ with corresponding eigenvalues
  $\lambda_1 = \sigma_1^2,\ \lambda_2 = \sigma_2^2,\ \cdots, \lambda_r = \sigma_r^2, \lambda_{r+1} = 0, \cdots, \lambda_{l_2} = 0$.
- null($\mathbf{F}$)=span$\{\mathbf{v}_{r+1}, \cdots, \mathbf{v}_{l_2}\}$. (null($\mathbf{F}$)=$\{\mathbf{x} \in \mathbb{C}^{l_2} | \mathbf{F}\mathbf{x} = 0\}$)
- range($\mathbf{F}$)=span$\{\mathbf{u}_1, \cdots, \mathbf{u}_r\}$.
  (range($\mathbf{F}$)=$\{\mathbf{y} \in \mathbb{C}^{l_1} | \mathbf{y} = \mathbf{F}, \mathbf{x}\ \forall \mathbf{x} \in \mathbb{C}^{l_2}\}$)

Proposition.

Let $\mathbf{F} = \mathbf{USV}^H$. Then

$$\mathbf{FF}^H = \mathbf{U}(\mathbf{SS}^H)\mathbf{U}^H,$$

and

$$\mathbf{F}^H\mathbf{F} = \mathbf{V}(\mathbf{S}^H\mathbf{S})\mathbf{V}^H.$$

Remark.

The sets $(\sigma_i^2, \mathbf{u}_i)$ and $(\sigma_i^2, \mathbf{v}_i)$ are *eigenpairs* of $\mathbf{FF}^H$ and $\mathbf{F}^H\mathbf{F}$, respectively. Computing the singular values is equivalent to solving a symmetric eigenvalue problem.

Pseudo Inverse

- In mathematics and, in particular, in linear algebra, the pseudo inverse of a matrix **A** is the most widely known generalization of the matrix inverse.
- A common use of the pseudoinverse is to compute a "best fit" (least squares) solution to a system of linear equations. Another use is to find the minimum (Euclidean) norm solution to a system of linear equations with multiple solutions.

Pseudo Inverse

Definition

Let $\mathbf{D} = diag_r(\sigma_1, \cdots, \sigma_r)$, such that $\sigma_i \neq 0,\ i = 1, 2, \cdots, r$ and $\mathbf{S}$ be a $I_1 \times I_2$ matrix as

$$\mathbf{S} = \begin{pmatrix} \mathbf{D} & 0 \\ 0 & 0 \end{pmatrix}.$$

The pseudo inverse of $\mathbf{S}$ is denoted by $\mathbf{S}^\dagger$ and defined as

$$\mathbf{S}^\dagger = \begin{pmatrix} \mathbf{D}^{-1} & 0 \\ 0 & 0 \end{pmatrix}.$$

Definition.

Let $\mathbf{F}$ be a $I_1 \times I_2$ matrix and $\mathbf{F} = \mathbf{USV}^H$. The pseudo inverse of $\mathbf{F}$ is denoted by $\mathbf{F}^\dagger$ and defined as $\mathbf{F}^\dagger = \mathbf{VS}^\dagger\mathbf{U}^H$.

Remark.

- If $\mathbf{F} \in \mathbb{C}^{I_1 \times I_2}$ be a non-singular matrix, then $\mathbf{F}^\dagger = \mathbf{F}^{-1}$.
- If $\mathbf{F} \in \mathbb{C}^{I_1 \times I_2}$ and $I_1 \geq I_2$ and rank($\mathbf{F}$)=I_2, then $\mathbf{F}^\dagger = (\mathbf{F}^H\mathbf{F})^{-1}\mathbf{F}^H$. Tall matrix

If I1<=I2 (broad matrix), then F+ = F^H(FF^H)^{-1} (right inverse)

Proposition 1. Least squares problem

Let $\mathbf{A} \in \mathbb{C}^{I_1 \times I_2}$ and $\mathbf{b} \in \mathbb{C}^{I_1}$. If $\mathbf{x}^* = \mathbf{A}^\dagger \mathbf{b}$, then

- $\min_{\mathbf{x} \in \mathbf{C}} ||\mathbf{A}\mathbf{x} - \mathbf{b}||_2 = ||\mathbf{A}\mathbf{x}^* - \mathbf{b}||_2$.
- For every $\tilde{\mathbf{x}} \in \mathbb{C}^{I_1}$, such that $||\mathbf{A}\mathbf{x}^* - \mathbf{b}||_2 = ||\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}||_2$, and $\tilde{\mathbf{x}} \neq \mathbf{x}^*$, we have

$$||\tilde{\mathbf{x}}||_2 \geq ||\mathbf{x}^*||_2.$$

Proposition 2: Low-rank decomposition of a matrix

A matrix $\mathbf{F} \in \mathbb{C}^{I_1 \times I_2}$ can be written as the sum of rank-1 components,

$$\mathbf{F} = \sum_{i=1}^{\min\{I_1, I_2\}} \sigma_i \mathbf{u}_i \mathbf{v}_i^T.$$

Proposition 3: Truncated SVD

Let $\mathbf{F} \in \mathbb{C}^{I_1 \times I_2}$. $k < r = \text{rank}(\mathbf{F})$, and

$$\mathbf{F}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T,$$

Then

$$\min_{rank(\mathbf{G})=k} ||\mathbf{F} - \mathbf{G}||_2 = ||\mathbf{F} - \mathbf{F}_k||_2 = \sigma_{k+1}.$$

Image Compression with SVD

- Image compression is an application of data compression for digital images.
- It reduces the redundancy of the image contents to store or transfer information optimally.
- It can be done by means of the truncated SVD.
- The image is treated as a matrix of pixels with corresponding color values, and is decomposed into smaller components that retain the essential information about the image.
- Some of the singular values σ are significant while the remaining ones are small and not significant. Compression is achieved by keeping the significant singular values and discarding the smallest ones. This corresponds to keeping only the columns of $\mathbf{U}$ and $\mathbf{V}$ that correspond to the significant singular values (truncation).

Image Compression with Truncated SVD

$\mathbf{A}_{230\times322}$
$u = 74.060$
ε_ρ:

$37,948\mathbf{u}_1\mathbf{v}_1^T$
$c_1 = 553$
$\varepsilon_1 = 27.3$

$+5,050\mathbf{u}_2\mathbf{v}_2^T$
$c_2 = 1,106$
$\varepsilon_2 = 24.3$

$+4,331\mathbf{u}_3\mathbf{v}_3^T$
$c_3 = 1,659$
$\varepsilon_3 = 21.6$

$+3,428\mathbf{u}_4\mathbf{v}_4^T$
$c_4 = 2,212$
$\varepsilon_4 = 20.1$

$+2.916\mathbf{u}_5\mathbf{v}_5^T$
$c_5 = 2,765$
$\varepsilon_5 = 18.8$

$+3,045\mathbf{u}_6\mathbf{v}_6^T$
$c_6 = 3,318$
$\varepsilon_6 = 15.9$

$+2,718\mathbf{u}_7\mathbf{v}_7^T$
$c_7 = 3,871$
$\varepsilon_7 = 14.7$

$+2,532\mathbf{u}_8\mathbf{v}_8^T$
$\varepsilon_8 = 4,424$
$\varepsilon_8 = 13.7$

$+2,417\mathbf{u}_9\mathbf{v}_9^T$
$c_9 = 4,977$
$\varepsilon_9 = 13.0$

$+1,975\mathbf{u}_{10}\mathbf{v}_{10}^T$
$c_{10} = 5,530$
$\varepsilon_{10} = 12.4$

$\ldots + 947\mathbf{u}_{20}\mathbf{v}_{20}^T$
$c_{20} = 11,060$
$\varepsilon_{20} = 7.9$

Image Compression with Truncated 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:	$\varepsilon_{30} = 6.3$	$\varepsilon_{40} = 4.9$	$\varepsilon_{50} = 4.3$	$\varepsilon_{60} = 4.1$	$\varepsilon_{70} = 3.5$
					
$\sigma_{80} = 226$	$\sigma_{90} = 196$	$\sigma_{100} = 177$	$\sigma_{110} = 155$	$\sigma_{120} = 141$	$\sigma_{130} = 122$
$c_{80} = 44,240$	$c_{90} = 49,770$	$c_{100} = 55,300$	$c_{110} = 60,830$	$c_{120} = 66,360$	$c_{130} = 71,890$
$\varepsilon_{80} = 3.1$	$\varepsilon_{90} = 2.8$	$\varepsilon_{100} = 2.5$	$\varepsilon_{110} = 2.2$	$\varepsilon_{120} = 1.9$	$\varepsilon_{130} = 1.6$

Image Compression with Truncated SVD

where

- $u = I_1 I_2$ −number of pixels,
- $c_\rho = \rho(1 + I_1 + I_2)$ which denotes the compressed data volume,
- $\epsilon_\rho = \frac{1}{I_1 I_2} \sum_{i,j}^{I_1,I_2} |\mathbf{F}_{i,j} - \mathbf{F}_{k\,i,j}|$ which denotes the mean absolute reconstruction error per pixel,
- $\mathbf{F}_{k\,i,j}$ is the estimation of $\mathbf{F}_{i,j}$.

Face Recognition

- Human beings are very skillful at recognizing faces even under variation on the facial expression, illumination conditions, viewing angles, etc.
- Develop automatic procedures for face recognition that are robust with respect to varying conditions is a challenging research problem that has been investigated using several different approaches.
- However, principal component analysis (PCA) (i.e., SVD) is a popular technique that could be applied to a collection of face images to form a set of basis features.
- PCA method is best when all pictures are taken under similar conditions. It does not perform well when the images are subject to variations on the environmental factors.

Face Recognition

- PCA often goes by the name "eigenfaces". An eigenface is the name given to a set of eigenvectors when used in the computer vision problem of human face recognition. Any human face can be considered to be a combination of these standard faces.
- The eigenvectors are derived from the covariance matrix of the probability distribution over the high-dimensional vector space of the face images.
- The eigenface approach begins with a search for a low-dimensional representation of face images.
- If the training set consists of M images, PCA can provide a basis set of N images, where $N < M$. The reconstruction error is reduced by increasing the number of eigenfaces.

Face Recognition

- In 1991 M. Turk and A. Pentland showed a way of calculating the eigenvectors of a covariance matrix such that computers of that time could perform the eigendecomposition on a large number of face images.
- Face images usually occupy a high-dimensional space and conventional PCA was intractable on such data sets.
- Turk and Pentland's paper demonstrated ways to extract the eigenvectors based on matrices sized by the number of images rather than by the number of pixels.

How create a set of eigenfaces?

1. Prepare a training set of face images. All images of this training set are stored in a single matrix $\mathbf{G}$, where each column is an image. In other words, $\mathbf{G}$ is a matrix of size $n \times p$, with n images of p pixels, with $p \gg n$,

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

How create a set of eigenfaces?

2. Subtract the mean. The average image $\bar{\mathbf{g}}$ has to be calculated and then subtracted from each original image in $\mathbf{G}$

$$\bar{\mathbf{g}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{g}_i,$$

and

$$\bar{\mathbf{G}} = \begin{pmatrix} \mathbf{g}_1^T - \bar{\mathbf{g}}^T \\ \mathbf{g}_2^T - \bar{\mathbf{g}}^T \\ \vdots \\ \mathbf{g}_n^T - \bar{\mathbf{g}}^T \end{pmatrix}^T.$$

How create a set of eigenfaces?

3. Scatter of a set $\mathbf{G}$ of centered images described by the signal covariance $p \times p$ matrix $\hat{\mathbf{G}} = \frac{1}{p-1}\bar{\mathbf{G}}\bar{\mathbf{G}}^T$.
4. Calculate the eigenvectors $\mathbf{u}_i$ and eigenvalues λ_i of the covariance matrix $\hat{\mathbf{G}}$. Each eigenvector (principlal component) has the same dimensionality as the original image, and thus can itself be seen as an image. The eigenvectors of this covariance matrix are the eigenfaces. They are the directions in which the images differ from the mean image.
5. Choose the principal components. Sort the eigenvalues in descending order and arrange the eigenvectors accordingly. The number of principal components k is determined arbitrarily by setting a threshold ϵ on the total variance. The total variance is given by $v = \lambda_1 + \lambda_2 + \cdots + \lambda_p$.
6. k is the smallest number that satisfies $\frac{\lambda_1 + \cdots + \lambda_k}{v} > \epsilon$.

- These eigenfaces can now be used to represent both existing and new faces. We can project a new (mean-subtracted) image on the eigenfaces and thereby record how that new face differs from the mean face.
- The eigenvalues associated with each eigenface represent how much the images in the training set vary from the mean image in that direction. Information is lost by projecting the image on a subset of the eigenvectors, but losses are minimized by keeping those eigenfaces with the largest eigenvalues.

Subset of MIT-CBCL Face Recognition Database

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

Challenge of Large Data Dimension

- Usually step 4 will be a computationally expensive step. Even a "small" image contains many individual signals, e.g. $200 \times 200 = 40,000$.
- For the MIT-CBCL set of 20 images, the $40,000 \times 40,000$ covariance matrix cannot be processed in a straightforward way by existing PCA softwares, since most of them are computationally infeasible.
- But the practical applicability of eigenfaces stems from the possibility to efficiently compute the eigenvectors of $\bar{\mathbf{G}}$ without ever computing $\bar{\mathbf{G}}$ explicitly, as detailed below.

Turk – Pentland's Tractable PCA of an Image Set

- Consider the eigenvector decomposition of $\hat{\mathbf{G}}$ $\hat{\mathbf{G}}\mathbf{u}_i = \bar{\mathbf{G}}\bar{\mathbf{G}}^T\mathbf{u}_i = \lambda_i\mathbf{u}_i$.
- Since $\bar{\mathbf{G}}\bar{\mathbf{G}}^T$ is a large matrix, take instead the eigenvalue decomposition of $\bar{\mathbf{G}}^T\bar{\mathbf{G}}\mathbf{v}_i = \lambda_i\mathbf{v}_i$.
- Then by pre-multiplying both sides of the equation with $\bar{\mathbf{G}}$, we obtain $\bar{\mathbf{G}}\bar{\mathbf{G}}^T\bar{\mathbf{G}}\mathbf{v}_i = \lambda_i\bar{\mathbf{G}}\mathbf{v}_i$.
- So, if $\mathbf{v}_i$ is an eigenvector of $\bar{\mathbf{G}}^T\bar{\mathbf{G}}$, then $\mathbf{u}_i = \bar{\mathbf{G}}\mathbf{v}_i$ is an eigenvector of $\hat{\mathbf{G}}$.

Notice that if we have a training set of 100 images of 200×200 pixels, the matrix $\bar{\mathbf{G}}^T\bar{\mathbf{G}}$ is a 100×100 matrix, which is much more manageable than the $40,000 \times 40,000$ covariance matrix.

Connection with SVD

- Let the SVD of $\bar{\mathbf{G}}$ be

$$\bar{\mathbf{G}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T.$$

- We have

$$\bar{\mathbf{G}}\bar{\mathbf{G}}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T.$$

- Notice that the first k ($k \leq n$) columns of $\mathbf{U}$ associated with the nonzero singular values are the eigenfaces set.
- So using SVD on data matrix $\bar{\mathbf{G}}$, it is unnecessary to calculate the actual covariance matrix to get eigenfaces.

PCA-based restoration of an image

Goal: Given an image $\mathbf{g}$ of an unknown person, represented by a vector in $\mathbb{R}^p$, determine which of the n persons it represents, or decide that the unknown person is not in the database.

$$\tilde{\mathbf{g}}_{[k]} = \sum_{i=1}^{k} \alpha_i \mathbf{u}_i,$$

where $\alpha_i = \mathbf{g}^T \mathbf{u}_i$.

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

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

Limitations of SVD and PCA

- Matrices **U** and **V** in SVD are not at all sparse, so we need a significant cost for computing and using them.
- SVD and PCA work only with 2D data.

Face Recognition Using HOSVD

- Recently, methods for multilinear analysis of image ensembles have been intensively studied. In particular, the face recognition problem can be modeled using a tensor approach, called *TensorFaces*.
- By letting the modes of the tensor represent a different viewing conditions, e.g., illumination, angle, and/or facial expression, it is possible to improve the precision of the recognition algorithm compared to the PCA method.
- Using the HOSVD leads to efficient dimensionality reduction to reduce complexity and storage costs.

TensorFaces

- Consider n person and assume that each person has been photographed with e different facial expressions. The image of each person are stacked as a vector of size p (pixels). The collection of images is stored as a tensor,

$$\mathcal{G} \in \mathbb{R}^{p \times e \times n}.$$

- $\mathcal{G}$ has 3 different modes, image mode, expression mode and person mode, respectively.
- If, for instance we also had photos of each person with different illumination, viewing angles, etc., then we could represent the image collection by a tensor of higher order.

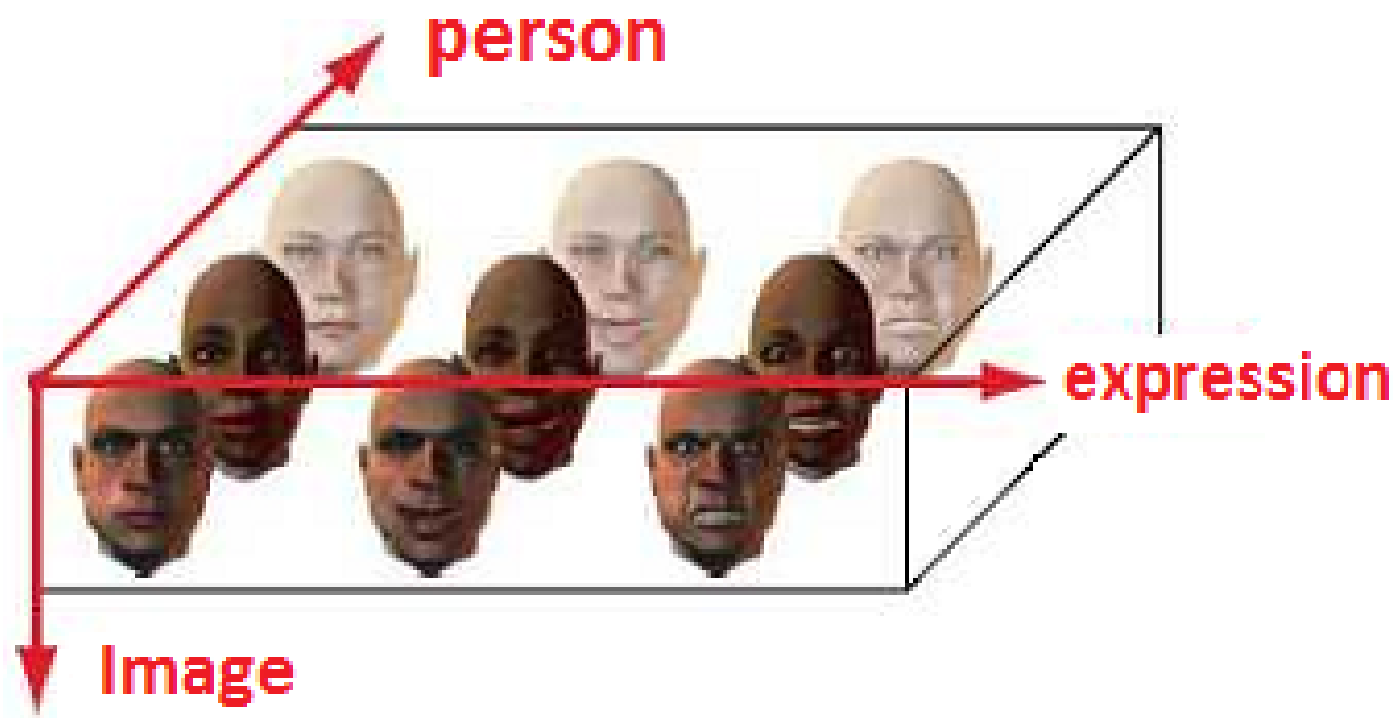


Figure: A 3-order tensor with 3 different modes, image mode, expression mode and person mode.

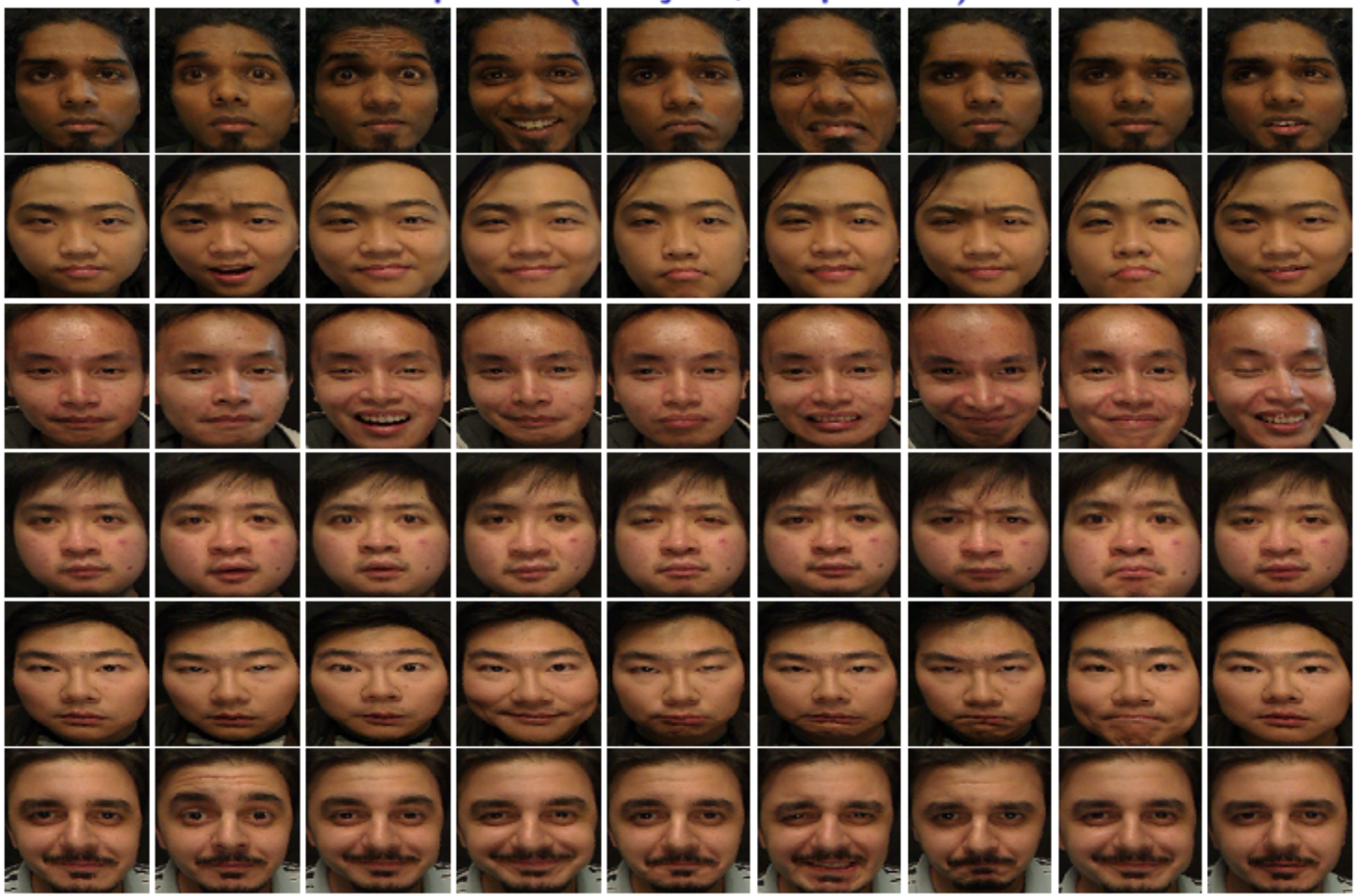
TensorFaces

- Let $p \gg en$ and consider the HOSVD $\mathcal{G}$ as follows

$$\mathcal{G} = \mathcal{S} \times_p \mathbf{U} \times_e \mathbf{V} \times_n \mathbf{W},$$

  where $\mathcal{S} \in \mathbb{R}^{en \times e \times n}$ is the core tensor, $\mathbf{U} \in \mathbb{R}^{p \times en}$ has the orthonormal columns $\mathbf{V} \in \mathbb{R}^{e \times e}$ and $\mathbf{W} \in \mathbb{R}^{n \times n}$ are the orthogonal.
- $\mathcal{G}(:, e_0, n_0) = \mathcal{S} \times_p \mathbf{U} \times_e \mathbf{V}(:, e_0) \times_n \mathbf{W}(:, n_0)$ denotes person n_0 in expression e_0.

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



TensorFaces. Example.

For example person 6 in expression 4 (happy) is uniquely characterized via the following multilinear

$$\mathcal{G}(:,4,6) = \mathcal{S} \times_p \mathbf{U} \times_e \mathbf{V}(:,4) \times_n \mathbf{W}(:,6).$$

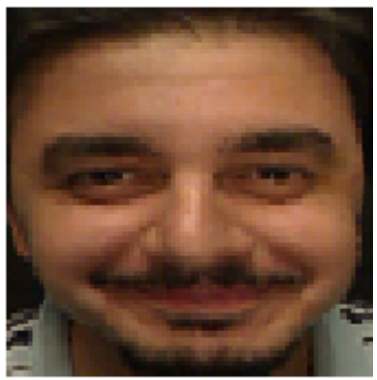


Figure: Person 6 in expression 4 (happy).

Goal: Given an image of an unknown person, represented by a vector in $\mathbb{R}^p$, determine which of the n persons it represents, or decide that the unknown person is not in the database.

How creat a set of TensorFaces

1. For the classification write the HOSVD in the following form

$$\mathcal{G} = \mathcal{C} \times_n \mathbf{W},$$

where $\mathcal{C} = \mathcal{S} \times_p \mathbf{U} \times_e \mathbf{V}$.

2. For a particular expression e

$$\mathcal{G}(:, e, :) = \mathcal{C}(:, e, :) \times_n \mathbf{W}.$$

How creat a set of TensorFaces

3. Since $\mathcal{G}(:, e, :)$ and $\mathcal{C}(:, e, :)$ are matrices, for all the expressions, we have linear relations

$$\mathbf{G}_e = \mathbf{C}_e \mathbf{W}^T.$$

4. So,

$$\mathbf{G}_e(:, n) = \mathbf{C}_e \mathbf{W}(n, :).$$

Interpretation of steps 3 and 4:

Column n of $\mathbf{G}_e$ contains the image of person n in expression e. The columns of $\mathbf{C}_e$ are basis vectors for expression e, and row n of $\mathbf{W}$, i.e., $\mathbf{W}(n, :)$, holds the coordinates of the image of person n in this basis. Furthermore, the same $\mathbf{W}(n, :)$ holds the coordinates of the images of person n in all expression bases.

Restoration of an Image using TensorFaces

- Consider $\mathbf{z} \in \mathbb{R}^p$ is an image of an unknown person in an unknown expression (out of the expressions) and that we want to classify it.
- We refer to $\mathbf{z}$ as a test image.
- Obviously, if it is an image of person n in expression e, then the coordinates of $\mathbf{z}$ in that basis are equal to $\mathbf{W}(n, :)$. Thus we can classify $\mathbf{z}$ by computing its coordinates in all the expression bases and checking, for each expression, whether the coordinates of $\mathbf{z}$ coincide (or almost coincide) with the elements of any row of $\mathbf{W}$.
- The coordinates of $\mathbf{z}$ in expression basis e can be found by solving the least squares problem

$$\min_{\alpha_e} \| \mathbf{C}_e \alpha_e - \mathbf{z} \|_2 .$$

Restoration of an Image using TensorFaces

Algorithm: Classification algorithm

- **Initialization:** $\mathbf{z}$ is a test image.
- **For** $i = 1, 2, \cdots, e$
  - Solve $\min_{\alpha_i} \| \mathbf{C}_i \alpha_i - \mathbf{z} \|_2$.
  - **For** $j = 1, 2, \cdots, n$
    - If $\| \alpha_i - \mathbf{W}(j, :) \|_2 <$tol, then classify as person j and stop.
  - **End**
- **End**