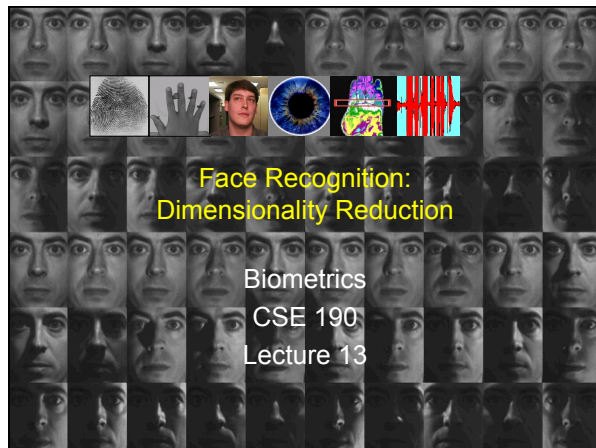




Announcements

- HW1 returned
- New attendance policy
 - On time: 1 point
 - Five minutes or more late: 0.5 points
 - Absent: 0 points

CSE190, Winter 2011

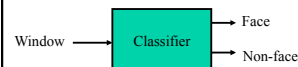
Face



CSE190, Winter 2011

Example: Face Detection

- Scan window over image.
- Classify window as either:
 - Face
 - Non-face

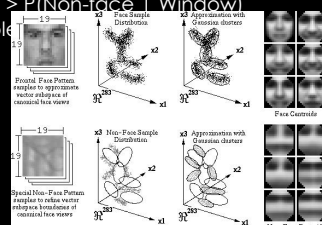


CS252A, Winter 2011

Computer Vision I

Discriminative Model

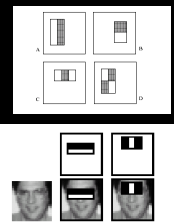
- Learn face and nonface models from examples $P(\text{Window} | \text{Face})$ and $P(\text{Window} | \text{Non-face})$
- Cluster samples of each class to create subclasses, and project the examples to a lower dimensional space based on multi-discriminant analysis.
- Detect faces in lower-dimensional space when $P(\text{Face} | \text{Window}) > P(\text{Non-face} | \text{Window})$
- Add non-face examples using bootstrapping



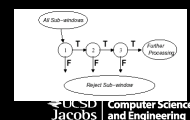
[Sung and Poggio 98]

"State of the Art Method:" Viola Jones

- [Viola and Jones CVPR 01]:
 - A face is modeled as a set of Harr-like features
 - A fast way to compute simple rectangle features
 - Use Adaboost to focus on a small set of features
 - Cascade of simple classifiers

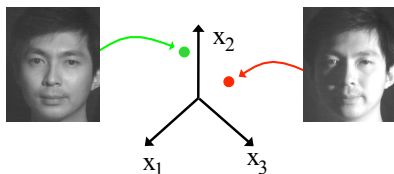


- Error rate comparable to the best
- Fast: 15 Fps, 700Mhz Pentium, half resolution video images



UCSD Computer Science and Engineering
Jacobs

Image as a Feature Vector



- Consider an n -pixel image to be a point in an n -dimensional space, $\mathbf{x} \in \mathbf{R}^n$.
- Each pixel value is a coordinate of $\mathbf{x}$.

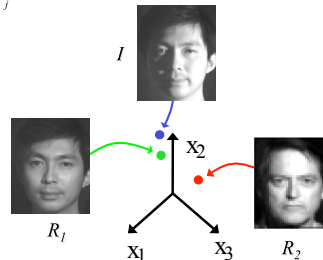
CS252A, Winter 2011

Computer Vision I

Nearest Neighbor Classifier

$\{R_j\}$ are set of training images.

$$ID = \arg \min_j \text{dist}(R_j, I)$$



CS252A, Winter 2011

Computer Vision I

Comments

- Sometimes called “Template Matching”
- Variations on distance function (e.g. L_1 , robust distances)
- Multiple templates per class- perhaps many training images per class.
- Expensive to compute k distances, especially when each image is big (N dimensional).
- May not generalize well to unseen examples of class.
- Some solutions:
 - Bayesian classification
 - Dimensionality reduction

CS252A, Winter 2011

Computer Vision I

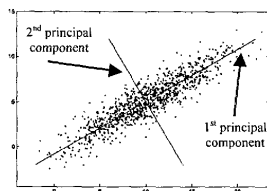
The Curse of Dimensionality

CS252A, Winter 2011

Computer Vision I

Eigenface (Turk, Pentland, 91) -1

- Use Principle Component Analysis (PCA) to determine the most discriminating features between images of faces.



CS252A, Winter 2011

Computer Vision I

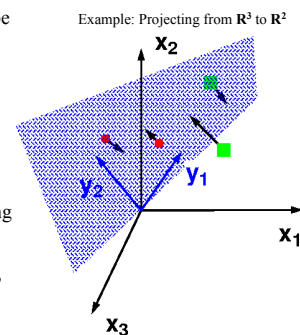
Eigenfaces: linear projection

- An n -pixel image $\mathbf{x} \in \mathbf{R}^n$ can be projected to a low-dimensional feature space $\mathbf{y} \in \mathbf{R}^m$ by

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

where $\mathbf{W}$ is an n by m matrix.

- Recognition is performed using nearest neighbor in $\mathbf{R}^m$.
- How do we choose a good $\mathbf{W}$?



CS252A, Winter 2011

Computer Vision I

Eigenfaces: Principal Component Analysis (PCA)

Assume we have a set of n feature vectors $\mathbf{x}_i$ ($i = 1, \dots, n$) in $\mathbb{R}^d$. Write

$$\mu = \frac{1}{n} \sum_i \mathbf{x}_i$$

$$\Sigma = \frac{1}{n-1} \sum_i (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T$$

The unit eigenvectors of Σ — which we write as $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d$, where the order is given by the size of the eigenvalue and $\mathbf{v}_1$ has the largest eigenvalue — give a set of features with the following properties:

- They are independent.
- Projection onto the basis $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ gives the k -dimensional set of linear features that preserves the most variance.

Algorithm 22.5: Principal components analysis identifies a collection of linear features that are independent, and capture as much variance as possible from a dataset.

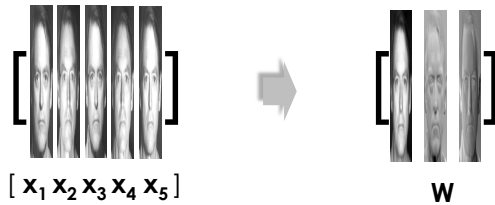
Some details:

- How big is Σ ?
- Use Singular value decomposition, “trick”
- to compute basis when $n \ll d$

CS252A, Winter 2011

Computer Vision I

How do you construct Eigenspace?



Construct data matrix by stacking vectorized images and then apply Singular Value Decomposition (SVD)

CS252A, Winter 2011

Computer Vision I

Matrix Decompositions

- Definition: The factorization of a matrix M into two or more matrices $M_1, M_2, \dots, M_n$, such that $M = M_1 M_2 \dots M_n$.
- Many decompositions exist...
 - QR Decomposition
 - LU Decomposition
 - LDU Decomposition
 - Etc.

CS252A, Winter 2011

Computer Vision I

Singular Value Decomposition

Excellent ref: “Matrix Computations,” Golub, Van Loan

- Any m by n matrix A may be factored such that

$$A = U \Sigma V^T$$

$$[m \times n] = [m \times m][m \times n][n \times n]$$

- U : m by m , orthogonal matrix
 - Columns of U are the eigenvectors of AA^T
- V : n by n , orthogonal matrix,
 - columns are the eigenvectors of $A^T A$
- Σ : m by n , diagonal with non-negative entries ($\sigma_1, \sigma_2, \dots, \sigma_s$) with $s = \min(m, n)$ are called the singular values
 - Singular values are the square roots of eigenvalues of both AA^T and $A^T A$
 - Result of SVD algorithm: $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_s$ (sorted by significance)

CS252A, Winter 2011

Computer Vision I

SVD Properties

- In Matlab $[U \Sigma V] = \text{svd}(A)$, and you can verify that: $A = U \Sigma V^T$
- $r = \text{Rank}(A) = \#$ of non-zero singular values.
- U, V give us orthonormal bases for the subspaces of A :
 - 1st r columns of U : Column space of A
 - Last $m - r$ columns of U : Left nullspace of A
 - 1st r columns of V : Row space of A
 - last $n - r$ columns of V : Nullspace of A
- For $d \leq r$, the first d column of U provide the best d -dimensional basis for columns of A in least squares sense.

CS252A, Winter 2011

Computer Vision I

Thin SVD

- Any m by n matrix A may be factored such that

$$A = U \Sigma V^T$$

$$[m \times n] = [m \times n][n \times n][n \times n]$$

- If $m > n$, then one can view Σ as:

$$\begin{bmatrix} \Sigma' \\ 0 \end{bmatrix}$$

- Where $\Sigma' = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_s)$ with $s = \min(m, n)$, and lower matrix is $(n - m)$ by m of zeros.

- Alternatively, you can write:

$$A = U \Sigma' V^T$$

- In Matlab, thin SVD is: $[U \Sigma V] = \text{svds}(A)$

CS252A, Winter 2011

Computer Vision I

Performing PCA with SVD

- Singular values of A are the square roots of eigenvalues of both AA^T and A^TA & Columns of U are corresponding Eigenvectors

- Given a collection of n vectors $a_1, \dots, a_n$

$$\sum_{i=1}^n a_i a_i^T = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix} \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}^T = AA^T$$

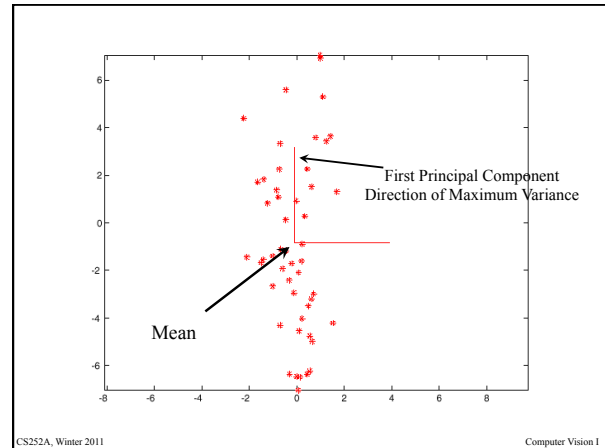
- Covariance matrix is:

$$\Sigma = \frac{1}{n} \sum_{i=1}^n (\bar{x}_i - \bar{\mu})(\bar{x}_i - \bar{\mu})^T$$

- So, ignoring $1/n$ subtract mean image μ from each input image, create data matrix, and perform thin SVD on the data matrix and take top Columns of U .

CS252A, Winter 2011

Computer Vision I



CS252A, Winter 2011

Computer Vision I

Eigenfaces

[Turk, Pentland 91]

- Modeling
 - Given a collection of n labeled training images,
 - Compute mean image and covariance matrix.
 - Compute k Eigenvectors (note that these are images) of covariance matrix corresponding to k largest Eigenvalues.
 - Project the training images to the k -dimensional Eigenspace.
- Recognition
 - Given a test image, project to Eigenspace.
 - Perform classification to the projected training images.

CS252A, Winter 2011

Computer Vision I

Eigenfaces: Training Images



[Turk, Pentland 91]

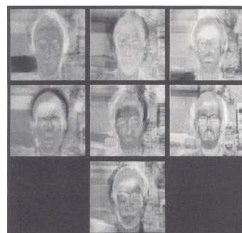
CS252A, Winter 2011

Computer Vision I

Eigenfaces



Mean Image

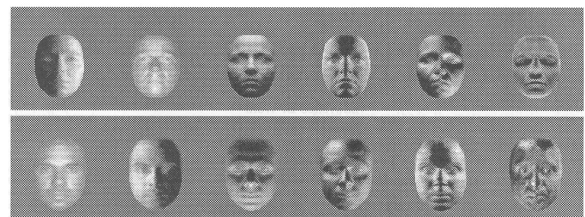


Basis Images

CS252A, Winter 2011

Computer Vision I

Variable Lighting



CS252A, Winter 2011

Computer Vision I

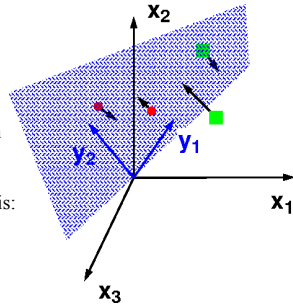
Projection, and reconstruction

- An n -pixel image $\mathbf{x} \in \mathbb{R}^n$ can be projected to a low-dimensional feature space $\mathbf{y} \in \mathbb{R}^m$ by

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$

- From $\mathbf{y} \in \mathbb{R}^m$, the reconstruction of the point is $\mathbf{W}^T\mathbf{y}$

- The error of the reconstruction is: $\|\mathbf{x} - \mathbf{W}^T\mathbf{W}\mathbf{x}\|$



CS252A, Winter 2011

Computer Vision I

Reconstruction using Eigenfaces

- Given image on left, project to Eigenspace, then reconstruct an image (right).



CS252A, Winter 2011

Computer Vision I

Underlying assumptions

- Background is not cluttered (or else only looking at interior of object)
- Lighting in test image is similar to that in training image.
- No occlusion
- Size of training image (window) same as window in test image.

CS252A, Winter 2011

Computer Vision I

Face detection using “distance to face space”

- Scan a window ω across the image, and classify the window as face/not face as follows:

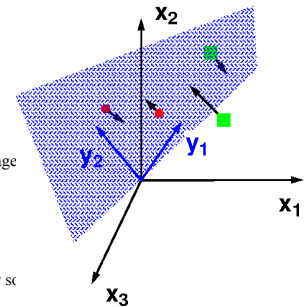
- Project window to subspace, and reconstruct as described earlier.

- Compute distance between ω and reconstruction.

- Local minima of distance over all image locations less than some threshold are taken as locations of faces.

- Repeat at different scales.

- Possibly normalize windows intensity so that $|\omega| = 1$.



CS252A, Winter 2011

Computer Vision I