HCI/ComS 575X:
Computational Perception

Instructor: Alexander Stoytchev
http://www.cs.iastate.edu/~alex/classes/2007_Spring_575X/

Eigenfaces

February 21, 2007

M. Turk and A. Pentland (1991).

``Eigenfaces for recognition''.
Journal of Cognitive
Neuroscience, 3(1).

Dana H. Ballard (1999).

``An Introduction to Natural Computation
(Complex Adaptive Systems)'',
Chapter 4, pp 70-94, MIT Press.

Readings for Next Time

- Henry A. Rowley, Shumeet Baluja and Takeo Kanade (1997). ``Rotation Invariant Neural Network-Based Face Detection,'' Carnegie Mellon Technical Report, CMU-CS-97-201.
- Paul Viola and Michael Jones (2001). ``Robust Real-time Object Detection'', Second International Workshop on Statistical and Computational Theories of Vision Modeling, Learning, Computing, and Sampling, Vancouver, Canada, July 13, 2001.

Review of Eigenvalues and Eigenvectors

Review Questions

- What is a vector?
- What is a Matrix?
- What is the result when a vector is multiplied by a matrix? (Ax = ?)

Systems of LinearEquations

$m{\times}n$ matrix → $\boldsymbol{Ax} = \boldsymbol{b}$ ← m-vector (column)

n-vector (column)

Example:

$$\begin{pmatrix} 2 & 1 \\ -3 & 4 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$

$$\leftrightarrow \begin{matrix} 2x_1 + x_2 = 1 \\ -3x_1 + 4x_2 = 4 \end{matrix} \leftrightarrow \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

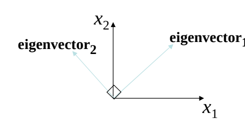
Intuitive Definition

- For any matrix there are some vectors such that the matrix multiplication changes only the magnitude of the vector.
- These vectors are called eigenvectors.

Mathematical Formulation

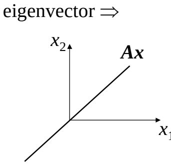
$$\boldsymbol{Ax} = \lambda \boldsymbol{x}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leftrightarrow \begin{matrix} \lambda = 1 \wedge x_1 = x_2 \\ \lambda = -1 \wedge x_1 = -x_2 \end{matrix}$$



A symmetric $\Rightarrow \lambda$ real
A positive definite $\Rightarrow \lambda > 0$
A positive semi-definite $\Rightarrow \lambda \geq 0$

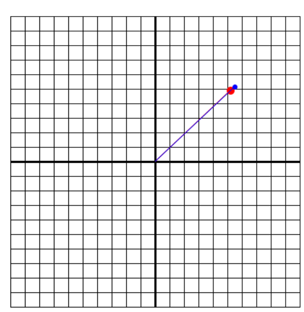
$\boldsymbol{x}$ an eigenvector $\Rightarrow$



On-line Java Applets

- http://www.math.duke.edu/education/webfeatsII/Lite_Applets/contents.html
- http://www.math.ucla.edu/~tao/resource/general/115a.3.02f/EigenMap.html

Java Applet Example

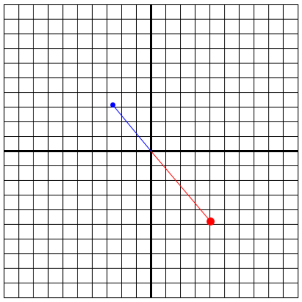


a11 = 0.30 a12 = 0.80

a21 = 0.90 a22 = 0.10

[http://www.math.duke.edu/education/webfeatsII/Lite_Applets/contents.html]

Java Applet Example



a11 = 0.30 a12 = 0.80

a21 = 0.90 a22 = 0.10

[http://www.math.duke.edu/education/webfeatsII/Lite_Applets/contents.html]

Finding the Eigenvectors

$$\begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \lambda \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

[From Ballard (1999)]

Finding the Eigenvectors

$$\begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} - \lambda \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Finding the Eigenvectors

$$\begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} - \lambda I \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Identity Matrix

Finding the Eigenvectors

$$\begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Finding the Eigenvectors

$$\begin{bmatrix} 3-\lambda & 1 \\ 2 & 2-\lambda \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

[From Ballard (1999)]

Finding the Eigenvectors

- The solution exists if:

Determinant, not matrix

$$|W| = 0 \qquad \textit{i.e.,} \qquad \begin{vmatrix} 3-\lambda & 1 \\ 2 & 2-\lambda \end{vmatrix} = 0$$

- Characteristic equation:

$$(3-\lambda)(2-\lambda) - 2 = 0$$

[From Ballard (1999)]

Finding the Eigenvectors

- The solutions to the characteristic equation are the eigenvalues

$$(3-\lambda)(2-\lambda) - 2 = 0$$

- In this case:

$$\lambda_1 = 4 \text{ and } \lambda_2 = 1$$

[From Ballard (1999)]

Finding the Eigenvectors

- Substituting with $\lambda_1 = 4$
- We get this system of equations

$$\begin{bmatrix} -1 & 1 \\ 2 & -2 \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

[From Ballard (1999)]

Finding the Eigenvectors

- Thus, first eigenvector is: $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$
- The corresponding eigenvalue is: $\lambda_1 = 4$
- Verification:

$$\begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 4 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

[From Ballard (1999)]

Eigenvectors and Eigenvalues in Matlab

```
>> A=[3 1; 2 2]
A =
     3     1
     2     2
```

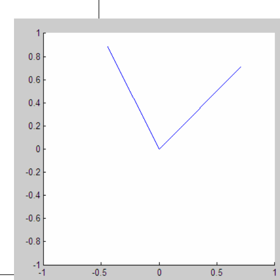
```
>>  [V, D]=eig(A)
V =
    0.7071   -0.4472
    0.7071    0.8944

D =
     4     0
     0     1
```

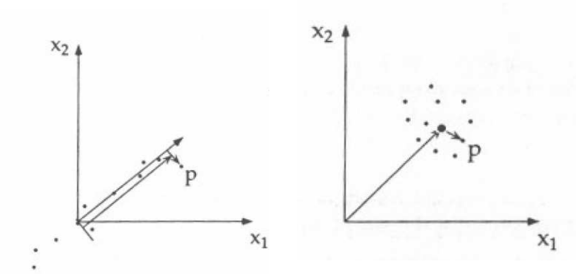
Eigenvectors and Eigenvalues in Matlab

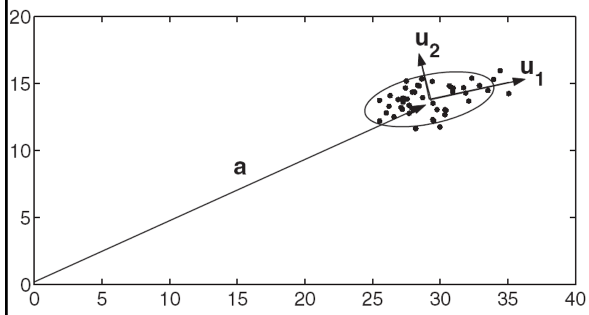
```
>> L = [ V(:,1),  [0; 0], V(:,2)]'
L =
     0.7071    0.7071
       0         0
    -0.4472    0.8944

>> line(L(:,1), L(:,2))
>> axis([-1, 1, -1, 1])
```



Eigenvectors v.s. Clustering





[http://web.math.umt.edu/bardsley/courses/471/eigenface.pdf]

Simple Example

Three Sample Data Points

$$X^1 = \begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix}, X^2 = \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix}, X^3 = \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix}$$

[From Ballard (1999)]

The mean is equal to …

$$M = \frac{1}{N}\sum_{k=1}^{N} X^k$$

$$M = \frac{1}{3}\left\{ \begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix} + \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix} \right\}$$

$$M = \frac{1}{3}\begin{pmatrix} 3 \\ 6 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

[From Ballard (1999)]

Subtract the Mean From All Data Points

$$X^1 - M = \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}, X^2 - M = \begin{pmatrix} 1 \\ -1 \\ -2 \end{pmatrix}, X^3 - M = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$$

[From Ballard (1999)]

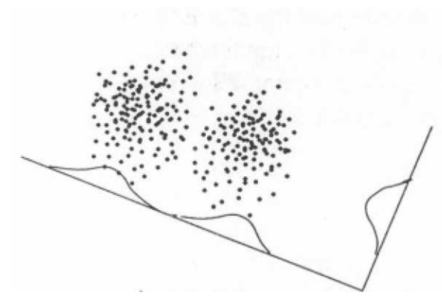
The Covariance Matrix is Given By

$$\Sigma = \frac{1}{N}\sum_{k=1}^{N}(X^k - M)(X^k - M)^T$$

$$\Sigma = \frac{1}{3}\left\{\begin{bmatrix} 4 & -2 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & -1 & -2 \\ -1 & 1 & 2 \\ -2 & 2 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix}\right\}$$

$$= \frac{1}{3}\begin{bmatrix} 6 & -3 & 0 \\ -3 & 2 & 2 \\ \not{-2}\ 0 & 2 & 8 \end{bmatrix}$$

[From Ballard (1999)]



[From Ballard (1999)]

[From Ballard (1999)]

Why can we drop 1/M?

$$\Sigma = \frac{1}{M}\sum_{n=1}^{M} X_n X_n^T$$

$$= AA^T$$

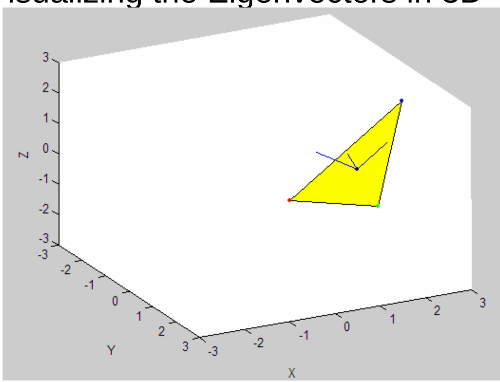
What about that Trick with the Dimensions?

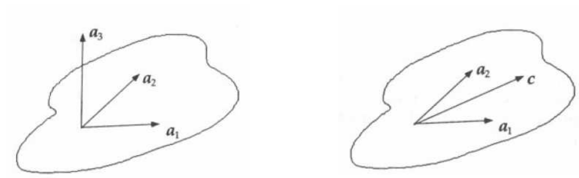
$$A^T A v = \mu v$$

$$AA^T A v = \mu A v$$

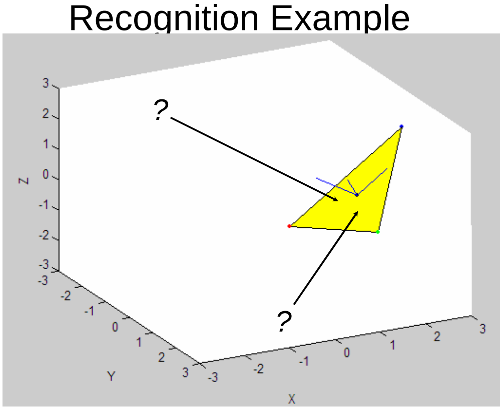
Matlab Demo

Visualizing the Eigenvectors in 3D

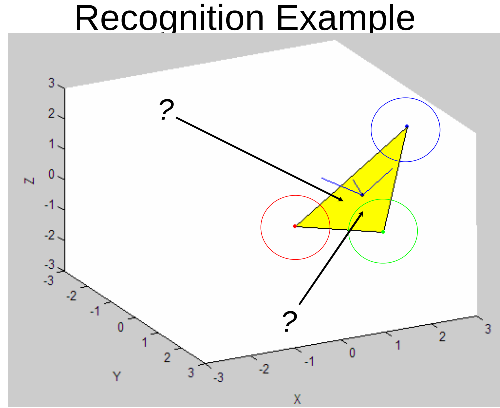




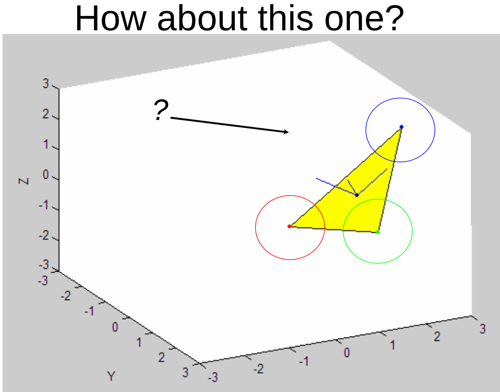
Recognition Example



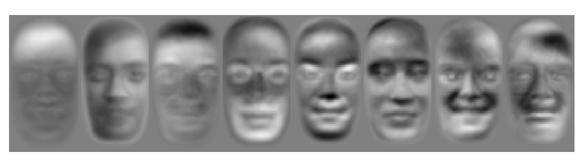
Recognition Example



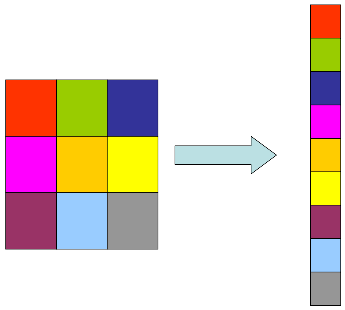
How about this one?



Eigenfaces



Representing images as vectors



The Next set of slides come from:

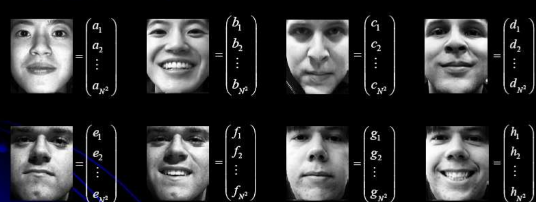
Prof. Ramani Duraiswami's class

CMSC: 426 Computer Vision

http://www.umiacs.umd.edu/~ramani/cmsc426/

Eigenfaces, the algorithm

- The database



[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- We compute the average face

$$\vec{m} = \frac{1}{M}\begin{pmatrix} a_1 & +b_1 & +\cdots+h_1 \\ a_2 & +b_2 & +\cdots+h_2 \\ \vdots & \vdots & \vdots \\ a_{N^2} & +b_{N^2} & +\cdots+h_{N^2} \end{pmatrix}, \quad where\ M = 8$$

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- Then subtract it from the training faces

$$\vec{a}_m = \begin{pmatrix} a_1 - m_1 \\ a_2 - m_2 \\ \vdots \\ a_{N^2} - m_{N^2} \end{pmatrix}, \ \vec{b}_m = \begin{pmatrix} b_1 - m_1 \\ b_2 - m_2 \\ \vdots \\ b_{N^2} - m_{N^2} \end{pmatrix}, \ \vec{c}_m = \begin{pmatrix} c_1 - m_1 \\ c_2 - m_2 \\ \vdots \\ c_{N^2} - m_{N^2} \end{pmatrix}, \ \vec{d}_m = \begin{pmatrix} d_1 - m_1 \\ d_2 - m_2 \\ \vdots \\ d_{N^2} - m_{N^2} \end{pmatrix},$$

$$\vec{e}_m = \begin{pmatrix} e_1 - m_1 \\ e_2 - m_2 \\ \vdots \\ e_{N^2} - m_{N^2} \end{pmatrix}, \ \vec{f}_m = \begin{pmatrix} f_1 - m_1 \\ f_2 - m_2 \\ \vdots \\ f_{N^2} - m_{N^2} \end{pmatrix}, \ \vec{g}_m = \begin{pmatrix} g_1 - m_1 \\ g_2 - m_2 \\ \vdots \\ g_{N^2} - m_{N^2} \end{pmatrix}, \ \vec{h}_m = \begin{pmatrix} h_1 - m_1 \\ h_2 - m_2 \\ \vdots \\ h_{N^2} - m_{N^2} \end{pmatrix}$$

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- Now we build the matrix which is N^2 by M

$$A = \left[\vec{a}_m\ \vec{b}_m\ \vec{c}_m\ \vec{d}_m\ \vec{e}_m\ \vec{f}_m\ \vec{g}_m\ \vec{h}_m\right]$$

- The covariance matrix which is N^2 by N^2

$$Cov = AA^{\mathrm{T}}$$

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- Find eigenvalues of the covariance matrix
  - The matrix is very large
  - The computational effort is very big
- We are interested in at most M eigenvalues
  - We can reduce the dimension of the matrix

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- Compute another matrix which is M by M

$$L = A^{\mathrm{T}} A$$

- Find the M eigenvalues and eigenvectors
  - Eigenvectors of Cov and L are equivalent
- Build matrix V from the eigenvectors of L

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- Eigenvectors of Cov are linear combination of image space with the eigenvectors of L

$$U = AV$$

- Eigenvectors represent the variation in the faces

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- Compute for each face its projection onto the face space

$$\Omega_1 = U^{\mathrm{T}}(\vec{a}_m),\quad \Omega_2 = U^{\mathrm{T}}(\vec{b}_m),\quad \Omega_3 = U^{\mathrm{T}}(\vec{c}_m),\quad \Omega_4 = U^{\mathrm{T}}(\vec{d}_m),$$
$$\Omega_5 = U^{\mathrm{T}}(\vec{e}_m),\quad \Omega_6 = U^{\mathrm{T}}(\vec{f}_m),\quad \Omega_7 = U^{\mathrm{T}}(\vec{g}_m),\quad \Omega_8 = U^{\mathrm{T}}(\vec{h}_m)$$

- Compute the threshold

$$\theta = \frac{1}{2}\max\left\{\left\|\Omega_i - \Omega_j\right\|\right\}\ for\ i, j = 1..M$$

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- To recognize a face

$$= \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{N^2} \end{pmatrix}$$

- Subtract the average face from it

$$\vec{r}_m = \begin{pmatrix} r_1 - m_1 \\ r_2 - m_2 \\ \vdots \quad \vdots \\ r_{N^2} - m_{N^2} \end{pmatrix}$$

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- Compute its projection onto the face space

$$\Omega = U^{\mathrm{T}}(\vec{r}_m)$$

- Compute the distance in the face space between the face and all known faces

$$\varepsilon_i^2 = \left\|\Omega - \Omega_i\right\|^2 \quad for\ i = 1..M$$

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- Reconstruct the face from eigenfaces

$$\vec{s} = U\Omega$$

- Compute the distance between the face and its reconstruction

$$\xi^2 = \|\vec{r}_m - \vec{s}\|^2$$

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- Distinguish between
  - If $\xi \geq \theta$ then it's not a face
  - If $\xi < \theta$ *and* $\varepsilon_i \geq \theta, (i = 1..M)$ then it's a new face
  - If $\xi < \theta$ *and* $\min\{\varepsilon_i\} < \theta$ then it's a known face

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Eigenfaces, the algorithm

- Problems with eigenfaces
  - Different illumination
  - Different head pose
  - Different alignment
  - Different facial expression

[http://www.umiacs.umd.edu/~ramani/cmsc426/]

Matlab Demo

THE END