Principal Component Analysis

Aryan Mokhtari, Santiago Paternain, and Alejandro Ribeiro
Dept. of Electrical and Systems Engineering
University of Pennsylvania
aribeiro@seas.upenn.edu
`http://www.seas.upenn.edu/users/~aribeiro/`

March 28, 2018

The DFT and iDFT with unitary matrices

The discrete Fourier transform with unitary matrices

Stochastic signals

Principal Component Analysis (PCA) transform

Dimensionality reduction

Principal Components

Face recognition

The discrete Fourier transform, again

- It is time to write and understand the DFT in a more abstract way
- Write signal x and complex exponential e_{kN} as vectors $\mathbf{x}$ and $\mathbf{e}_{kN}$

$$\mathbf{x} = \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{pmatrix} \qquad \mathbf{e}_{kN} = \frac{1}{\sqrt{N}} \begin{pmatrix} e^{j2\pi k0/N} \\ e^{j2\pi k1/N} \\ \vdots \\ e^{j2\pi k(N-1)/N} \end{pmatrix}$$

- Use vectors to write the kth DFT component as $(\mathbf{e}_{kN}^H = (\mathbf{e}_{kN}^*)^T)$

$$X(k) = \mathbf{e}_{kN}^H \mathbf{x} = \langle \mathbf{x}, \mathbf{e}_{kN} \rangle = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$$

- kth DFT component $X(k)$ is the product of $\mathbf{x}$ with exponential $\mathbf{e}_{kN}^H$

DFT in matrix notation

- Write DFT $\mathbf{X}$ as a stacked vector and stack individual definitions

$$\mathbf{X} = \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{0N}^H \mathbf{x} \\ \mathbf{e}_{1N}^H \mathbf{x} \\ \vdots \\ \mathbf{e}_{(N-1)N}^H \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{0N}^H \\ \mathbf{e}_{1N}^H \\ \vdots \\ \mathbf{e}_{(N-1)N}^H \end{bmatrix} \mathbf{x}$$

- Define the DFT matrix $\mathbf{F}^H$ so that we can write $\mathbf{X} = \mathbf{F}^H \mathbf{x}$

$$\mathbf{F}^H = \begin{bmatrix} \mathbf{e}_{0N}^H \\ \mathbf{e}_{1N}^H \\ \vdots \\ \mathbf{e}_{(N-1)N}^H \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-j2\pi(1)(1)/N} & \cdots & e^{-j2\pi(1)(N-1)/N} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi(N-1)(1)/N} & \cdots & e^{-j2\pi(N-1)(N-1)/N} \end{bmatrix}$$

- The DFT of signal x is a matrix multiplication $\Rightarrow \mathbf{X} = \mathbf{F}^H \mathbf{x}$

The DFT as a matrix product

- In case you are having trouble visualizing the matrix product

$$e^{-j2\pi(k)(0)/N}x(0) \qquad e^{-j2\pi(k)(n)/N}x(n) \qquad e^{-j2\pi(k)(N-1)/N}x(N-1)$$

$$\begin{bmatrix} x(0) \\ \cdot \\ x(n) \\ \cdot \\ x(N-1) \end{bmatrix} = \mathbf{x}$$

$$\mathbf{F}^H = \begin{bmatrix} e^{-j2\pi(0)(0)/N} & \cdot & e^{-j2\pi(0)(n)/N} & \cdot & e^{-j2\pi(0)(N-1)/N} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ e^{-j2\pi(k)(0)/N} & \cdot & e^{-j2\pi(k)(n)/N} & \cdot & e^{-j2\pi(k)(N-1)/N} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ e^{-j2\pi(N-1)(0)/N} & \cdot & e^{-j2\pi(N-1)(n)/N} & \cdot & e^{-j2\pi(N-1)(N-1)/N} \end{bmatrix} \qquad \begin{bmatrix} X(0) \\ \cdot \\ X(k) \\ \cdot \\ X(N-1) \end{bmatrix} = \mathbf{X} = \mathbf{F}^H\mathbf{x}$$

- The kth DFT component $X(k)$ is the kth row of matrix product $\mathbf{F}^H\mathbf{x}$

Some properties of the DFT matrix $\mathbf{F}^H$

- The (k, n)th element of the matrix $\mathbf{F}^H$ is the complex exponential

$$\left((\mathbf{F}^H)\right)_{kn} = e^{-j2\pi(k)(n)/N} = \left(e^{-j2\pi(k)/N}\right)^{(n)}$$

- Since elements of rows are indexed powers we say $\mathbf{F}^H$ is Vandermonde

- Also observe that since $e^{-j2\pi(k)(n)/N} = e^{-j2\pi(n)(k)/N}$ we have

$$\left((\mathbf{F}^H)\right)_{kn} = e^{-j2\pi(k)(n)/N} = e^{-j2\pi(n)(k)/N} = \left((\mathbf{F}^H)\right)_{nk}$$

- The DFT matrix $\mathbf{F}$ is symmetric $\Rightarrow (\mathbf{F}^H)^T = \mathbf{F}^H$

- Can write $\mathbf{F}^H$ as $\Rightarrow \mathbf{F}^H = (\mathbf{F}^H)^T = \left[\begin{array}{cccc} \mathbf{e}^*_{0N} & \mathbf{e}^*_{1N} & \cdots & \mathbf{e}^*_{(N-1)N} \end{array}\right]$

The Hermitian of the DFT matrix $\mathbf{F}^H$

- Let $\mathbf{F} = \left(\mathbf{F}^H\right)^H$ be conjugate transpose of $\mathbf{F}^H$. We can write $\mathbf{F}$ as

$$\mathbf{F} = \begin{bmatrix} \mathbf{e}_{0N}^T \\ \mathbf{e}_{1N}^T \\ \vdots \\ \mathbf{e}_{(N-1)N}^T \end{bmatrix} \quad \Leftarrow \quad \mathbf{F}^H = \begin{bmatrix} \mathbf{e}_{0N}^* & \mathbf{e}_{1N}^* & \cdots & \mathbf{e}_{(N-1)N}^* \end{bmatrix}$$

- We say that $\mathbf{F}^H$ and $\mathbf{F}$ are Hermitians of each other (that's why $\mathbf{F}^H$)
- The nth row of $\mathbf{F}$ is the nth complex exponential $\mathbf{e}_{nN}^T$
- The kth column of $\mathbf{F}^H$ is the kth conjugate complex exponential $\mathbf{e}_{kN}^*$

The product of $\mathbf{F}$ and its Hermitian $\mathbf{F}^H$

- The product between the DFT matrix $\mathbf{F}$ and its Hermitian $\mathbf{F}^H$ is

$$
\begin{array}{cc}
 & \begin{bmatrix} \mathbf{e}_{0N}^* & \cdots & \mathbf{e}_{kN}^* & \cdots & \mathbf{e}_{(N-1)N}^* \end{bmatrix} \\
\begin{bmatrix} \mathbf{e}_{0N}^T \\ \vdots \\ \mathbf{e}_{kN}^T \\ \vdots \\ \mathbf{e}_{(N-1)N}^T \end{bmatrix} &
\begin{bmatrix}
\mathbf{e}_{0N}^T\mathbf{e}_{0N}^* & \cdots & \mathbf{e}_{0N}^T\mathbf{e}_{kN}^* & \cdots & \mathbf{e}_{0N}^T\mathbf{e}_{(N-1)N}^* \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\mathbf{e}_{kN}^T\mathbf{e}_{0N}^* & \cdots & \mathbf{e}_{kN}^T\mathbf{e}_{kN}^* & \cdots & \mathbf{e}_{kN}^T\mathbf{e}_{(N-1)N}^* \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\mathbf{e}_{(N-1)N}^T\mathbf{e}_{0N}^* & \cdots & \mathbf{e}_{(N-1)N}^T\mathbf{e}_{kN}^* & \cdots & \mathbf{e}_{(N-1)N}^T\mathbf{e}_{(N-1)N}^*
\end{bmatrix} = \mathbf{F}\mathbf{F}^H
\end{array}
$$

- The (n, k) element of product matrix is the inner product $\mathbf{e}_{nN}^T\mathbf{e}_{kN}^*$
- Orthonormality of complex exponentials $\Rightarrow \mathbf{e}_{nN}^T\mathbf{e}_{kN}^* = \delta(n-k)$
  $\Rightarrow$ Only the diagonal elements survive in the matrix product

The matrix $\mathbf{F}$ and its inverse

- The DFT matrix $\mathbf{F}$ and its Hermitian are inverses of each other

$$\mathbf{F}\mathbf{F}^H = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} = \mathbf{I}$$

- Matrices whose inverse is its Hermitian, are called unitary matrices
- Have proved the following fundamental theorem. Orthonormality

Theorem
The DFT matrix $\mathbf{F}$ *is unitary* $\Rightarrow \mathbf{F}\mathbf{F}^H = \mathbf{I} = \mathbf{F}^H\mathbf{F}$

The iDFT in matrix form

- We can retrace methodology to also write the iDFT in matrix form
- No new definitions are needed. Use vectors $\mathbf{e}_{nN}$ and $\mathbf{X}$ to write

$$\tilde{x}(n) = \mathbf{e}_{nN}^T \mathbf{X} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N}$$

- Define stacked vector $\tilde{\mathbf{x}}$ and stack definitions. Use expression for $\mathbf{F}$

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x}(0) \\ \tilde{x}(1) \\ \vdots \\ \tilde{x}(N-1) \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{0N}^T \mathbf{X} \\ \mathbf{e}_{1N}^T \mathbf{X} \\ \vdots \\ \mathbf{e}_{(N-1)N}^T \mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{0N}^T \\ \mathbf{e}_{1N}^T \\ \vdots \\ \mathbf{e}_{(N-1)N}^T \end{bmatrix} \mathbf{X} = \mathbf{F}\mathbf{X}$$

- The iDFT is, as the DFT, just a matrix product $\Rightarrow \tilde{\mathbf{x}} = \mathbf{F}\mathbf{X}$

The iDFT in matrix form

- Again, in case you are having trouble visualizing the matrix product

$$e^{j2\pi(0)(n)/N}X(0) \qquad e^{j2\pi(k)(n)/N}X(k) \qquad e^{j2\pi(N-1)(n)/N}X(N-1)$$

$$\begin{bmatrix} X(0) \\ \vdots \\ X(k) \\ \vdots \\ X(N-1) \end{bmatrix} = \mathbf{X}$$

$$\mathbf{F} = \begin{bmatrix} e^{j2\pi(0)(0)/N} & \cdots & e^{j2\pi(k)(0)/N} & \cdots & e^{j2\pi(N-1)(0)/N} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ e^{j2\pi(0)(n)/N} & \cdots & e^{j2\pi(k)(n)/N} & \cdots & e^{j2\pi(N-1)(n)/N} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ e^{j2\pi(0)(N-1)/N} & \cdots & e^{j2\pi(k)(N-1)/N} & \cdots & e^{j2\pi(N-1)(N-1)/N} \end{bmatrix} \qquad \begin{bmatrix} \tilde{\mathbf{x}}(0) \\ \vdots \\ \tilde{\mathbf{x}}(n) \\ \vdots \\ \tilde{\mathbf{x}}(N-1) \end{bmatrix} = \tilde{\mathbf{x}} = \mathbf{F}\mathbf{X}$$

- Can write the iDFT of $\mathbf{X}$ as the matrix product $\Rightarrow \tilde{\mathbf{x}} = \mathbf{F}\mathbf{X}$

Inverse theorem, like a pro

- When we proved theorems we had monkey steps and one smart step
  $\Rightarrow$ That was orthonormality $\Rightarrow$ matrix $\mathbf{F}$ is unitary $\Rightarrow \mathbf{F}^H\mathbf{F} = \mathbf{I}$

Theorem

The iDFT is, indeed, the inverse of the DFT

Proof.

- Write $\tilde{\mathbf{x}} = \mathbf{F}\mathbf{X}$ and $\mathbf{X} = \mathbf{F}^H\mathbf{x}$ and exploit fact that $\mathbf{F}$ is unitary

$$\tilde{\mathbf{x}} = \mathbf{F}\mathbf{X} = \mathbf{F}\mathbf{F}^H\mathbf{x} = \mathbf{I}\mathbf{x} = \mathbf{x} \qquad \square$$

- Actually, this theorem would be true for any transform pair

$$\mathbf{X} = \mathbf{T}^H\mathbf{x} \qquad \Longleftrightarrow \qquad \tilde{\mathbf{x}} = \mathbf{T}\mathbf{X}$$

- As long as the transform matrix $\mathbf{T}$ is unitary $\Rightarrow \mathbf{T}^H\mathbf{T} = \mathbf{I}$

Theorem
The DFT preserves energy $\Rightarrow \|\mathbf{x}\|^2 = \mathbf{x}^H\mathbf{x} = \mathbf{X}^H\mathbf{X} = \|\mathbf{X}\|^2$

Proof.

- Use iDFT to write $\mathbf{x} = \mathbf{F}\mathbf{X}$ and exploit fact that $\mathbf{F}$ is unitary

$$\|\mathbf{x}\|^2 = \mathbf{x}^H\mathbf{x} = (\mathbf{F}\mathbf{X})^H\mathbf{F}\mathbf{X} = \mathbf{X}^H\mathbf{F}^H\mathbf{F}\mathbf{X} = \mathbf{X}^H\mathbf{X} = \|\mathbf{X}\|^2 \qquad \square$$

- This theorem would also be true for any transform pair

$$\mathbf{X} = \mathbf{T}^H\mathbf{x} \qquad \Longleftrightarrow \qquad \tilde{\mathbf{x}} = \mathbf{T}\mathbf{X}$$

- As long as the transform matrix $\mathbf{T}$ is unitary $\Rightarrow \mathbf{T}^H\mathbf{T} = \mathbf{I}$

The discrete cosine transform

- Are there other useful transforms defined by unitary matrices $\mathbf{T}$?
  $\Rightarrow$ Many. One we have already found is the DCT
- Define the inverse DCT matrix $\mathbf{C}$ to write the iDCT as $\tilde{\mathbf{x}} = \mathbf{C}\mathbf{X}$

$$\mathbf{C} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \sqrt{2}\cos\left[\frac{2\pi(1)((1)+1/2)}{N}\right] & \cdots & \sqrt{2}\cos\left[\frac{2\pi(N-1)((1)+1/2)}{N}\right] \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \sqrt{2}\cos\left[\frac{2\pi(1)((N-1)+1/2)}{N}\right] & \cdots & \sqrt{2}\cos\left[\frac{2\pi(N-1)((N-1)+1/2)}{N}\right] \end{bmatrix}$$

- It is ready to verify that $\mathbf{C}$ is unitary (the cosines are orthonormal)
- From where the inverse and energy conservation theorems follow
  $\Rightarrow$ Proofs hold for all unitary matrices, $\mathbf{C}$ in particular

Designing transformations adapted to signals

- A basic information processing theory can be built for any $\mathbf{T}$
- Then, why do we specifically choose the DFT? Or the DCT?
  - $\Rightarrow$ Oscillations represent different rates of change
  - $\Rightarrow$ Different rates of change represent different aspects of a signal
- Not a panacea, though. E.g., $\mathbf{F}^H$ is independent of the signal
- If we know something about signal, we should use it to build better $\mathbf{T}$
- A way of "knowing something" is a stochastic model of the signal
- PCA: Principal component analysis
  - $\Rightarrow$ Use the eigenvectors of the covariance matrix to build $\mathbf{T}$

Stochastic signals

The discrete Fourier transform with unitary matrices

Stochastic signals

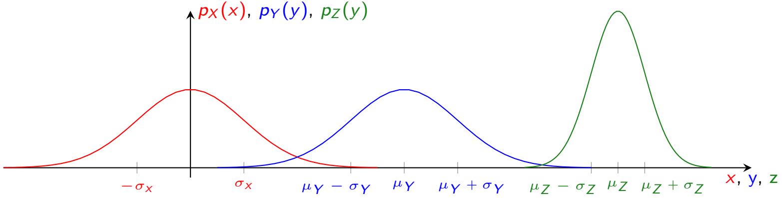
Principal Component Analysis (PCA) transform

Dimensionality reduction

Principal Components

Face recognition

- A random variable X models a random phenomena
  - $\Rightarrow$ One in which many different outcomes are possible
  - $\Rightarrow$ And one in which some outcomes may be more likely than others
- Thus, a random variable represents two things
  - $\Rightarrow$ All possible outcomes and their respective likelihoods



- Random variable X takes values around 0 and Y values around μ_Y
- Z takes values around μ_Z and the values are more concentrated

- Probabilities measure the likelihood of observing different outcomes
  - $\Rightarrow$ Larger probability means an outcome that is more likely
  - $\Rightarrow$ Or, observed more often when seeing many realizations

- Random variables represented by uppercase $\Rightarrow$ E.g., X
- Values that it can take represented by lowercase $\Rightarrow$ E.g., x

- The probability that X takes values between x and x' is written as

$$\mathrm{P}\big(x < X \leq x'\big)$$

- Here, we describe probabilities with density functions (pdf) $\Rightarrow p_X(x)$

$$\mathrm{P}\big(x < X \leq x'\big) = \int_{x}^{x'} p_X(u)\, du$$

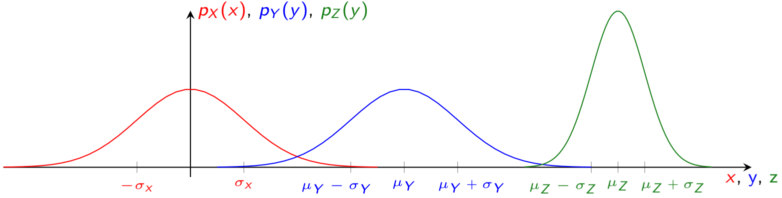
- $p_X(x) \approx$ How likely random variable X is to take a value around x

Gaussian random variables

- A random variable X is Gaussian (or Normal) if its pdf is of the form

$$p_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/\sigma^2}$$

- The mean μ determines center. The variance σ^2 determines width



- Means satisfy $0 = \mu_X < \mu_Y < \mu_Z$. Variances are $\sigma_X^2 = \sigma_Y^2 > \sigma_Z^2$

Expectation

- Expectation of random variable is an average weighted by likelihoods

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x p_X(x)\, dx$$

- Regular average $\Rightarrow$ Sum all values and divide by number of values
- Expectation $\Rightarrow$ Weight values x by their relative likelihoods $p_X(x)$

- For a Gaussian random variable X the expectation is the mean μ

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/\sigma^2}\, dx = \mu$$

- Not difficult to evaluate integral, but besides the point to do so here

Variance

- Measure of variability around the mean weighted by likelihoods

$$\text{var}\left[X\right] = \mathbb{E}\left[\big(X - \mathbb{E}\left[X\right]\big)^2\right] = \int_{-\infty}^{\infty} \big(x - \mathbb{E}\left[X\right]\big)^2 p_X(x)\, dx$$

- Large variance $\equiv$ likely values are spread out around the mean
- Small variance $\equiv$ likely values are concentrated around the mean

- For a Gaussian random variable X the variance is the variance σ^2

$$\text{var}\left[X\right] = \int_{-\infty}^{\infty} \big(x - \mathbb{E}\left[X\right]\big)^2 \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/\sigma^2}\, dx = \sigma^2$$

- Not difficult to evaluate either. But also besides the point here

Random signals

- A random signal $\mathbf{X}$ is a collection of random variables (length N)

$$\mathbf{X} = [X(0),\ X(1),\ \ldots,\ X(N-1)]^T$$

- Each of the random variables has its own pdf $\Rightarrow p_{X(n)}(x)$
- This pdf describes the likelihood of $X(n)$ taking a value around x

- This is not a sufficient description. Joint outcomes also important
- Joint pdf $p_{\mathbf{X}}(\mathbf{x})$ says how likely signal $\mathbf{X}$ is to be found around $\mathbf{x}$

$$\mathrm{P}\left(\mathbf{x} \in \mathcal{X}\right) = \iint_{\mathcal{X}} p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}$$

- The individual pdfs $p_{X(n)}(x)$ are said to be marginal pdfs

Face images

- Random signal $\mathbf{X}$ $\Rightarrow$ All possible images of human faces
- More manageable $\Rightarrow$ $\mathbf{X}$ is a collection of 400 face images
  - $\Rightarrow$ The random variable represents all the images
  - $\Rightarrow$ The likelihood of each of them being chosen. E.g., $1/400$ each

- Random variable specified by all outcomes and respective probabilities

Vectorization

- Do observe that the dataset consists of images $\equiv$ matrices
- Each image is stored in a matrix of size 112×92

$$\mathbf{M}_i = \begin{bmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,92} \\ m_{2,1} & m_{2,2} & \dots & m_{2,92} \\ \vdots & \vdots & \ddots & \vdots \\ m_{112,1} & m_{112,2} & \dots & m_{112,92} \end{bmatrix}$$

- Stack columns of image M_i into the vector $\mathbf{x}_i$ with length $10,304$

$$\mathbf{x}_i = \Big[m_{1,1},\ m_{21},\ \dots,\ m_{112,1},\ m_{1,2},\ m_{2,2},\ \dots,\ m_{112,2},\ \vdots,\ m_{1,92},\ m_{2,92},\ \dots,\ m_{112,92}\Big]^T$$

- Images are matrices $\mathbf{M}_i \in \mathbb{R}^{112\times 92}$. Signals are vectors $\mathbf{x}_i \in \mathbb{R}^{10,304}$

Realizations

- Realization $\mathbf{x}$ is an individual face pulled from set of possible outcomes
- Three possible realizations shown



- Realizations are just regular signals. Nothing random about them

Expectation, variance and covariance

- Signal's expectation is the concatenation of individual expectations

$$\mathbb{E}[\mathbf{X}] = \Big[\mathbb{E}[X(0)],\ \mathbb{E}[X(1)],\ \ldots\ \mathbb{E}[X(N-1)]\Big]^T = \int\!\!\int \mathbf{x} p_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}$$

- Variance of nth element $\Rightarrow \Sigma_{nn} = \text{var}[X(n)] = \mathbb{E}\Big[\big(X(n) - \mathbb{E}[X(n)]\big)^2\Big]$
- Measures variability of nth component

- Covariance between the signal components $X(n)$ and $X(m)$

$$\Sigma_{nm} = \mathbb{E}\Big[\big(X(n) - \mathbb{E}[X(n)]\big)\big(X(m) - \mathbb{E}[X(m)]\big)\Big] = \Sigma_{mn}$$

- Measures how much $X(n)$ predicts $X(m)$. Love, hate, and indifference
  - $\Rightarrow \Sigma_{nm} = 0$, components are unrelated. They are orthogonal
  - $\Rightarrow \Sigma_{nm} > 0$ ($\Sigma_{nm} < 0$), move in same (opposite) direction

Covariance matrix

- Assume that $\mathbb{E}[\mathbf{X}] = \mathbf{0}$ so that covariances are $\Sigma_{nm} = \mathbb{E}[X(n)X(m)]$
- Consider the expectation $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ of the (outer) product $\mathbf{x}\mathbf{x}^T$
- We can write the outer product $\mathbf{x}\mathbf{x}^T$ as

$$\mathbf{x}\mathbf{x}^T = \begin{bmatrix} x(0)x(0) & \cdots & x(0)x(n) & \cdots & x(0)x(N-1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x(n)x(0) & \cdots & x(n)x(n) & \cdots & x(n)x(N-1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x(N-1)x(0) & \cdots & x(N-1)x(n) & \cdots & x(N-1)x(N-1) \end{bmatrix}$$

-

Covariance matrix

- Assume that $\mathbb{E}[\mathbf{X}] = \mathbf{0}$ so that covariances are $\Sigma_{nm} = \mathbb{E}[X(n)X(m)]$
- Consider the expectation $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ of the (outer) product $\mathbf{x}\mathbf{x}^T$
- Expectation $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ implies expectation of each individual element

$$\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] = \begin{bmatrix} \mathbb{E}[x(0)x(0)] & \cdots & \mathbb{E}[x(0)x(n)] & \cdots & \mathbb{E}[x(0)x(N-1)] \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbb{E}[x(n)x(0)] & \cdots & \mathbb{E}[x(n)x(n)] & \cdots & \mathbb{E}[x(n)x(N-1)] \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbb{E}[x(N-1)x(0)] & \cdots & \mathbb{E}[x(N-1)x(n)] & \cdots & \mathbb{E}[x(N-1)x(N-1)] \end{bmatrix}$$

-

Covariance matrix

- Assume that $\mathbb{E}[\mathbf{X}] = \mathbf{0}$ so that covariances are $\Sigma_{nm} = \mathbb{E}[X(n)X(m)]$
- Consider the expectation $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ of the (outer) product $\mathbf{x}\mathbf{x}^T$
- The (n, m) element of the matrix $\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$ is the covariance Σ_{nm}

$$\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] = \begin{bmatrix} \Sigma_{00} & \cdots & \Sigma_{0n} & \cdots & \Sigma_{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{n0} & \cdots & \Sigma_{nn} & \cdots & \Sigma_{n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{(N-1)0} & \cdots & \Sigma_{(N-1)n} & \cdots & \Sigma_{(N-1)(N-1)} \end{bmatrix}$$

- Define the covariance matrix of random signal $\mathbf{X}$ as $\boldsymbol{\Sigma} := \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$

Definition of covariance matrix

- When the mean is not null define the covariance matrix of $\mathbf{X}$ as

$$\boldsymbol{\Sigma} := \mathbb{E}\left[\big(\mathbf{x} - \mathbb{E}[\mathbf{x}]\big)\big(\mathbf{x} - \mathbb{E}[\mathbf{x}]\big)^T\right]$$

- As before, the (n, m) element of $\boldsymbol{\Sigma}$ is the covariance Σ_{nm}

$$((\boldsymbol{\Sigma}))_{nm} = \mathbb{E}\left[\big(X(n) - \mathbb{E}[X(n)]\big)\big(X(m) - \mathbb{E}[X(m)]\big)\right] = \Sigma_{nm}$$

- The covariance matrix $\boldsymbol{\Sigma}$ is an arrangement of the covariances Σ_{nm}

- The diagonal of $\boldsymbol{\Sigma}$ contains the (auto)variances $\Sigma_{nn} = \text{var}\left[X(n)\right]$

- Covariance matrix is symmetric $\Rightarrow ((\boldsymbol{\Sigma}))_{nm} = \Sigma_{nm} = \Sigma_{mn} = ((\boldsymbol{\Sigma}))_{mn}$

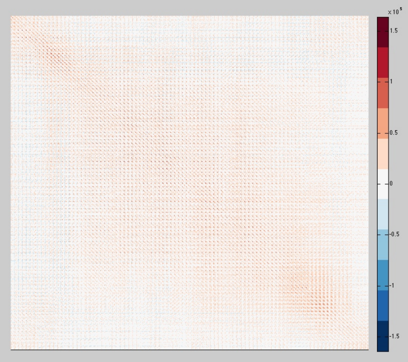
Mean of face images

- All images are equally likely $\Rightarrow$ probability $1/400$ for each image
- The mean face is the regular average $\Rightarrow \mathbb{E}[\mathbf{x}] = \frac{1}{400}\sum_{i=1}^{400}\mathbf{x}_i$

- Average image looks something, sort of, like an average face

Covariance matrix of face images

- Covariance matrix $\Rightarrow \boldsymbol{\Sigma} = \dfrac{1}{400}\sum_{i=1}^{400}\Big(\mathbf{x}_i - \mathbb{E}\left[\mathbf{x}\right]\Big)\Big(\mathbf{x}_i - \mathbb{E}\left[\mathbf{x}\right]\Big)^T$



- Heat map of covariance matrix $\boldsymbol{\Sigma}$ shown on left
- Large correlation values around diagonal
- Large correlation values every 112 elements (jump a row on matrix)

Principal Component Analysis (PCA) transform

The discrete Fourier transform with unitary matrices

Stochastic signals

Principal Component Analysis (PCA) transform

Dimensionality reduction

Principal Components

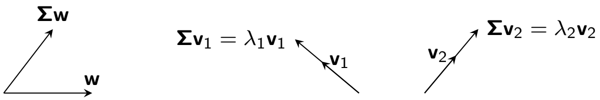
Face recognition

Eigenvectors and eigenvalues of covariance matrix

- Consider a vector with N elements $\Rightarrow \mathbf{v} = [v(0), v(1), \ldots, v(N-1)]$
- We say that $\mathbf{v}$ is an eigenvector of $\mathbf{\Sigma}$ if for some scalar $\lambda \in \mathbb{R}$

$$\mathbf{\Sigma v} = \lambda \mathbf{v}$$

- We say that λ is the eigenvalue associated to $\mathbf{v}$



- In general, non-eigenvectors $\mathbf{w}$ and $\mathbf{\Sigma w}$ point in different directions
- But for eigenvectors $\mathbf{v}$, the product vector $\mathbf{\Sigma v}$ is collinear with $\mathbf{v}$

Normalization

- If $\mathbf{v}$ is an eigenvector, $\alpha\mathbf{v}$ is also an eigenvector for any scalar $\alpha \in \mathbb{R}$,

$$\mathbf{\Sigma}(\alpha\mathbf{v}) = \alpha(\mathbf{\Sigma}\mathbf{v}) = \alpha\lambda\mathbf{v} = \lambda(\alpha\mathbf{v})$$

- Eigenvectors are defined up to a constant

- We use normalized eigenvectors with unit energy $\Rightarrow \|\mathbf{v}\|^2 = 1$
- If we compute $\mathbf{v}$ with $\|\mathbf{v}\|^2 \neq 1$ replace $\mathbf{v}$ with $\mathbf{v}/\|\mathbf{v}\|$

- There are N eigenvalues and distinct associated eigenvectors

  $\Rightarrow$ Some technical qualifications are needed in this statement

Ordering

Theorem

The eigenvalues of $\mathbf{\Sigma}$ *are real and nonnegative* $\Rightarrow \lambda \in \mathbb{R}$ *and* $\lambda \geq 0$

Proof.

- Begin by observing that we can write $\lambda = \mathbf{v}^H \mathbf{\Sigma} \mathbf{v} / \|\mathbf{v}\|^2$. Indeed

$$\mathbf{v}^H \mathbf{\Sigma} \mathbf{v} = \mathbf{v}^H (\mathbf{\Sigma} \mathbf{v}) = \mathbf{v}^H (\lambda \mathbf{v}) = \lambda \mathbf{v}^H \mathbf{v} = \lambda \|\mathbf{v}\|^2$$

- Complete by showing that $\mathbf{v}^T \mathbf{\Sigma} \mathbf{v}$ is nonnegative. Indeed (assume $\mathbb{E}[\mathbf{x}] = \mathbf{0}$)

$$\mathbf{v}^H \mathbf{\Sigma} \mathbf{v} = \mathbf{v}^H \mathbb{E}\left[\mathbf{x}\mathbf{x}^H\right] \mathbf{v} = \mathbb{E}\left[\mathbf{v}^H \mathbf{x}\mathbf{x}^H \mathbf{v}\right] = \mathbb{E}\left[\left(\mathbf{v}^H \mathbf{x}\right)\left(\mathbf{x}^H \mathbf{v}\right)\right] = \mathbb{E}\left[\left(\mathbf{v}^H \mathbf{x}\right)^2\right] \geq 0$$

□

- Order eigenvalues from largest to smallest $\Rightarrow \lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_{N-1}$
- Eigenvectors inherit order $\Rightarrow \mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{N-1}$
- The nth eigenvector of $\mathbf{\Sigma}$ is associated with its nth largest eigenvalue

Eigenvectors are orthonormal

Theorem

Eigenvectors of $\boldsymbol{\Sigma}$ associated with different eigenvalues are orthogonal

Proof.

- Normalized eigenvectors $\mathbf{v}$ and $\mathbf{u}$ associated with eigenvalues $\lambda \neq \mu$

$$\boldsymbol{\Sigma}\mathbf{v} = \lambda\mathbf{v}, \qquad \boldsymbol{\Sigma}\mathbf{u} = \mu\mathbf{u}$$

- Since the matrix $\boldsymbol{\Sigma}$ is symmetric we have $\boldsymbol{\Sigma}^H = \boldsymbol{\Sigma}$, and it follows

$$\mathbf{u}^H\Sigma\mathbf{v} = \left(\mathbf{u}^H\Sigma\mathbf{v}\right)^H = \mathbf{v}^H\Sigma^H\mathbf{u} = \mathbf{v}^H\Sigma\mathbf{u}$$

- Make $\Sigma\mathbf{v} = \lambda\mathbf{v}$ on the leftmost side and $\Sigma\mathbf{u} = \mu\mathbf{u}$ on the rightmost

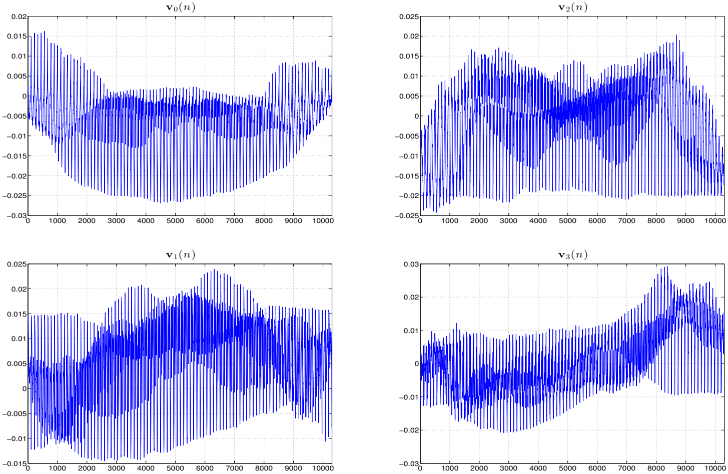
$$\mathbf{u}^H\lambda\mathbf{v} = \lambda\mathbf{u}^H\mathbf{v} = \mu\mathbf{v}^H\mathbf{u} = \mathbf{v}^H\mu\mathbf{u}$$

- Eigenvalues are different $\Rightarrow$ Relationship can only be true if $\mathbf{v}^H\mathbf{u} = 0$

$\square$

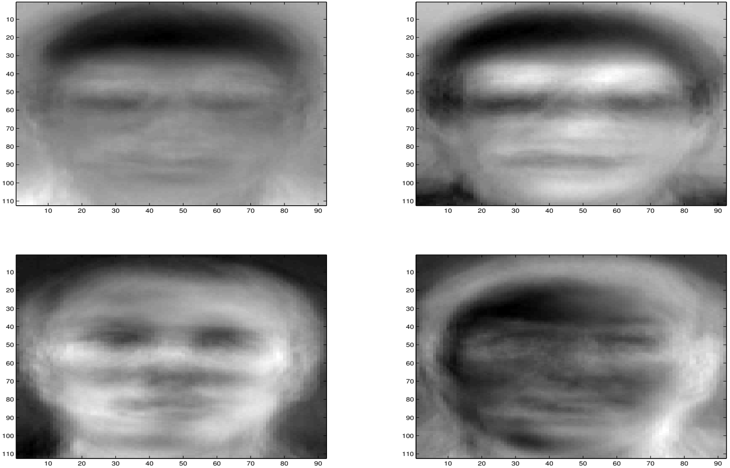
Eigenvectors of face images (1D)

- One dimensional representation of first four eigenvectors $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$



Eigenvectors of face images (2D)

- Two dimensional representation of first four eigenvectors $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$



Eigenvector matrix

- Define the matrix $\mathbf{T}$ whose kth column is the kth eigenvector of $\boldsymbol{\Sigma}$

$$\mathbf{T} = [\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{N-1}]$$

- Since the eigenvectors $\mathbf{v}_k$ are orthonormal, the product $\mathbf{T}^H\mathbf{T}$ is

$$\begin{bmatrix} \mathbf{v}_0 & \cdots & \mathbf{v}_k & \cdots & \mathbf{v}_{N-1} \end{bmatrix}$$

$$\mathbf{T}^H\mathbf{T} = \begin{bmatrix} \mathbf{v}_0^H \\ \vdots \\ \mathbf{v}_k^H \\ \vdots \\ \mathbf{v}_{N-1}^H \end{bmatrix} \begin{bmatrix} \mathbf{v}_0^H\mathbf{v}_0 & \cdots & \mathbf{v}_1^H\mathbf{v}_k & \cdots & \mathbf{v}_0^H\mathbf{v}_{N-1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{v}_k^H\mathbf{v}_0 & \cdots & \mathbf{v}_k^H\mathbf{v}_k & \cdots & \mathbf{v}_k^H\mathbf{v}_{N-1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{v}_{N-1}^H\mathbf{v}_{N-1} & \cdots & \mathbf{v}_{N-1}^H\mathbf{v}_k & \cdots & \mathbf{v}_{N-1}^H\mathbf{v}_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

- The eigenvector matrix $\mathbf{T}$ is Hermitian $\Rightarrow \mathbf{T}^H\mathbf{T} = \mathbf{I}$

Principal component analysis transform

- Any Hermitian $\mathbf{T}$ can be used to define an info processing transform
- Define principal component analysis (PCA) transform $\Rightarrow \mathbf{y} = \mathbf{T}^H\mathbf{x}$
- And the inverse (i)PCA transform $\Rightarrow \tilde{\mathbf{x}} = \mathbf{T}\mathbf{y}$
- Since $\mathbf{T}$ is Hermitian, iPCA is, indeed, the inverse of the PCA

$$\tilde{\mathbf{x}} = \mathbf{T}\mathbf{y} = \mathbf{T}\left(\mathbf{T}^H\mathbf{x}\right) = \mathbf{T}\mathbf{T}^H\mathbf{x} = \mathbf{I}\mathbf{x} = \mathbf{x}$$

- Thus $\mathbf{y}$ is an equivalent representation of $\mathbf{x}$ $\Rightarrow$ Back and forth
- And, also because $\mathbf{T}$ is Hermitian, Parseval's theorem holds

$$\|\mathbf{x}\|^2 = \mathbf{x}^H\mathbf{x} = (\mathbf{T}\mathbf{y})^H\,\mathbf{T}\mathbf{y} = \mathbf{y}^H\mathbf{T}^H\mathbf{T}\mathbf{y} = \mathbf{y}^H\mathbf{y} = \|\mathbf{y}\|^2$$

- Modifying elements y_k means altering energy composition of signal

Discussions

- The PCA transform is defined for any signal (vector) $\mathbf{x}$

  $\Rightarrow$ But we expect to work well only when $\mathbf{x}$ is a realization of $\mathbf{X}$

- Write the iPCA in expanded form and compare with the iDFT

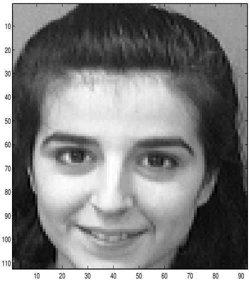
$$x(n) = \sum_{k=0}^{N-1} y(k) v_k(n) \quad \Leftrightarrow \quad x(n) = \sum_{k=0}^{N-1} X(k) e_{kN}(n)$$

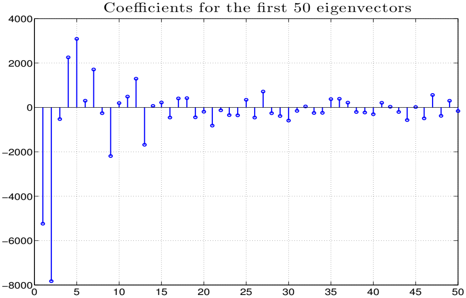
- The same except that they use different bases for the expansion
- Still, like developing a new sense.
- But not one that is generic. Rather, adapted to the random signal $\mathbf{X}$

Coefficients of a projected face image

- PCA transform coefficients for given face image with 10,304 pixels
- Substantial energy in the first 15 PCA coefficients $y(k)$ with $k \leq 15$
- Almost all energy in the first 50 PCA coefficients $y(k)$ with $k \leq 50$

  $\Rightarrow$ This is a compression factor of more than 200





Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - ⇒ Increasing number of coefficients increases accuracy.
  - ⇒ Using 50 coefficients suffices

Figure: image

Figure: No. P.C.s $= 1$

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - $\Rightarrow$ Increasing number of coefficients increases accuracy.
  - $\Rightarrow$ Using 50 coefficients suffices

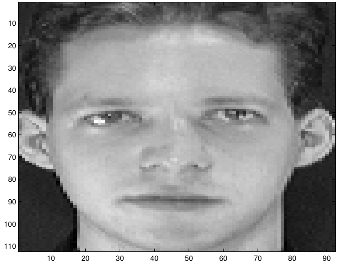


Figure: image

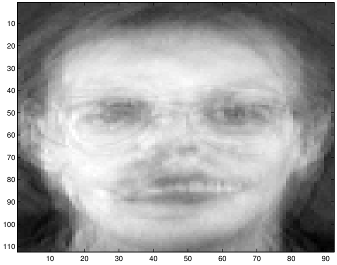


Figure: No. P.C.s $= 5$

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - $\Rightarrow$ Increasing number of coefficients increases accuracy.
  - $\Rightarrow$ Using 50 coefficients suffices

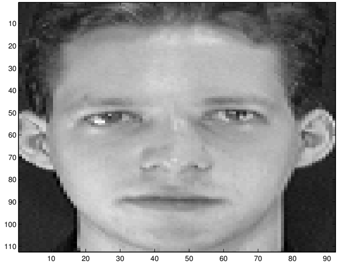


Figure: image

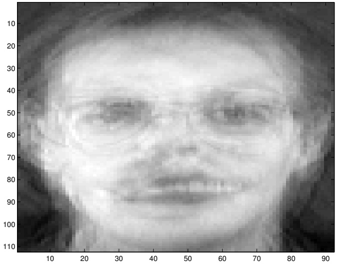


Figure: No. P.C.s $= 10$

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - $\Rightarrow$ Increasing number of coefficients increases accuracy.
  - $\Rightarrow$ Using 50 coefficients suffices

Figure: image

Figure: No. P.C.s = 20

Reconstructed face images

- Reconstructed image for increasing number of PCA coefficients
  - $\Rightarrow$ Increasing number of coefficients increases accuracy.
  - $\Rightarrow$ Using 50 coefficients suffices

Figure: image

Figure: No. P.C.s $= 30$

- Reconstructed image for increasing number of PCA coefficients
  - $\Rightarrow$ Increasing number of coefficients increases accuracy.
  - $\Rightarrow$ Using 50 coefficients suffices

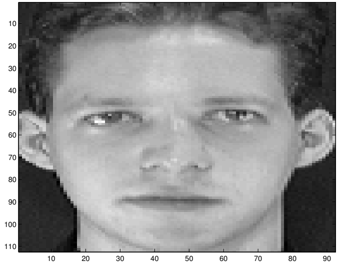


Figure: image

Figure: No. P.C.s $= 40$

Reconstructed face images

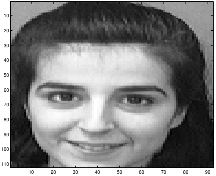
- Reconstructed image for increasing number of PCA coefficients
  - $\Rightarrow$ Increasing number of coefficients increases accuracy.
  - $\Rightarrow$ Using 50 coefficients suffices

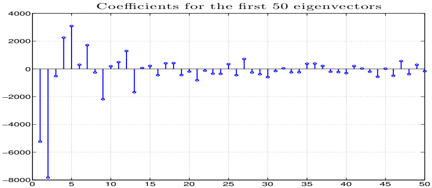
Figure: image

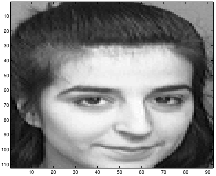
Figure: No. P.C.s $= 50$

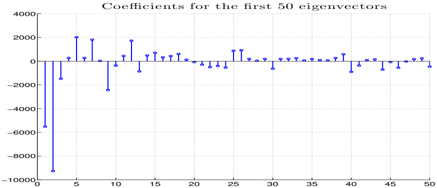
Coefficients of the same person

- PCA transform y for two different pictures of the same person
- Coefficients are similar, even if pose and attitude are different
  - $\Rightarrow$ E.g., first two coefficients almost identical



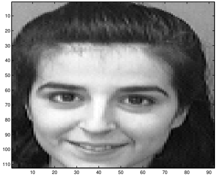


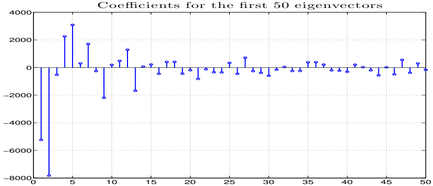


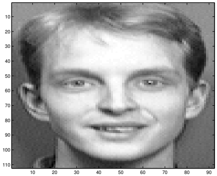


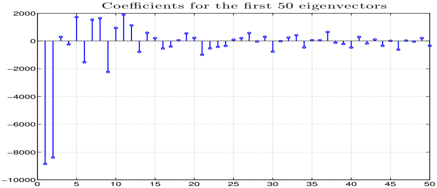
Coefficients of different persons

- PCA transform y for pictures of different persons
- Similar pose and attitude, but PCA coefficients are still different
  - $\Rightarrow$ Can be used to perform face recognition. More later









Dimensionality reduction

Compression with the DFT

- Transform signal $\mathbf{x}$ into frequency domain with DFT $\mathbf{X} = \mathbf{F}^H\mathbf{x}$
- Recover $\mathbf{x}$ from $\mathbf{X}$ through iDFT matrix multiplication $\mathbf{x} = \mathbf{F}\mathbf{X}$
- We compress by retaining $K < N$ DFT coefficients to write

$$\tilde{\mathbf{x}}(n) = \sum_{k=0}^{K-1} X(k) e^{j2\pi kn/N}$$

- Equivalently, we define the compressed DFT as

$$\tilde{\mathbf{X}}(k) = X(k) \quad \text{for} \quad k < K, \qquad \tilde{\mathbf{X}}(k) = 0 \text{ otherwise}$$

- Reconstructed signal is obtained with iDFT $\Rightarrow \tilde{\mathbf{x}} = \mathbf{F}\tilde{\mathbf{X}}$

Compression with the PCA

- Transform signal $\mathbf{x}$ into eigenvector domain with PCA $\mathbf{y} = \mathbf{T}^H \mathbf{x}$
- Recover $\mathbf{x}$ from $\mathbf{y}$ through iPCA matrix multiplication $\mathbf{x} = \mathbf{T}\mathbf{y}$

- We compress by retaining $K < N$ PCA coefficients to write

$$\tilde{\mathbf{x}}(n) = \sum_{k=0}^{K-1} y(k)\mathbf{v}_k(n)$$

- Equivalently, we define the compressed PCA as

$$\tilde{\mathbf{y}}(k) = y(k) \quad \text{for} \quad k < K, \qquad \tilde{\mathbf{y}}(k) = 0 \text{ otherwise}$$

- Reconstructed signal is obtained with iPCA $\Rightarrow \tilde{\mathbf{x}} = \mathbf{T}\tilde{\mathbf{y}}$

Why keeping the first K coefficients?

- Why do we keep the first K DFT coefficients?
  - $\Rightarrow$ Because faster oscillations tend to represent faster variation
  - $\Rightarrow$ Also, not always, sometimes we keep the largest coefficients
- Why do we keep the first K PCA coefficients?
  - $\Rightarrow$ Eigenvectors with lower ordinality have larger eigenvalues
  - $\Rightarrow$ Larger eigenvalues entail more variability
  - $\Rightarrow$ And more variability signifies more dominant features
- Eigenvectors with large ordinality represent finer signal features
  - $\Rightarrow$ And can often be omitted

Dimensionality reduction

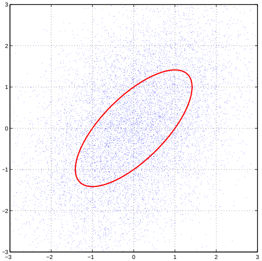
- PCA compression is (more accurately) called dimensionality reduction
  - $\Rightarrow$ Do not compress signal. Reduce number of dimensions

$$\mathbf{\Sigma} = \begin{bmatrix} 3/2 & 1/2 \\ 1/2 & 3/2 \end{bmatrix}$$

- Covariance eigenvectors mix coordinates

$$\mathbf{v}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

- Eigenvalues are $\lambda_0 = 2$ and $\lambda_1 = 1$



- Signal varies more in $\mathbf{v}_0 = [1, 1]^T$ direction than in $\mathbf{v}_1 = [1, -1]^T$
  - $\Rightarrow$ Study one dimensional signal $\tilde{\mathbf{x}} = y(0)\mathbf{v}_0$
  - $\Rightarrow$ instead of the original two dimensional signal $\mathbf{x}$

Expected reconstruction error

- PCA dimensionality reduction minimizes the expected error energy
- To see that this is true, define the error signal as $\Rightarrow \mathbf{e} := \mathbf{x} - \tilde{\mathbf{x}}$
- The energy of the error signal is $\Rightarrow \|\mathbf{e}\|^2 = \|\mathbf{x} - \tilde{\mathbf{x}}\|^2$
- The expected value of the energy of the error signal is

$$\mathbb{E}\left[\|\mathbf{e}\|^2\right] = \mathbb{E}\left[\|\mathbf{x} - \tilde{\mathbf{x}}\|^2\right]$$

- Keeping the first K PCA coefficients minimizes $\mathbb{E}\left[\|\mathbf{e}\|^2\right]$

  $\Rightarrow$ Among all reconstructions that use, at most, K coefficients

Dimensionality reduction expected error

Theorem

The expectation of the reconstruction error is the sum of the eigenvalues corresponding to the eigenvectors of the coefficients that are discarded

$$\mathbb{E}\left[\|\mathbf{e}\|^2\right] = \sum_{k=K}^{N-1} \lambda_k$$

- It follows that keeping the first K PCA coefficients is optimal
  - $\Rightarrow$ In the sense that it minimizes the Expected error energy
- Good on average. Across realizations of the stochastic signal $\mathbf{X}$
- Need not be good for given realization (but we expect it to be good)

Proof of expected error expression

Proof.

- Error signal signal is $\mathbf{e} := \mathbf{x} - \tilde{\mathbf{x}}$. Define error PCA transform as $\mathbf{f} = \mathbf{T}^H \mathbf{x}$
- Using Parseval's (energy conservation) we can write the energy of $\mathbf{e}$ as

$$\|\mathbf{e}\|^2 = \|\mathbf{f}\|^2 = \sum_{k=K}^{N-1} y^2(k)$$

- In the last equality we used that $\mathbf{f} = \mathbf{y} - \tilde{\mathbf{y}} = [0, \ldots, 0, y(K), \ldots, y(N-1)]$
- Here, we are interested in the expected value of the error's energy
- Take expectation on both sides of equality $\Rightarrow \mathbb{E}\left[\|\mathbf{e}\|^2\right] = \sum_{k=K}^{N-1} \mathbb{E}\left[y^2(k)\right]$
- Used the fact that expectations are linear operators

Proof of expected error expression

Proof.

- Compute expected value $\mathbb{E}\left[y^2(k)\right]$ of the squared PCA coefficient $y(k)$
- As per PCA transform definition $y(k) = \mathbf{v}^H\mathbf{x}$, which implies

$$\mathbb{E}\left[y^2(k)\right] = \mathbb{E}\left[(\mathbf{v}_k^H\mathbf{x})^2\right] = \mathbb{E}\left[\mathbf{v}_k^H\mathbf{x}\mathbf{x}^T\mathbf{v}_k\right] = \mathbf{v}_k^H\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]\mathbf{v}_k$$

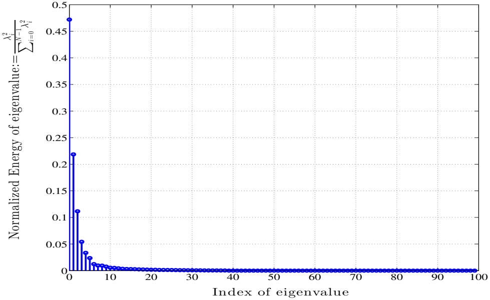
- Covariance matrix: $\mathbf{\Sigma} := \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]$. Eigenvector definition $\mathbf{\Sigma}\mathbf{v}_k = \lambda_k$. Thus

$$\mathbb{E}\left[y^2(k)\right] = \mathbf{v}_k^H\mathbf{\Sigma}\mathbf{v}_k = \mathbf{v}_k^H\lambda_k\mathbf{v}_k = \lambda_k\left\|\mathbf{v}_k\right\|^2$$

- Substitute into expression for $\mathbb{E}\left[\|\mathbf{e}\|^2\right]$ to write $\Rightarrow \mathbb{E}\left[\|\mathbf{e}\|^2\right] = \sum_{k=K}^{N-1}\lambda_k$ $\square$

Principal eigenvalues for face dataset

- Covariance matrix eigenvalues for faces dataset.
- Expected approximation error $\Rightarrow$ Tail sum of eigenvalue distribution
  $\Rightarrow$ Average across all realizations. Not the same as actual error



- First 10 coefficients have 98% of energy.
- Eigenvectors with index $k > 50$ have 10^{-3}% of energy on average

Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")

- Keep 1 coefficient $\Rightarrow$ Reconstruction error $\Rightarrow$ 0.06

  $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0.52

Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")
- Keep 5 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow$ 0.03
  $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0.11

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")

- Keep 10 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow$ 0.02
  $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0.04

Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")
- Keep 20 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow$ 0.01

  $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0.01

Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")

- Keep 30 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow$ 0.006
  $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow$ 0.003

Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")

- Keep 40 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow 0$

  $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow 0$

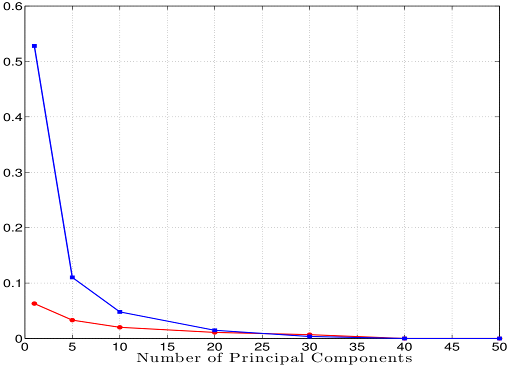
Reconstructed face images

- Increasing number of coefficients reduces reconstruction error
- Average and actual reconstruction not the same (although "close")
- Keep 50 coefficients $\Rightarrow$ Reconstruction error $\Rightarrow 0$

  $\Rightarrow$ Sum of removed eigenvalues $\Rightarrow 0$

Evolution of reconstruction error

- Error for reconstruction process
- one realization (red), energy of removed eigenvalues (blue)



Principal Components

The discrete Fourier transform with unitary matrices

Stochastic signals

Principal Component Analysis (PCA) transform

Dimensionality reduction

Principal Components

Face recognition

Signals with uncorrelated components

- A random signal X with uncorrelated components is one with

$$\Sigma_{nm} = \mathbb{E}\left[\big(X(n) - \mathbb{E}\left[X(n)\right]\big)\big(X(m) - \mathbb{E}\left[X(m)\right]\big)\right] = 0$$

- Different components are unrelated to each other.
- They represent different (orthogonal) aspects of signal
- Components uncorrelated $\Rightarrow$ The covariance matrix is diagonal

$$\mathbf{\Sigma} = \mathbb{E}\left[\big(\mathbf{x} - \mathbb{E}\left[\mathbf{x}\right]\big)\big(\mathbf{x} - \mathbb{E}\left[\mathbf{x}\right]\big)^T\right] = \begin{bmatrix} \Sigma_{00} & \cdots & \Sigma_{0n} & \cdots & \Sigma_{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{n0} & \cdots & \Sigma_{nn} & \cdots & \Sigma_{n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{(N-1)0} & \cdots & \Sigma_{(N-1)n} & \cdots & \Sigma_{(N-1)(N-1)} \end{bmatrix}$$

- How do eigenvectors (principal components) of uncorrelated signals look?

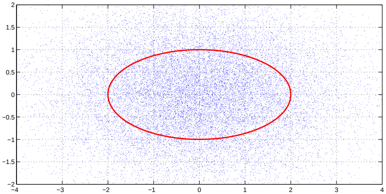
Uncorrelated signal with 2 components

- Signal $\mathbf{X} = [X(0), X(1)]^T$ with 2 components and diagonal covariance

$$\mathbf{\Sigma} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

- Covariance eigenvectors are

$$\mathbf{v}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \mathbf{v}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



- The respective associated eigenvalues are $\lambda_0 = 2$ and $\lambda_1 = 1$
- Eigenvectors are orthogonal, as they should.
  - $\Rightarrow$ Represent directions of separate signal variability
  - $\Rightarrow$ Rate of variability given by associated eigenvalue

Another uncorrelated signal with 2 components

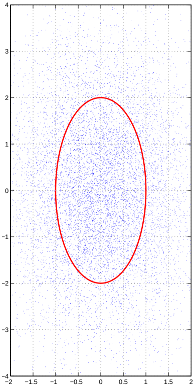
- Signal $\mathbf{X} = [X(0), X(1)]^T$ with 2 components and diagonal covariance

$$\mathbf{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

- Covariance eigenvectors reverse order

$$\mathbf{v}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- Associated eigenvalues are $\lambda_0 = 2$ and $\lambda_1 = 1$
- Eigenvectors still orthogonal, as they should.
  - $\Rightarrow$ Directions of separate signal variability
  - $\Rightarrow$ Rate given by associated eigenvalue



Signal with correlated components

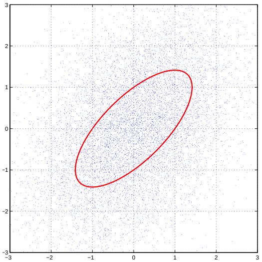
- Signal $\mathbf{X} = [X(0), X(1)]^T$ with 2 components and diagonal covariance

$$\mathbf{\Sigma} = \begin{bmatrix} 3/2 & 1/2 \\ 1/2 & 3/2 \end{bmatrix}$$

- Covariance eigenvectors mix coordinates

$$\mathbf{v}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

- Eigenvalues are $\lambda_0 = 2$ and $\lambda_1 = 1$



- The eigenvalues are orthogonal. This is true for any covariance matrix
  - $\Rightarrow$ Mix coordinates but still represent directions of separate variability
  - $\Rightarrow$ Rate of change also given by associated eigenvalue

Eigenvectors in uncorrelated signals

- Uncorrelated components means diagonal covariance matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \Sigma_{00} & \cdots & \Sigma_{0n} & \cdots & \Sigma_{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{n0} & \cdots & \Sigma_{nn} & \cdots & \Sigma_{n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{(N-1)0} & \cdots & \Sigma_{(N-1)n} & \cdots & \Sigma_{(N-1)(N-1)} \end{bmatrix}$$

- If variances are ordered, kth eigenvector is k-shifted delta $\delta(n-k)$
- The corresponding variance Σ_{kk} is the associated eigenvalue
- Eigenvectors represent directions of orthogonal variability
- Rate of variability given by associated eigenvalue

Eigenvectors in correlated signals

- Correlated components means a full covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} \Sigma_{00} & \cdots & \Sigma_{0n} & \cdots & \Sigma_{0(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{n0} & \cdots & \Sigma_{nn} & \cdots & \Sigma_{n(N-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{(N-1)0} & \cdots & \Sigma_{(N-1)n} & \cdots & \Sigma_{(N-1)(N-1)} \end{bmatrix}$$

- The eigenvectors $\mathbf{v}_k$ now mix different components
  - $\Rightarrow$ But they still represent directions of orthogonal variability
  - $\Rightarrow$ With the rate of variability given by associated eigenvalue
- PCA transform represents a signal as a sum of orthonormal vectors
  - $\Rightarrow$ Each of which represents independent variability
- Principal components (eigenvectors) with larger eigenvalues represent directions in which the signal has more variability

Face recognization

The discrete Fourier transform with unitary matrices

Stochastic signals

Principal Component Analysis (PCA) transform

Dimensionality reduction

Principal Components

Face recognition

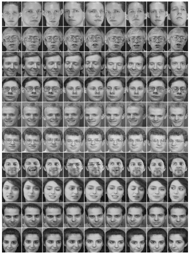
Face Recognition

- Observe faces of known people $\Rightarrow$ Use them to train classifier
- Observe a face of unknown character $\Rightarrow$ Compare and classify

- The dataset we've used contains 10 different images of 40 people

Training set

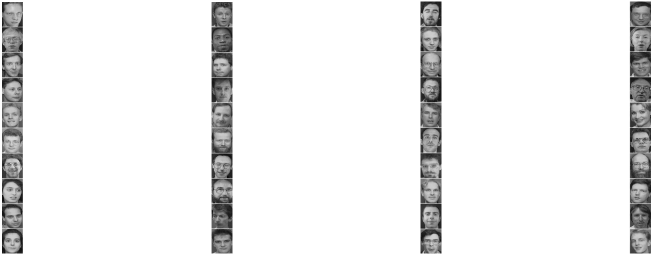
- Separate the first 9 of each person to construct training set




- Interpret these images as know, and use them to train classifier

Test set

- Utilize the last image of each person to construct a test set



- Interpret these images as unknown, and use them to test classifier

Nearest neighbor classification

- Training set contains (signal, label) pairs $\Rightarrow \mathcal{T} = \{(\mathbf{x}_i, z_i)\}_{i=1}^{N}$
- Signal $\mathbf{x}$ is the face image. Label z is the person's "name"
- Given (unknown) signals $\mathbf{x}$, we want to assign a label
- Nearest neighbor classification rule

  $\Rightarrow$ Find nearest neighbor signal in the training set

$$\mathbf{x}_{\text{NN}} := \underset{\mathbf{x}_i \in \mathcal{T}}{\operatorname{argmin}} \|\mathbf{x}_i - \mathbf{x}\|^2$$

  $\Rightarrow$ Assign the label associated with the nearest neighbor

$$\mathbf{x}_{\text{NN}} \quad \Rightarrow \quad (\mathbf{x}_i, z_i) \quad \Rightarrow \quad z = z_i$$

- Reasonable enough. It should work. But it doesn't

The signal and the noise

- Image has a part that is inherent to the person $\Rightarrow$ The actual signal
- But it also contains variability $\Rightarrow$ Which we model as noise

$$\mathbf{x}_i = \tilde{\mathbf{x}}_i + \mathbf{w}$$

- Problem is, there is more variability (noise) than signal

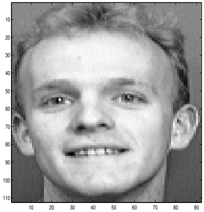


Figure: Test image

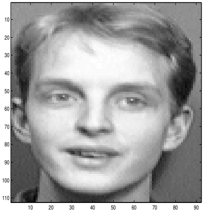


Figure: Nearest neighbor

PCA nearest neighbor classification

- Compute PCA for all elements of training set $\Rightarrow \mathbf{y}_i = \mathbf{T}^H \mathbf{x}_i$
- Redefine training set as one with PCA transforms $\Rightarrow \mathcal{T} = \{(\mathbf{y}_i, z_i)\}_{i=1}^N$
- Compute PCA transform of (unknown) signal $\mathbf{x}$ $\Rightarrow \mathbf{y} = \mathbf{T}^H \mathbf{x}$
- PCA nearest neighbor classification rule

  $\Rightarrow$ Find nearest neighbor signal in training set with PCA transforms

$$\mathbf{y}_{\text{NN}} := \underset{\mathbf{y}_i \in \mathcal{T}}{\operatorname{argmin}} \|\mathbf{y}_i - \mathbf{y}\|^2$$

  $\Rightarrow$ Assign the label associated with the nearest neighbor

$$\mathbf{y}_{\text{NN}} \quad \Rightarrow \quad (\mathbf{y}_i, z_i) \quad \Rightarrow \quad z = z_i$$

- Reasonable enough. It should work. And it does

Why does PCA work for face recognition?

- Recall: image = a part that belongs to the person + noise

$$\mathbf{x}_i = \tilde{\mathbf{x}}_i + \mathbf{w}$$

- PCA transformation $\mathbf{T} = [\mathbf{v}_0^T; \ldots; \mathbf{v}_{N-1}^T]$ leads to

$$\mathbf{y}_i = \mathbf{T}\mathbf{x}_i = \mathbf{T}\tilde{\mathbf{x}}_i + \mathbf{T}\mathbf{w}$$

- PCA concentrates energy of $\tilde{\mathbf{x}}_i$ on a few components
- But it keeps the energy of the noise on all components

- Keeping principal components improves the accuracy of classification

  $\Rightarrow$ Because it increases the signal to noise ratio

PCA on the training set

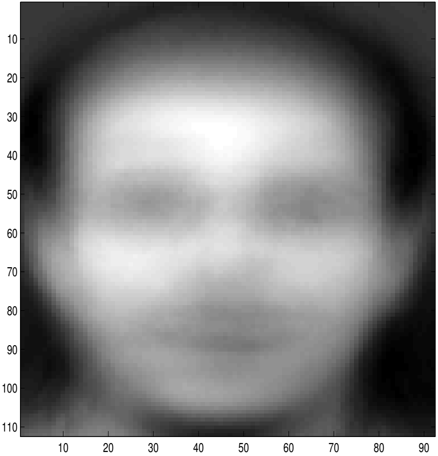
- The training set $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_{360}\}$ where $\mathbf{x}_i \in \mathbb{R}^{10304}$ is given
- Compute the mean vector and the covariance matrix as

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \quad \text{and} \quad \Sigma := \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}}_i)(\mathbf{x}_i - \bar{\mathbf{x}}_i)^T.$$

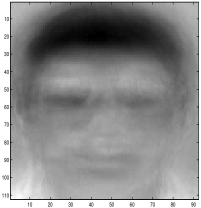
- Find the k largest eigenvalues of Σ
- Store their corresponding eigenvalues $\mathbf{v}_0, \ldots, \mathbf{v}_{k-1} \in \mathbb{R}^{10304}$ as P.C.

  $\Rightarrow$ The Principal Components $\mathbf{v}_0, \ldots, \mathbf{v}_{k-1}$ are called eigenfaces
- Create the PCA transform matrix as $\mathbf{T} = [\mathbf{v}_0^T; \ldots; \mathbf{v}_{k-1}^T]$
- Project the training set into the space of P.C.s $\mathbf{y}_i = \mathbf{T}\mathbf{x}_i$
- $\boldsymbol{\Sigma}$ depends training set, but is also a good description of the test set

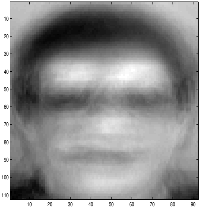
- The average face of the training set



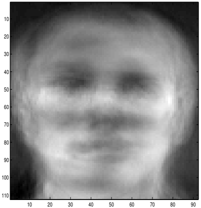
- The top 6 eigenfaces of the training set.



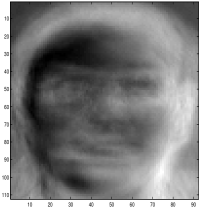
(1)



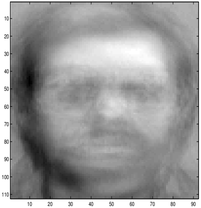
(2)



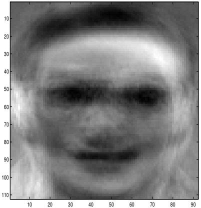
(3)



(4)



(5)

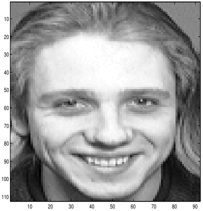


(6)

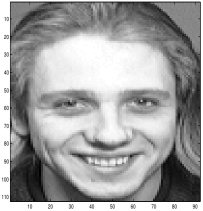
Finding the nearest neighbor

Num. of P.C. | test point | N.N. in the training set

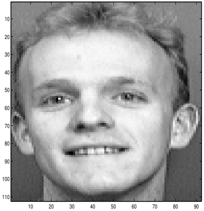
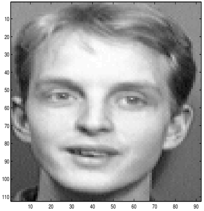
$k = 1$



$k = 5$



PCA improves classification accuracy

Classification method	test point	result of classification
Naive N.N.		
PCA-ed($k = 5$) N.N.