SVD and Applications

Olga Sorkine-Hornung
Dimensionality reduction

\[
\text{data vectors } \mathbf{a}_j \quad \text{(reduced) SVD of } \mathbf{A}
\]

\[
\begin{align*}
\mathbf{A} & \quad \mathbf{U} \quad \mathbf{\Sigma} \quad \mathbf{V}^T \\
& \quad m \times n \quad m \times n \quad n \times n \quad n \times n
\end{align*}
\]
Dimensionality reduction

Data vectors \( \mathbf{a}_j \) (reduced) SVD of \( \mathbf{A} \)

\[ \mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T \]

nullify the last (small) singular values!

\( m \times n \quad m \times n \quad n \times n \quad n \times n \)
Dimensionality reduction

Data vectors $a_j$

$$A \approx U_k \Sigma_k V_k^T$$

$A$: $m \times n$
$U_k$: $m \times k$
$\Sigma_k$: $k \times k$
$V_k^T$: $k \times n$
Dimensionality reduction

- Data vectors $\mathbf{a}_j$
- Basis for the reduced subspace

$$\mathbf{A} \approx \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$$

- $\mathbf{A}$: $m \times n$
- $\mathbf{U}_k$: $m \times k$
- $\mathbf{\Sigma}_k$: $k \times k$
- $\mathbf{V}_k^T$: $k \times n$
Dimensionality reduction

Data vectors $a_j$

$A \approx U_k \Sigma_k V_k^T$

$m \times n$

$U_k$

$m \times k$

$\Sigma_k$

$k \times k$

$V_k^T$

$k \times n$

Basis for the reduced subspace

Coordinates of the projections of the data vectors onto the reduced subspace
Also called PCA, Principal Component Analysis

\[ \begin{align*}
A & \quad m \times n \\
U_k & \quad m \times k \\
\Sigma_k & \quad k \times k \\
V_k^T & \quad k \times n
\end{align*} \]

data vectors \( a_j \)

basis for the reduced subspace

coordinates of the projections of the data vectors onto the reduced subspace
The space of human faces

Concatenate grayscale pixel values into data point $a \in \mathbb{R}^m$
Database of faces

- Many 100x100 pixel face images
- Different people
- Different poses
- Different illumination
- ...
- Each image is labeled with the person’s name/identification

Hypothesis: Faces live in a low-dimensional subspace of $\mathbb{R}^{10,000}$
PCA on face database: Eigenfaces

average face

\[ \mathbf{a}_{\text{avg}} = \frac{1}{n} \sum_{j=1}^{n} \mathbf{a}_j \]
PCA on face database: Eigenfaces

average face

\[ a_{\text{avg}} = \frac{1}{n} \sum_{j=1}^{n} a_j \]

data vectors \( \tilde{a}_j := a_j - a_{\text{avg}} \)
PCA on face database: Eigenfaces

average face

$$a_{avg} = \frac{1}{n} \sum_{j=1}^{n} a_j$$

data vectors $$\tilde{a}_j := a_j - a_{avg}$$

$$A = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}$$

$$m \times n$$
PCA on face database: Eigenfaces

average face

\[ \mathbf{a}_{\text{avg}} = \frac{1}{n} \sum_{j=1}^{n} \mathbf{a}_j \]

data vectors \( \tilde{\mathbf{a}}_j := \mathbf{a}_j - \mathbf{a}_{\text{avg}} \)

\[ \mathbf{A} \approx \begin{bmatrix} \mathbf{U}_k & \mathbf{O}_1 \ldots \mathbf{O}_k \\ \mathbf{0} & \mathbf{V}_k^T \end{bmatrix} \]

\[ \mathbf{A} \in \mathbb{R}^{m \times n}, \quad \mathbf{U}_k \in \mathbb{R}^{m \times k}, \quad \mathbf{O}_i \in \mathbb{R}^{k \times k}, \quad \mathbf{O}_i \text{ orthogonal}, \quad \mathbf{V}_k \in \mathbb{R}^{k \times n} \]
PCA on face database: Eigenfaces

average face
\[
a_{\text{avg}} = \frac{1}{n} \sum_{j=1}^{n} a_j
\]

data vectors \( \tilde{a}_j := a_j - a_{\text{avg}} \)

\[
\begin{align*}
A & \quad m \times n \\
U_k & \quad m \times k \\
\Sigma_k & \quad k \times k \\
V_k^T & \quad k \times n
\end{align*}
\]
PCA on face database: Eigenfaces

average face

$$a_{\text{avg}} = \frac{1}{n} \sum_{j=1}^{n} a_j$$

k=8 first principal components (columns of $U$)
PCA on face database: Eigenfaces

Coordinates of each projection of each $a_j$ onto the subspace of eigenfaces:

$$\Sigma_k V_k^T$$

$k \times n$

Each face is now represented by $k$ numbers!

$k=8$ first principal components (columns of $U$)
Face recognition

input image

downsample to 100x100 = m pixels, subtract $a_{avg}$

$\hat{a}$

project onto subspace, i.e. calculate coordinates w.r.t. $u_j$

$\xi_1 = \hat{a}^T u_1$
$\xi_2 = \hat{a}^T u_2$
$\vdots$
$\xi_8 = \hat{a}^T u_8$
Compare $\xi = (\xi_1, \ldots, \xi_k)^T$ with the coordinate vectors $\xi_j$ of $\tilde{a}_j$ in the database. Find the “nearest neighbor” $j^*$ such that $||\xi - \xi_{j^*}||$ is the smallest.
Face recognition

\[ a \approx a_{\text{avg}} + \xi_1 u_1 + \xi_2 u_2 + \xi_3 u_3 + \xi_4 u_4 + \xi_5 u_5 + \xi_6 u_6 + \xi_7 u_7 + \xi_8 u_8 \]

Compare \( \xi = (\xi_1, \ldots, \xi_k)^T \) with the coordinate vectors \( \xi_j \) of \( \tilde{a}_j \) in the database. Find the “nearest neighbor” \( j^* \) such that \( \|\xi - \xi_{j^*}\| \) is the smallest.

Most likely this face in the database is labeled as “Marc Pollefeys” 😊
Face recognition

- Searching for the nearest neighbor in a $k$-dimensional space is very fast if $k$ is small! And gives quite accurate results.

- Searching for the most similar face in a $m=10,000$-dimensional space is practically infeasible.

- Most $10,000$-dimensional vectors do not represent a human face!

- This kind of dimensionality reduction is the basis of many various machine learning methods.