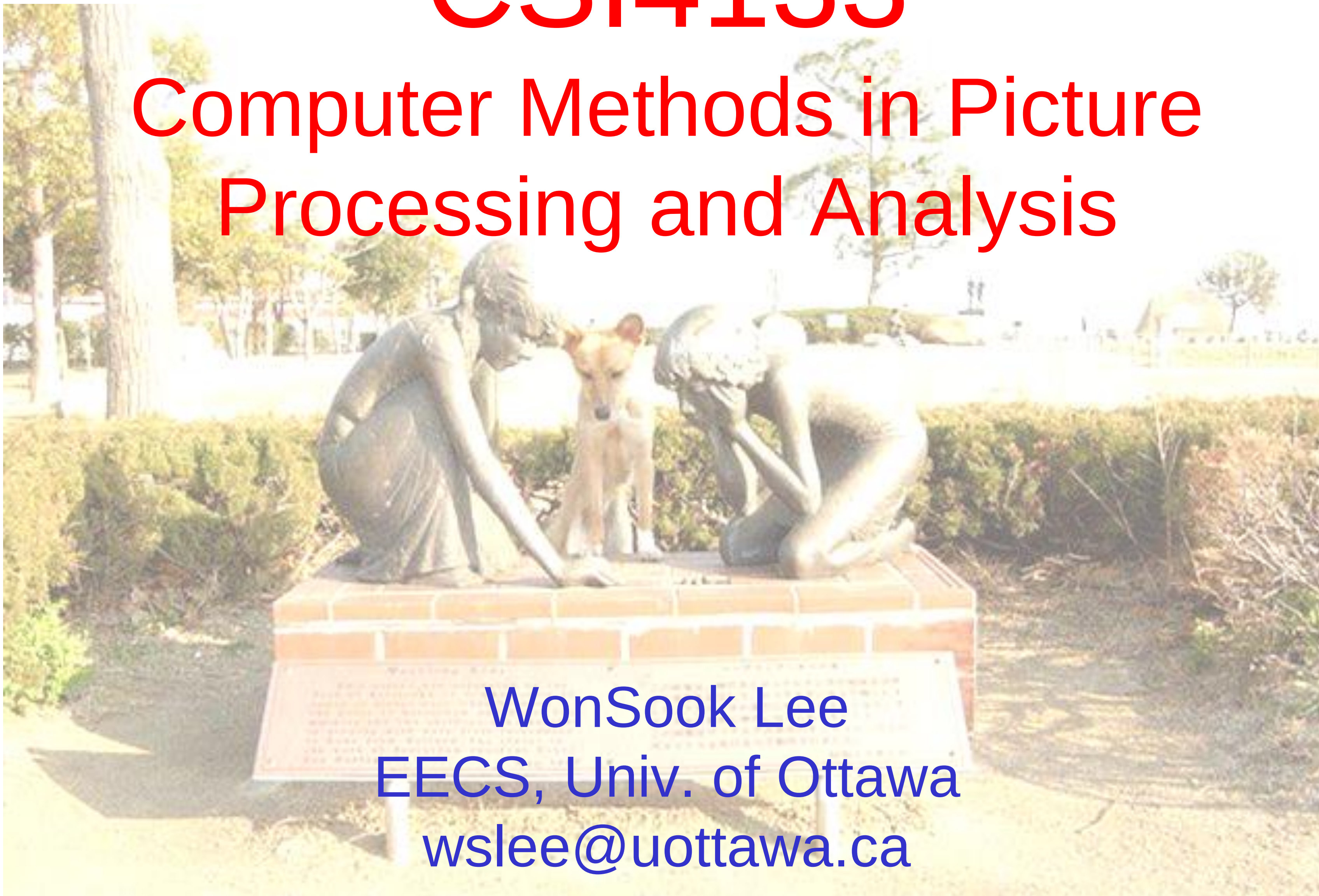
CSI4133

Computer Methods in Picture Processing and Analysis



WonSook Lee
EECS, Univ. of Ottawa
wslee@uottawa.ca

FACE RECOGNITION

Principal Components Analysis (PCA)

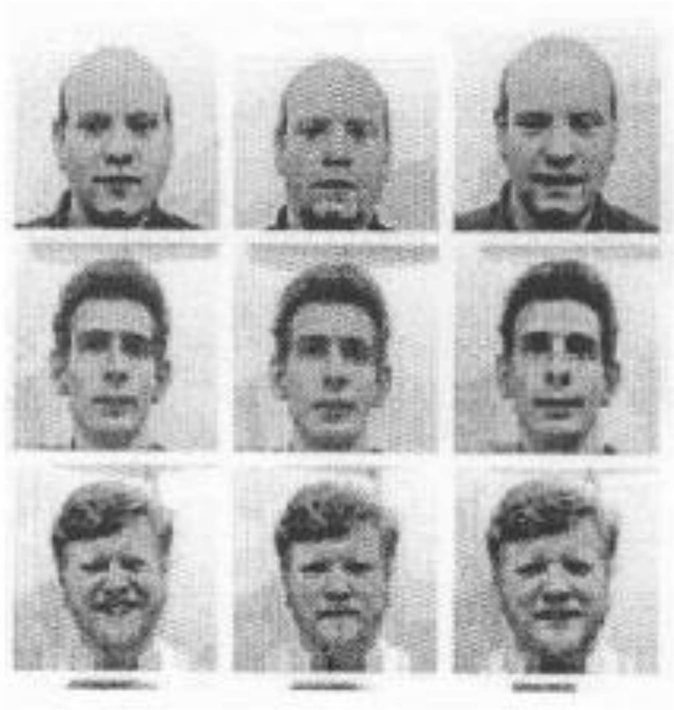
Face Recognition Problem

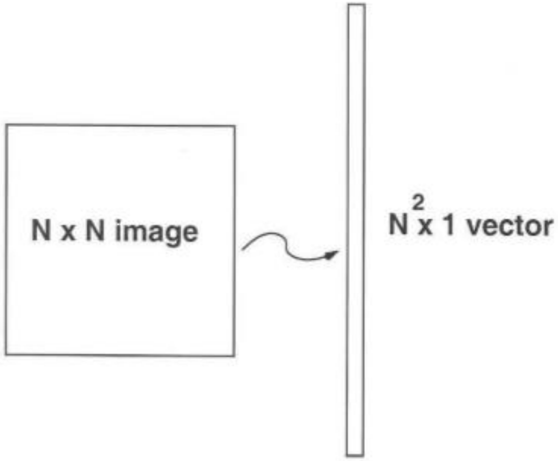
Who is this person?

A Statistical Approach

- Build a statistical model of face images
- Reduce a face image to a small number of parameters
- Classify / match face images using face parameters

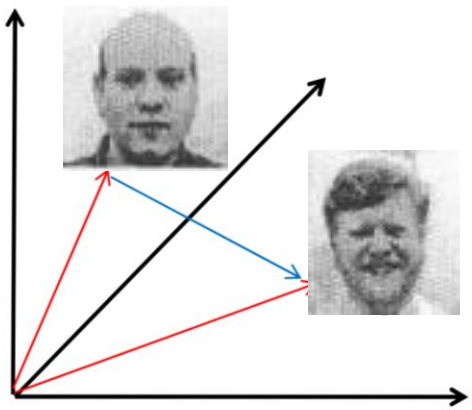
Represent images as vectors





N = 256, N^2 = 65536

Space of human faces



- An image is a point in a high-dimensional vector space

$$\boldsymbol{\varphi} \in \boldsymbol{R}^{d}$$

- Is there a low-dimensional sub-space for face images?

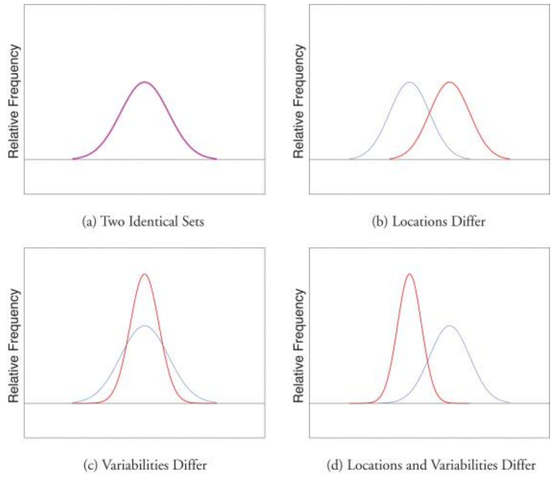
Variance, Covariance, Matrix

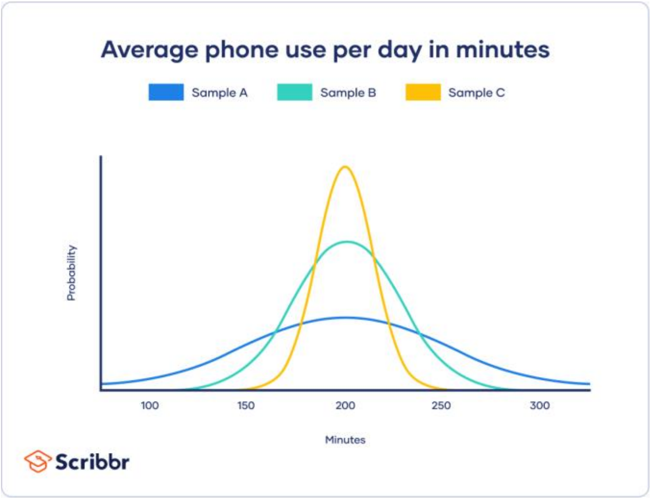
Mean and Variances

- Data set: 2, 7, 3, 12, 9
- Mean
  - The sum is 33. The mean is 33 ÷ 5 = 6.6
- Variance
  - (The squared differences are added) ÷ 5
  - (21.16 + 0.16 + 12.96 + 29.16 + 5.76 = 69.20) ÷ 5 = 13.84
  - The variance is 13.84

Variances

- Variance tells you the degree of spread in your data set. **The more spread the data, the larger the variance is in relation to the mean**





Variances

$$\text{variance} = \sigma^2 = \frac{\sum (x_r - \mu)^2}{n}$$

Standard Deviation is the **square root** of the **Variance**

$$\text{standard deviation} \quad \sigma = \sqrt{\frac{\sum (x_r - \mu)^2}{n}}$$

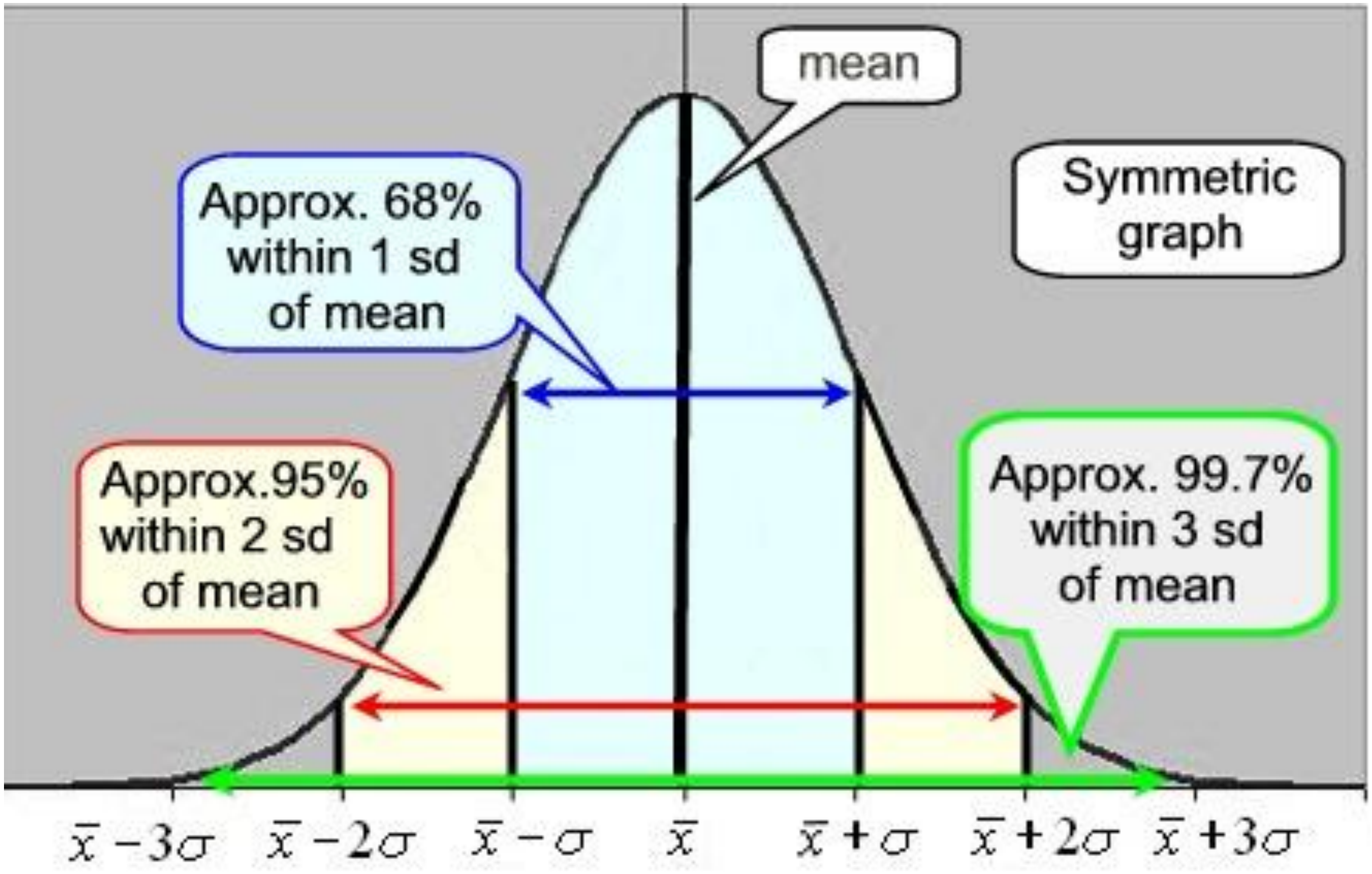
$$\mu = \text{mean}$$

Normal Curve Empirical Rule

- *Approximately* ...
  - 68% of the data lie within one standard deviation of the mean.
  - 95% of the data lies within two standard deviations of the mean.
  - 99.7% of the data lies within three standard deviations of the mean.

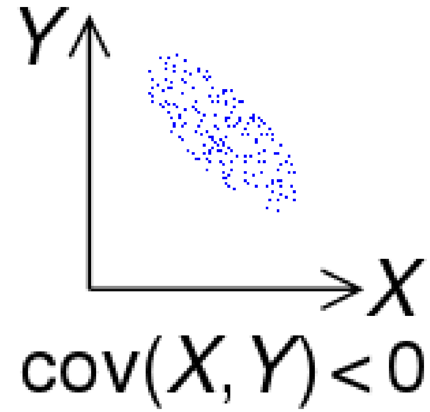
Normal Curve Empirical Rule

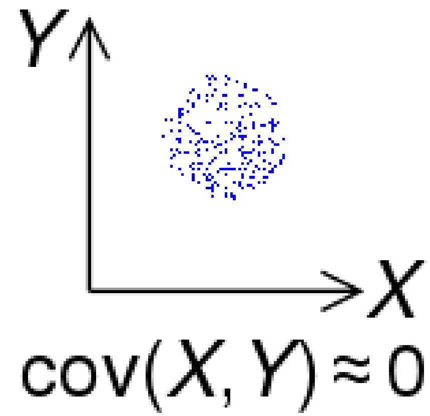
- *Approximately …*

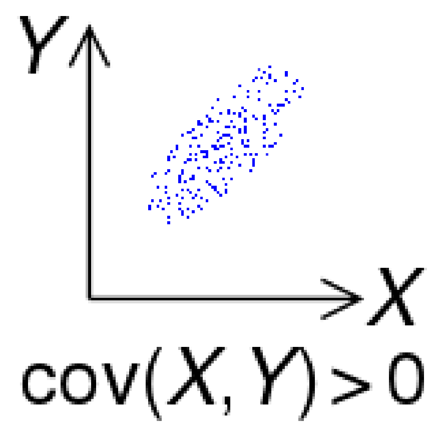


Covariance

- covariance is a measure of the joint variability of two random variables
- The sign of the covariance shows the tendency in the linear relationship between the variables.
- cov(X,Y)=E[(X-E[X])(Y-E[Y])]







Auto-covariance matrix of vectors

- For a vector $X = [X_1, X_2, \ldots X_n]^T$, its auto-covariance matrix is $cov(X, X)$
  – also known as the **variance–covariance matrix** or simply the **covariance matrix**
- $cov(X, X) = E[(X - E[X])(X - E[X])^T]$

Covariance Matrix

- If you have a set of m numeric data items, where each data item has d dimensions, then the covariance matrix is a d-by-d symmetric square matrix where there are variance values on the diagonal and covariance values off the diagonal.

Covariance Matrix

COVARIANCE MATRIX

	x_1	x_2	x_3	x_4
x_1	$Var(x_1)$	$Cov(x_1, x_2)$	$Cov(x_1, x_3)$	$Cov(x_1, x_4)$
x_2		$Var(x_2)$	$Cov(x_2, x_3)$	$Cov(x_2, x_4)$
x_3			$Var(x_3)$	$Cov(x_3, x_4)$
x_4				$Var(x_4)$

Covariance Matrix

X	Y	Z
Height	Score	Age
64.0	580.0	29.0
66.0	570.0	33.0
68.0	590.0	37.0
69.0	660.0	46.0
73.0	600.0	55.0
mean = 68.0	600.0	40.0

- m = 5
- d = 3

Covariance Matrix

- Covariance Matrix

	X	Y	Z
X	11.50	50.00	34.75
Y	50.00	1250.00	205.00
Z	34.75	205.00	110.00

- The 11.50 is the variance of X, 1250.0 is the variance of Y, and 110.0 is the variance of Z.

From data to covariance matrix

Sample data $Z_1, Z_2, \cdots, Z_M \in R^d$ $\quad \bar{Z} = \frac{1}{M}\sum_{i=1}^{M} z_i$

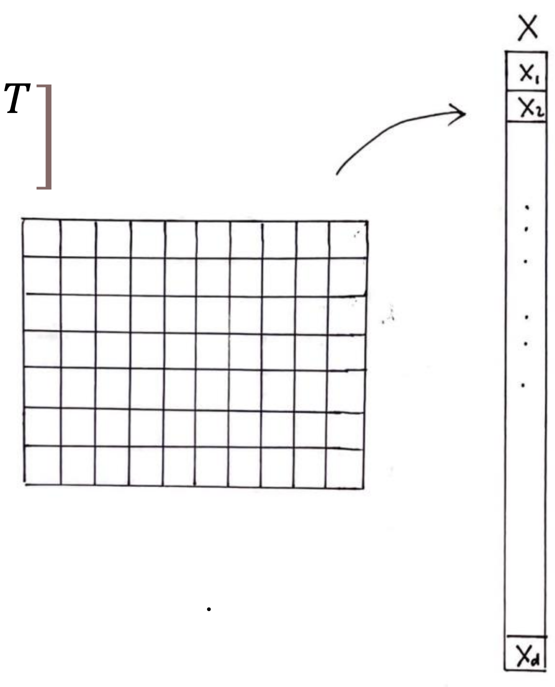
From random vector $X = [X_1, X_2, \cdots, X_d]^T$

Recall $\operatorname{cov}(X, X) = E\left[\left(X - E(X)\right)\left(X - E(X)\right)^T\right]$

Covariance matrix for data $C = \frac{1}{M}\sum_{i=1}^{M}\left(z_i - \bar{Z}\right)\left(z_i - \bar{Z}\right)^T$

Let $A = [z_1 - \bar{Z}, z_2 - \bar{Z}, \cdots, z_M - \bar{Z}]$

$$C = \frac{1}{M} A A^T$$



Eigenvalues and Eigenvectors

Eigenvalues and Eigenvectors

- **Definition**

Let A be an $n \times n$ matrix.

An ***eigenvector*** of A is a *nonzero* vector v in R^n such that $Av=\lambda v$, for some scalar λ.

An ***eigenvalue*** of A is a scalar λ such that the equation $Av=\lambda v$ has a *nontrivial* solution.

Consider the matrix

$$A = \begin{pmatrix} 2 & 2 \\ -4 & 8 \end{pmatrix} \quad \text{and vectors} \quad v = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad w = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Which are eigenvectors? What are their eigenvalues?

Solution

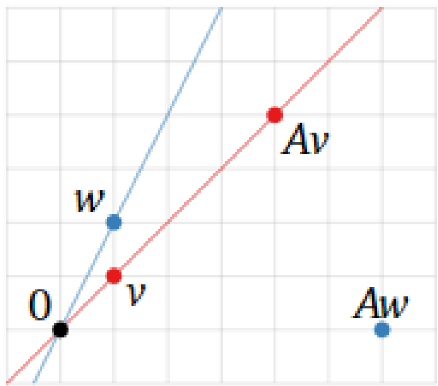
We have

$$Av = \begin{pmatrix} 2 & 2 \\ -4 & 8 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix} = 4v.$$

Hence, v is an eigenvector of A, with eigenvalue $\lambda = 4$. On the other hand,

$$Aw = \begin{pmatrix} 2 & 2 \\ -4 & 8 \end{pmatrix}\begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \end{pmatrix}.$$

which is not a scalar multiple of w. Hence, w is not an eigenvector of A.



v is an eigenvector

w is not an eigenvector

https://textbooks.math.gatech.edu/ila/eigenvectors.html

Eigenvalues and Eigenvectors

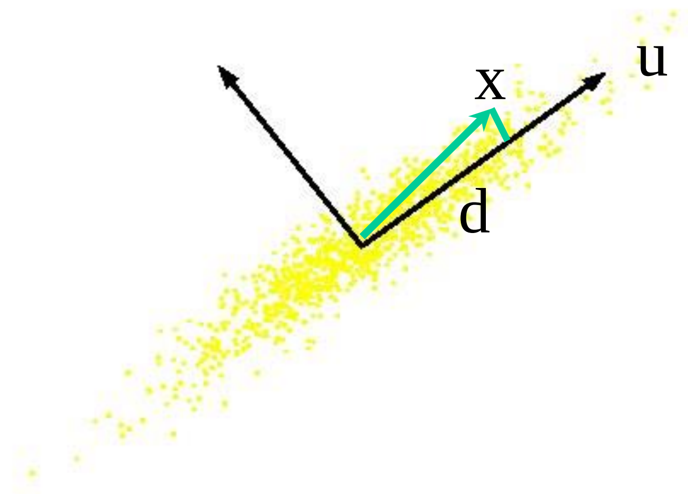
- Eigenvectors with distinct eigenvalues are linearly independent
  - An $n \times n$ matrix A has at most n eigenvalues.
  - In general, for any matrix, the eigenvectors are NOT always orthogonal. But for a special type of matrix, symmetric matrix, the eigenvalues are always real and the corresponding eigenvectors are always orthogonal.

Principal Component Analysis (PCA)

Principal Components Analysis (PCA)

- Principle
  - Linear projection method to reduce the number of parameters
  - Transfer a set of correlated variables into a new set of uncorrelated variables
  - Map the data into a space of lower dimensionality
  - Form of unsupervised learning
- Properties
  - It can be viewed as a rotation of the existing axes to new positions in the space defined by original variables
  - New axes are orthogonal and represent the directions with maximum variability

Vector projection



Suppose x is center at the mean of the dataset, project it onto a unit vector:

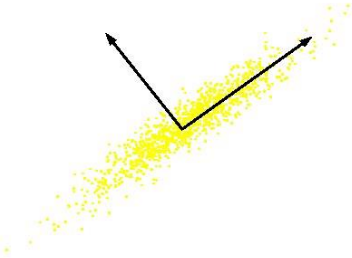
$$Proj_u(x) = (\mathrm{x} \cdot u)u$$

$$\|u\| = 1$$

$$d = \|(x \cdot u)u\| = |x \cdot u|$$

Computing the Components

- Data points are vectors in a multidimensional space
- Projection of vector **x** onto an axis (dimension) **u** is **u.x**
- Direction of greatest variability is that in which the average square of the projection is greatest
  - I.e. **u** such that E((**u.x**)^2) over all **x** is maximized
  - (we subtract the mean along each dimension, and center the original axis system at the centroid of all data points, for simplicity)
  - This direction of **u** is the direction of the first Principal Component



Computing the Components

- $E((\mathbf{u.x})^2) = E((\mathbf{u.x})(\mathbf{u.x})^T) = E(\mathbf{u.x.x}^T.\mathbf{u}^T)$
  - The matrix $\mathbf{C} = \mathbf{x.x}^T$ contains the correlations (similarities) of the original axes based on how the data values project onto them
  - So we are looking for **u** that maximizes $\mathbf{uCu}^T$, subject to **u** being unit-length
  - It is maximized when **u** is the principal eigenvector of the matrix **C**, in which case
    - $\mathbf{uCu}^T = \mathbf{u}\lambda\mathbf{u}^T = \lambda$ if **u** is unit-length, where λ is the principal eigenvalue of the correlation matrix C
    - The eigenvalue denotes the amount of variability captured along that dimension

PCA

Suppose $x_1, x_2, \ldots, x_M$ are N x 1 vectors

Step 1: $\bar{x} = \frac{1}{M} \sum_{i=1}^{M} x_i$

Step 2: subtract the mean: $\Phi_i = x_i - \bar{x}$

Step 3: form the matrix $A = [\Phi_1 \; \Phi_2 \cdots \Phi_M]$ ($N \mathrm{x} M$ matrix), then compute:

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T$$

(sample **covariance** matrix, $N \mathrm{x} N$, characterizes the *scatter* of the data)

Step 4: compute the eigenvalues of C: $\lambda_1 > \lambda_2 > \cdots > \lambda_N$

Step 5: compute the eigenvectors of C: $u_1, u_2, \ldots, u_N$

Methodology – cont.

- Since C is symmetric, $u_1, u_2, \ldots, u_N$ form a basis, (i.e., any vector x or actually $(x - \bar{x})$, can be written as a linear combination of the eigenvectors):

$$x - \bar{x} = b_1 u_1 + b_2 u_2 + \cdots + b_N u_N = \sum_{i=1}^{N} b_i u_i \qquad b_i = \frac{(x - \bar{x}).u_i}{(u_i.u_i)}$$

Step 6: (**dimensionality reduction step**) keep only the terms corresponding to the K largest eigenvalues:

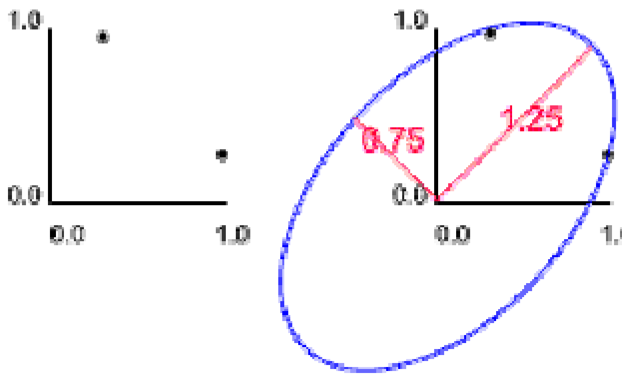
$$\hat{x} - \bar{x} = \sum_{i=1}^{K} b_i u_i \text{ where } K << N$$

- The representation of $\hat{x} - \bar{x}$ into the basis $u_1, u_2, \ldots, u_K$ is thus

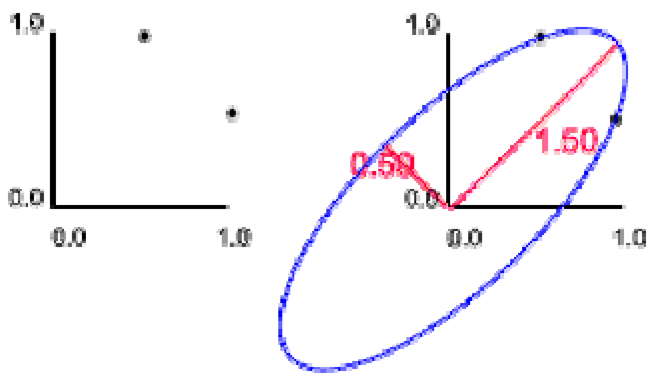
$$\begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_K \end{bmatrix}$$

Eigenvectors of a Correlation Matrix

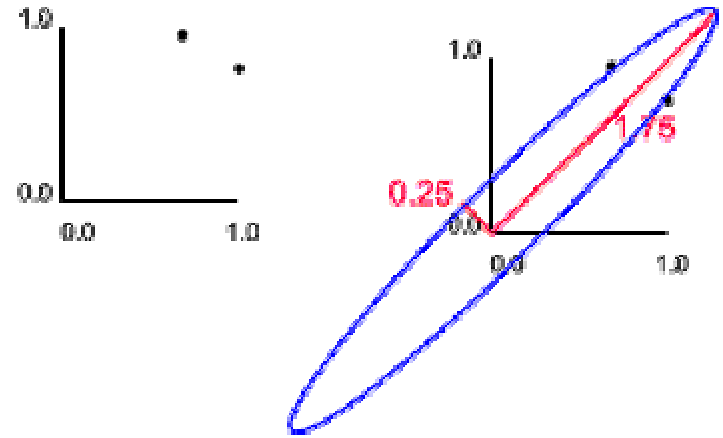
$$\begin{matrix} 1.00 & 0.25 \\ 0.25 & 1.00 \end{matrix}$$



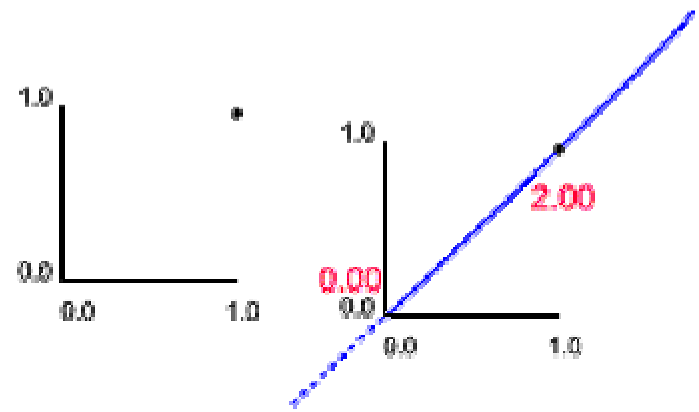
$$\begin{matrix} 1.00 & 0.50 \\ 0.50 & 1.00 \end{matrix}$$



$$\begin{matrix} 1.00 & 0.75 \\ 0.75 & 1.00 \end{matrix}$$



$$\begin{matrix} 1.00 & 1.00 \\ 1.00 & 1.00 \end{matrix}$$



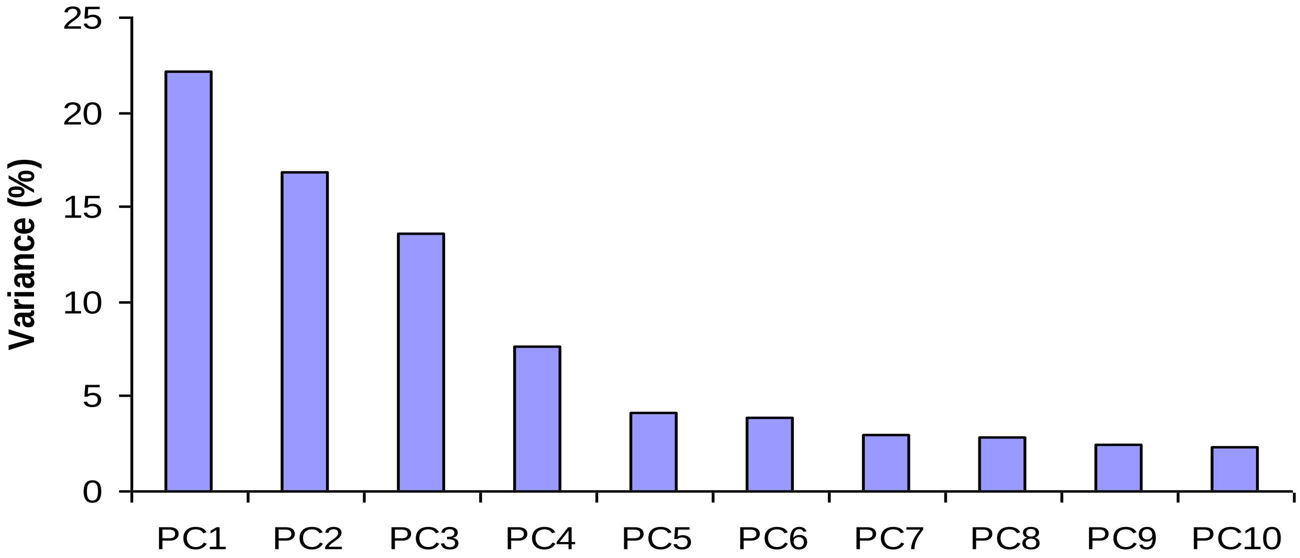
https://www.slideshare.net/ABINASHPADHY6/lecture1jpsppt

How Many PCs?

- For n original dimensions, correlation matrix is nxn, and has up to n eigenvectors. So n PCs.
- Where does dimensionality reduction come from?

Dimensionality Reduction

Can *ignore* the components of lesser significance.



You do lose some information, but if the eigenvalues are small, you don't lose much

- n dimensions in original data
- calculate n eigenvectors and eigenvalues
- choose only the first p eigenvectors, based on their eigenvalues
- final data set has only p dimensions

https://www.slideshare.net/ABINASHPADHY6/lecture1jpsppt

Face Recognition - Eigenfaces

Materials are borrowed from

M. Turk, A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, 3(1), pp. 71-86, 1991.

https://www.slideshare.net/ABINASHPADHY6/lecture1jpsppt
https://www.cse.unr.edu › Lectures › Eigenfaces
https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/

PCA applications -Eigenfaces

- Eigenfaces are
  the eigenvectors of
      the covariance matrix of
        the probability distribution of
          the vector space of human faces

PCA applications -Eigenfaces

- Eigenfaces are the 'standardized face ingredients' derived from the statistical analysis of many pictures of human faces
- A human face may be considered to be a combination of these standard faces

PCA applications -Eigenfaces

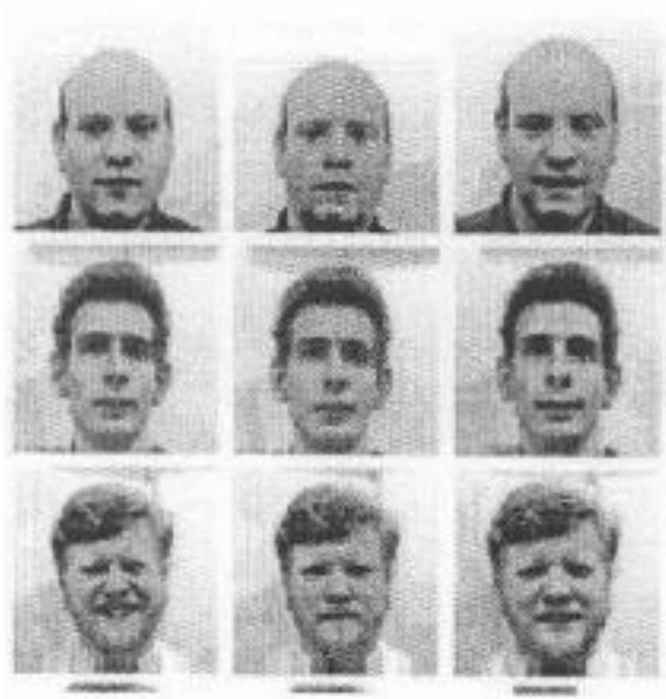
To generate a **set of eigenfaces**:

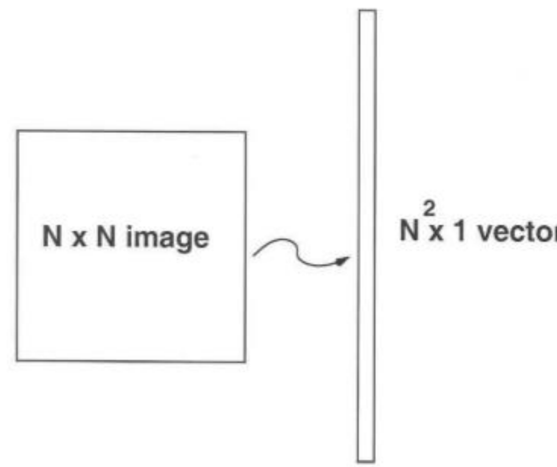
1. Large set of digitized images of human faces is taken under the same lighting conditions.
2. The images are normalized to line up the eyes and mouths.

Computation of eigenfaces

Step 1: obtain face images $I_1, I_2, \ldots, I_M$ (training faces)

(**very important:** the face images must be *centered* and of the same *size*)





Step 2: represent every image I_i as a vector Γ_i

Computation of eigenfaces

Step 3: compute the average face vector Ψ:

$$\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i$$

Step 4: subtract the mean face:

$$\Phi_i = \Gamma_i - \Psi$$

Step 5: compute the covariance matrix C:

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T \quad (N^2 \mathrm{x} N^2 \text{ matrix})$$

$$\text{where } A = [\Phi_1 \; \Phi_2 \cdots \Phi_M] \quad (N^2 \mathrm{x} M \text{ matrix})$$

N - An image of size 256 by 256 → a point in 65,536-D space.
M = 115 face images

Computation of eigenfaces

Step 6: compute the eigenvectors u_i of AA^T

The matrix AA^T is very large --> not practical !!

Step 6.1: consider the matrix A^TA (M x M matrix)

Step 6.2: compute the eigenvectors v_i of A^TA

$$A^TAv_i = \mu_i v_i$$

What is the relationship between us_i and v_i?

$A^TAv_i = \mu_i v_i \Rightarrow AA^TAv_i = \mu_i Av_i \Rightarrow$

$CAv_i = \mu_i Av_i$ or $Cu_i = \mu_i u_i$ where $u_i = Av_i$

Thus, AA^T and A^TA have the same eigenvalues and their eigenvectors are related as follows: $u_i = Av_i$!!

N - An image of size 256 by 256 → a point in 65,536-D space.
M = 115 face images

Computation of eigenfaces

Note 1: AA^T can have up to N^2 eigenvalues and eigenvectors.

Note 2: $A^T A$ can have up to M eigenvalues and eigenvectors.

Note 3: The M eigenvalues of $A^T A$ (along with their corresponding eigenvectors) correspond to the M *largest* eigenvalues of AA^T (along with their corresponding eigenvectors).

Step 6.3: compute the M best eigenvectors of AA^T: $u_i = Av_i$

(**important:** normalize u_i such that $\|u_i\| = 1$)

Step 7: keep only K eigenvectors (corresponding to the K largest eigenvalues)

K = 40 largest eigenfaces

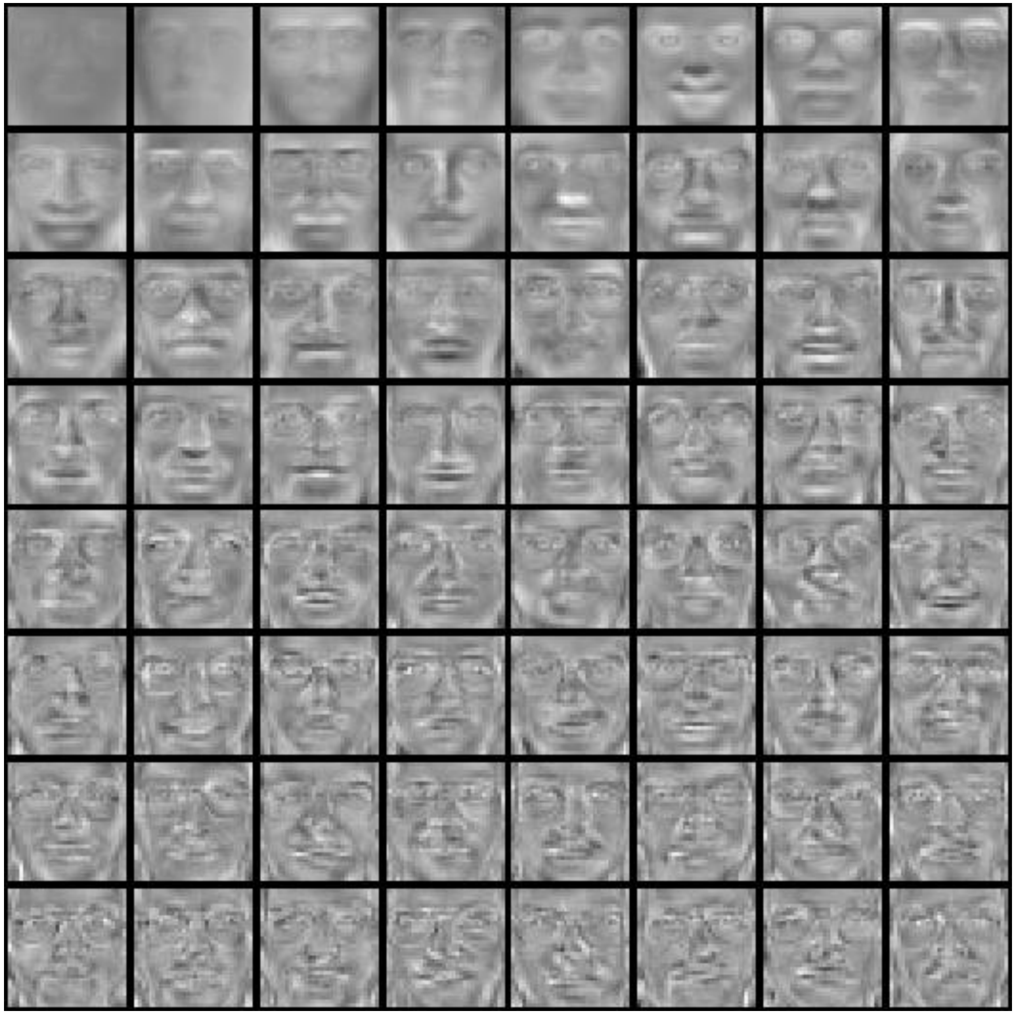
Eigenfaces example

Training images

Eigenfaces example

Mean: μ

Top eigenvectors: $u_1, \ldots u_k$



Faces onto this basis

- Each face (minus the mean) Φ_i in the training set can be represented as a linear combination of the best K eigenvectors:

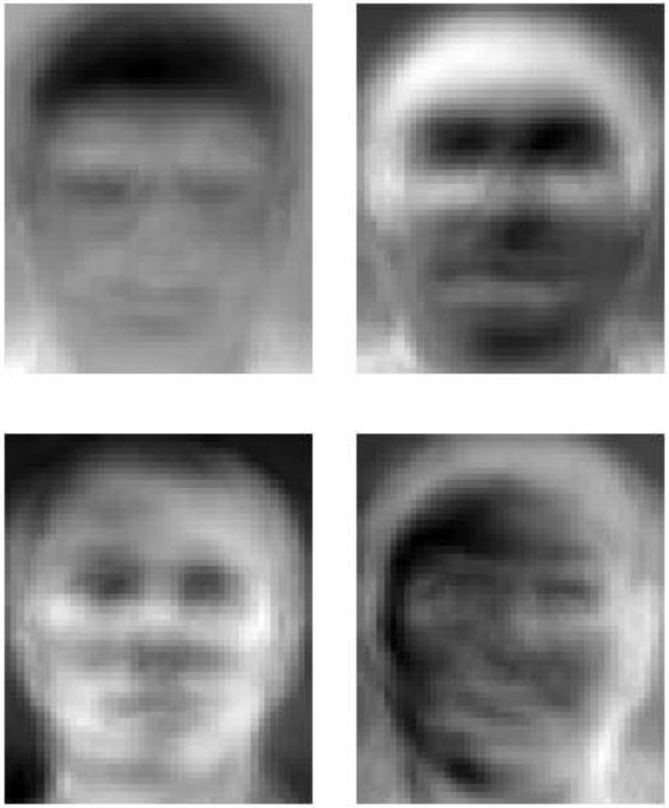
$$\hat{\Phi}_i - mean = \sum_{j=1}^{K} w_j u_j, \quad (w_j = u_j^T \Phi_i)$$

(we call the u_j's *eigenfaces*)

Face reconstruction:

PCA applications -Eigenfaces

- the principal eigenface looks like a bland androgynous average human face



http://en.wikipedia.org/wiki/Image:Eigenfaces.png

Eigenfaces – Face Recognition

- When properly weighted, eigenfaces can be summed together to create an approximate gray-scale rendering of a human face.
- Remarkably few eigenvector terms are needed to give a fair likeness of most people's faces
  - Hence eigenfaces provide a means of applying data compression to faces for identification purposes.
- Similarly, Expert Object Recognition in Video

Eigenfaces

- <u>Experiment and Results</u>

  Data used here are from the ORL database of faces. Facial images of 16 persons each with 10 views are used. - Training set contains 16×7 images.

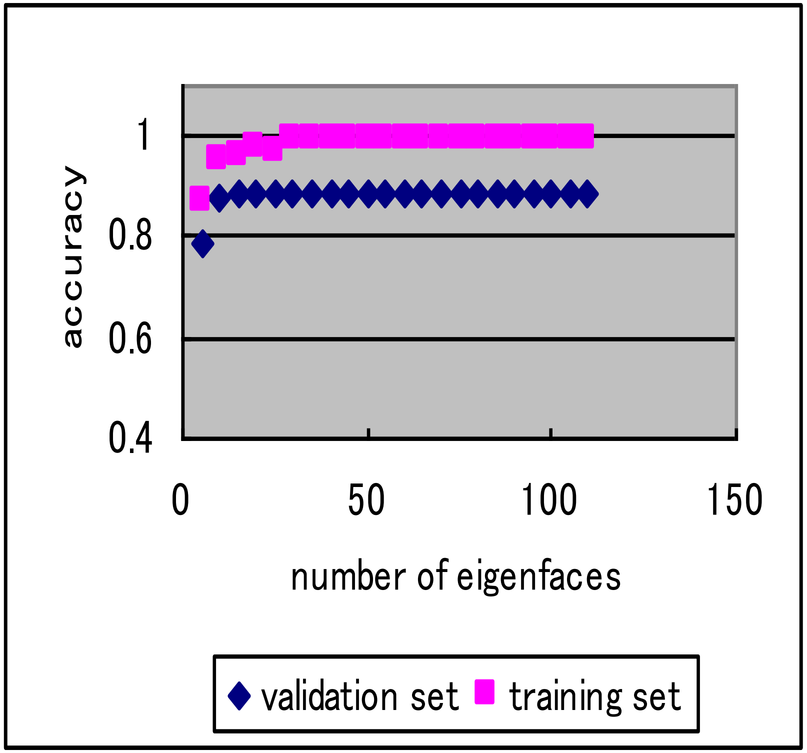
  Test set contains 16×3 images.

  First three eigenfaces :

Classification Using Nearest Neighbor

- Save average coefficients for each person. Classify new face as the person with the closest average.
- Recognition accuracy increases with number of eigenfaces till 15.

  Later eigenfaces do not help much with recognition.



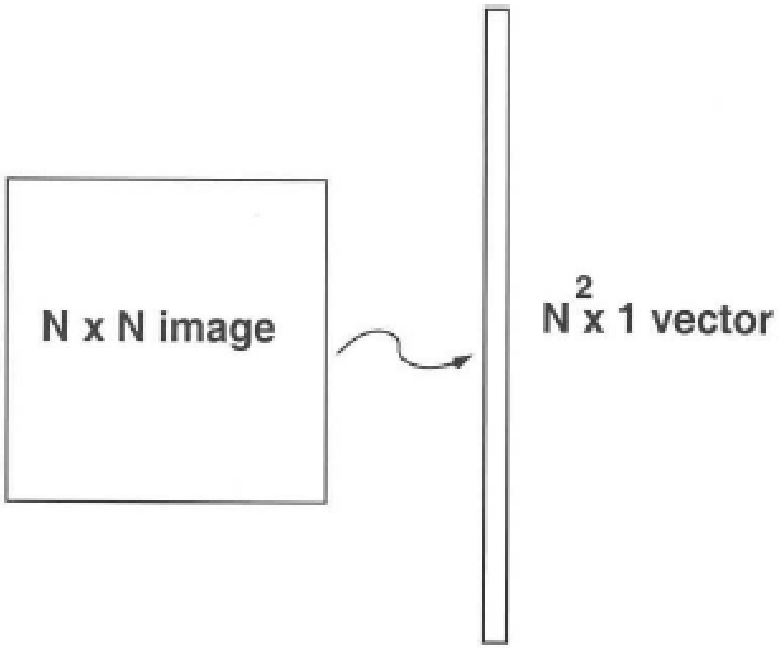
Best recognition rates

Training set 99%

Test set 89%

Eigenfaces

- Case Study: Eigenfaces for Face Detection/Recognition
- Face Recognition
  - The simplest approach is to think of it as a template matching problem
  - Problems arise when performing recognition in a high-dimensional space.
  - Significant improvements can be achieved by first mapping the data into a *lower dimensionality* space.



Face Recognition Using Eigenfaces

- Given an unknown face image Γ (centered and of the same size like the training faces) follow these steps:

Step 1: normalize Γ: $\Phi = \Gamma - \Psi$

Step 2: project on the eigenspace

$$\hat{\Phi} = \sum_{i=1}^{K} w_i u_i \quad (w_i = u_i^T \Phi) \qquad (where \| u_i \| = 1)$$

Step 3: represent Φ as: $\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_K \end{bmatrix}$

Step 4: find $e_r = \min_l \|\Omega - \Omega^l\|$ where $\| \Omega - \Omega^l \| = \sum_{i=1}^{K} (w_i - w_i^l)^2$

Step 5: if $e_r < T_r$, then Γ is recognized as face l from the training set.

Face Recognition Using Eigenfaces

- The distance e_r is called *distance within face space (difs)*

- The *Euclidean distance* can be used to compute e_r, however, the *Mahalanobis distance* has shown *to* work better:

$$\| \Omega - \Omega^k \| = \sum_{i=1}^{K} (w_i - w_i^k)^2$$

Euclidean distance

$$\| \Omega - \Omega^k \| = \sum_{i=1}^{K} \frac{1}{\lambda_i} (w_i - w_i^k)^2$$

Mahalanobis distance

(variations along all axes are treated as equally signifi cant)

Face detection and recognition

Detection

Recognition

“Sally”

Face Detection Using Eigenfaces

- Given an unknown image Γ

Step 1: compute $\Phi = \Gamma - \Psi$

Step 2: compute $\hat{\Phi} = \sum_{i=1}^{K} w_i u_i \quad (w_i = u_i^T \Phi)$ $\quad (where \|u_i\| = 1)$

Step 3: compute $e_d = \|\Phi - \hat{\Phi}\|$

Step 4: if $e_d < T_d$, then Γ is a face.

- The distance e_d is called distance from face space (dffs)

Backup slides

Consider the matrix

$$A = \begin{pmatrix} 0 & 6 & 8 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \end{pmatrix} \quad \text{and vectors} \quad v = \begin{pmatrix} 16 \\ 4 \\ 1 \end{pmatrix} \qquad w = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}.$$

Which are eigenvectors? What are their eigenvalues?

Solution

We have

$$Av = \begin{pmatrix} 0 & 6 & 8 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} 16 \\ 4 \\ 1 \end{pmatrix} = \begin{pmatrix} 32 \\ 8 \\ 2 \end{pmatrix} = 2v.$$

Hence, v is an eigenvector of A, with eigenvalue $\lambda = 2$. On the other hand,

$$Aw = \begin{pmatrix} 0 & 6 & 8 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 28 \\ 1 \\ 1 \end{pmatrix},$$

which is not a scalar multiple of w. Hence, w is not an eigenvector of A.

https://textbooks.math.gatech.edu/ila/eigenvectors.html

Let

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 6 \end{pmatrix} \qquad v = \begin{pmatrix} -3 \\ 1 \end{pmatrix}.$$

Is v an eigenvector of A? If so, what is its eigenvalue?

Solution

The product is

$$Av = \begin{pmatrix} 1 & 3 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} -3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = 0v.$$

Hence, v is an eigenvector with eigenvalue *zero*.

As noted above, an eigen*value* is allowed to be zero, but an eigen*vector* is not.

Eigenvalues and Eigenvectors

- **Rotation matrix in two-dimensions**

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \quad \text{where } 0 \le \theta < 2\pi$$

  - there are no real eigenvectors v
  - the eigenvalues

$$\lambda = \cos\theta \pm sqrt\,(\cos 2\,\theta - 1)$$

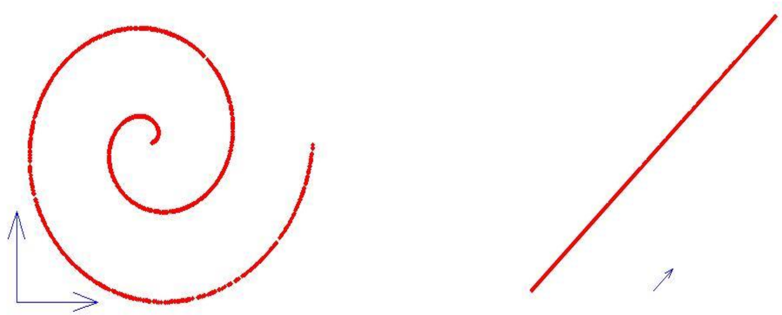
$$= \cos\theta \pm i \sin\theta$$

$$= e^{\pm i\theta}$$

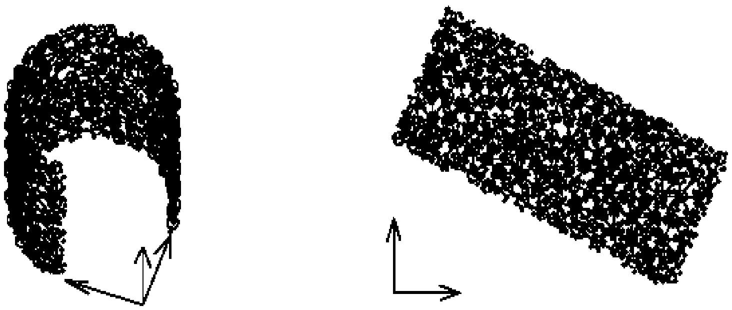
Limitations of PCA

- The reduction of dimensions for complex distributions may need non linear processing
- Curvilenear Component Analysis (CCA)
  - Non linear extension of PCA
  - Preserves the proximity between the points in the input space i.e. local topology of the distribution
  - Enables to unfold some varieties in the input data
  - Keep the local topology

Example of data representation using CCA



Non linear projection of a spiral



Non linear projection of a horseshoe