

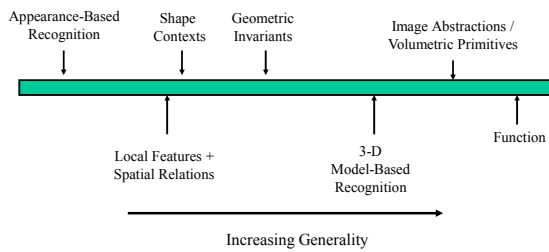
Recognition (Part 2)

Introduction to Computer Vision
CSE 152
Lecture 15

Announcements

- Homework 4 is due May 24, 11:59 PM
- Reading:
 - Chapter 15: Learning to Classify
 - Chapter 16: Classifying Images
 - Chapter 17: Detecting Objects in Images

A Rough Recognition Spectrum



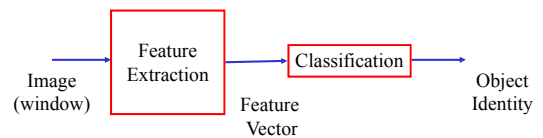
Appearance-Based Recognition

Appearance-Based Vision for Instances Level Recognition

- A Pattern Classification Viewpoint
 1. Bayesian Classification
 2. Appearance Manifolds
 3. Feature Space
 4. Dimensionality Reduction

Feature Space

- Sketch of a Pattern Recognition Architecture



Sliding window approaches



CSE 152, Spring 2017

Introduction to Computer Vision

Example: Face Detection

- Scan window over image
- Search over position & scale
- Classify window as either:
 - Face
 - Non-face



CSE 152, Spring 2017

Introduction to Computer Vision

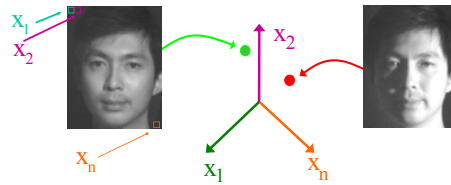
Feature Space

- **What are the features?**
- **What is the classifier?**

CSE 152, Spring 2017

Introduction to Computer Vision

The Space of Images



- We will treat an d -pixel image as a point in an d -dimensional space, $\mathbf{x} \in \mathbf{R}^d$.
- Each pixel value is a coordinate of \mathbf{x} .

CSE 152, Spring 2017

Introduction to Computer Vision

More features

- Filtered image
- Filter with multiple filters (bank of filters)
- Histogram of colors
- Histogram of Gradients (HOG)
- Haar wavelets
- Scale Invariant Feature Transform (SIFT)
- Speeded Up Robust Feature (SURF)

CSE 152, Spring 2017

Introduction to Computer Vision

Feature Space

- **What are the features?**
- **What is the classifier?**

CSE 152, Spring 2017

Introduction to Computer Vision

Nearest Neighbor Classifier

$\{R_j\}$ are set of training images.
 $ID = \arg \min_j \text{dist}(R_j, I)$

Variation of this:
 k nearest neighbors

CSE 152, Spring 2017 Introduction to Computer Vision

Comments on Nearest Neighbor

- 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 (d -dimensional)
- May not generalize well to unseen examples of class
- No worse than twice the error rate of the optimal classifier (if enough training samples)
- Some solutions:
 - Bayesian classification
 - Dimensionality reduction

CSE 152, Spring 2017 Introduction to Computer Vision

Do features vectors have structure in the image space?

- Faces of individuals cluster in the image space. (Not true)
- Faces of individuals are confined to a linear or affine subspace of \mathbf{R}^d
- Faces of an individual are approximated by a linear subspace
- Faces and objects lie on or near a manifold in the space of images

CSE 152, Spring 2017 Introduction to Computer Vision

An idea:

Represent the set of images as a linear subspace

What is a linear subspace?

Let V be a vector space and let W be a subset of V . Then W is a subspace if and only if:

1. The null vector $\mathbf{0}$ is in W
2. If \mathbf{u} and \mathbf{v} are elements of W , then any linear combination of \mathbf{u} and \mathbf{v} is an element of W ; $a\mathbf{u} + b\mathbf{v} \in W$
3. If \mathbf{u} is an element of W and c is a scalar, then the scalar product $c\mathbf{u} \in W$

- A k -dimensional subspace is spanned by k linearly independent vectors. It is spanned by a k -dimensional orthogonal basis

Example: A 2-D linear subspace of \mathbf{R}^3 spanned by y_1 and y_2

CSE 152, Spring 2017 Introduction to Computer Vision

Linear Subspaces & Linear Projection

- A d -pixel image $\mathbf{x} \in \mathbf{R}^d$ can be projected to a low-dimensional feature space $\mathbf{y} \in \mathbf{R}^k$ by

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

where W is an k by d matrix

- Each training image is projected to the subspace
- Recognition is performed in \mathbf{R}^k using, for example, nearest neighbor
- How do we choose a good W ?

Example: A 2-D linear subspace of \mathbf{R}^3 spanned by y_1 and y_2

CSE 152, Spring 2017 Introduction to Computer Vision

Linear Subspaces & Recognition

1. Eigenfaces: Approximate all training images as a single linear subspace
2. Distance to subspace: Represent lighting variation without shadowing for a single individual as a 3D linear subspace. n individuals are modeled as n 3D linear subspaces
3. Fisherfaces: Project all training images to a single subspace that enhances discriminability

CSE 152, Spring 2017 Introduction to Computer Vision

Eigenfaces: Principal Component Analysis (PCA)

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

$$\mu = \frac{1}{n} \sum_i x_i$$

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

The unit eigenvectors of Σ — which we write as v_1, v_2, \dots, v_d , where the order is given by the size of the eigenvalue and v_1 has the largest eigenvalue — give a set of features with the following properties:

- They are independent.
- Projection onto the basis $\{v_1, \dots, v_k\}$ 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.*

Eigen decomposition of covariance matrix.

Alternative: singular value decomposition of (mean-deviation form of) data matrix.

CSE 152, Spring 2017 Introduction to Computer Vision

Singular value decomposition and its relationship to eigen decomposition

- 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 called the singular values
 - **Singular values are the square roots of eigenvalues of both AA^T and $A^T A$**
 - *Result of SVD algorithm:* $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_s$

CSE 152, Spring 2017 Introduction to Computer Vision

SVD Properties

- In Matlab $[u \ s \ v] = \text{svd}(A)$, and you can verify that: $A = u * s * v'$
- $r = \text{Rank}(A) = \#$ of non-zero singular values.
- U, V give an 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
 - 1st $n - r$ columns of V : (Right) nullspace of A
- *For some d where $d \leq r$, the first d column of U provide the best d -dimensional basis for columns of A in least squares sense.*

CSE 152, Spring 2017 Introduction to Computer Vision

Performing PCA with SVD

- Singular values of A are the square roots of eigenvalues of AA^T (and $A^T A$)
- Columns of U are corresponding Eigenvectors of AA^T
- And $\sum_{i=1}^n a_i a_i^T = [a_1 \ a_2 \ \dots \ a_n] [a_1 \ a_2 \ \dots \ a_n]^T = AA^T$
- Covariance matrix is:

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^n (\bar{x}_i - \bar{\mu})(\bar{x}_i - \bar{\mu})^T$$
- So, ignoring $1/(n-1)$, subtract mean image μ from each input image, create a d by n data matrix, and perform thin SVD on the data matrix. $D = [x_1 - \mu \ | \ x_2 - \mu \ | \ \dots \ x_n - \mu \]$

CSE 152, Spring 2017 Introduction to Computer Vision

Economy SVD

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

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

$$[m \times n] = [m \times m][m \times n][n \times n]$$
- If $m > n$, then one can view Σ as: (i.e., more pixels than images)

$$\begin{bmatrix} \Sigma \\ 0 \end{bmatrix}$$
- 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, economy SVD is: $[U, S, V] = \text{svd}(A, 'econ')$

CSE 152, Spring 2017 Introduction to Computer Vision

PCA Example

CSE 152, Spring 2017 Introduction to Computer Vision

Eigenfaces

Modeling

1. Given a collection of n training images x_i , represent each one as a d -dimensional column vector
2. Compute the mean image and covariance matrix
3. Compute k Eigenvectors of the covariance matrix corresponding to the k largest Eigenvalues and form matrix $W^T = [u_1, u_2, \dots, u_k]$ (Or perform using SVD)
 - Note that the Eigenvectors are images
4. Project the training images to the k -dimensional Eigenspace.
 $y_i = Wx_i$

Recognition

1. Given a test image x , project the vectorized image to the Eigenspace by $y = Wx$
2. Perform classification of y to the projected training images

CSE 152, Spring 2017

Introduction to Computer Vision

Why is W a good projection?

- The linear subspace spanned by W maximizes the variance (i.e., the spread) of the projected data.
- W spans a subspace that is the best approximation to the data in a least squares sense. E.g., W is the subspace that minimizes the the sum of the squared distances from each datapoint to the the subspace.

CSE 152, Spring 2017

Introduction to Computer Vision

Eigenfaces: Training Images



[Turk, Pentland 91]

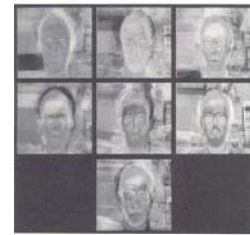
CSE 152, Spring 2017

Introduction to Computer Vision

Eigenfaces



Mean Image



Basis Images

CSE 152, Spring 2017

Introduction to Computer Vision

Difficulties with PCA

- Projection may suppress important detail
 - smallest variance directions may not be unimportant
- Method does not take discriminative task into account
 - typically, we wish to compute features that allow good discrimination
 - not the same as largest variance or minimizing reconstruction error.

CSE 152, Spring 2017

Introduction to Computer Vision

Alternative projections

CSE 152, Spring 2017

Introduction to Computer Vision

Fisherfaces: Class specific linear projection

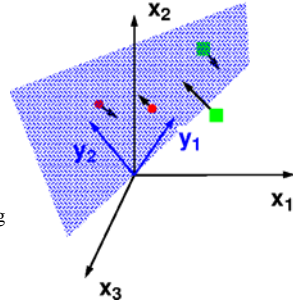
P. Belhumeur, J. Hespanha, D. Kriegman, *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection*, PAMI, July 1997, pp. 711–720.

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

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

where \mathbf{W} is an k by d matrix

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



CSE 152, Spring 2017

Introduction to Computer Vision

PCA & Fisher's Linear Discriminant

- Between-class scatter

$$S_B = \sum_{i=1}^c |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T$$

- Within-class scatter

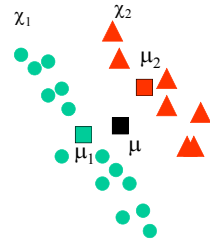
$$S_W = \sum_{i=1}^c \sum_{x_i \in \chi_i} (x_i - \mu_i)(x_i - \mu_i)^T$$

- Total scatter

$$S_T = \sum_{i=1}^c \sum_{x_i \in \chi_i} (x_i - \mu)(x_i - \mu)^T = S_B + S_W$$

- Where

- c is the number of classes
- μ_i is the mean of class χ_i
- $|\chi_i|$ is number of samples of χ_i .

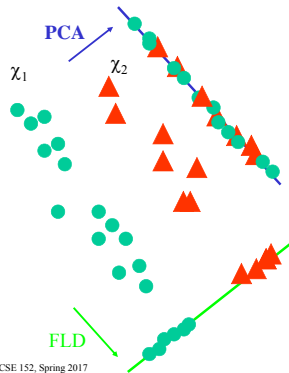


If the data points x_i are projected by $y_i = \mathbf{W}x_i$ and the scatter of x_i is S , then the scatter of the projected points y_i is $\mathbf{W}^T S \mathbf{W}$

CSE 152, Spring 2017

Introduction to Computer Vision

PCA & Fisher's Linear Discriminant



- PCA (Eigenfaces)

$$W_{PCA} = \arg \max_W |W^T S_T W|$$

Maximizes projected total scatter

- Fisher's Linear Discriminant

$$W_{fld} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|}$$

Maximizes ratio of projected between-class to projected within-class scatter

CSE 152, Spring 2017

Introduction to Computer Vision

Computing the Fisher Projection Matrix

$$W_{opt} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|} = [w_1 \ w_2 \ \dots \ w_m] \quad (4)$$

where $\{w_i | i = 1, 2, \dots, m\}$ is the set of generalized eigenvectors of S_B and S_W corresponding to the m largest generalized eigenvalues $\{\lambda_i | i = 1, 2, \dots, m\}$, i.e.,

$$S_B w_i = \lambda_i S_W w_i, \quad i = 1, 2, \dots, m$$

- The w_i are orthonormal
- There are at most $c-1$ non-zero generalized Eigenvalues, so $m \leq c-1$
- Can be computed with *eig* in Matlab

CSE 152, Spring 2017

Introduction to Computer Vision

Fisherfaces

$$W = W_{fld} W_{PCA}$$

$$W_{PCA} = \arg \max_W |W^T S_T W|$$

$$W_{fld} = \arg \max_W \frac{|W^T W_{PCA}^T S_B W_{PCA} W|}{|W^T W_{PCA}^T S_W W_{PCA} W|}$$

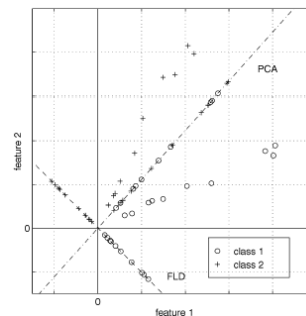
- Since S_W is rank $N-c$, project training set to subspace spanned by first $N-c$ principal components of the training set.
- Apply FLD to $N-c$ dimensional subspace yielding $c-1$ dimensional feature space.

- Fisher's Linear Discriminant projects away the within-class variation (lighting, expressions) found in training set.
- Fisher's Linear Discriminant preserves the separability of the classes.

CSE 152, Spring 2017

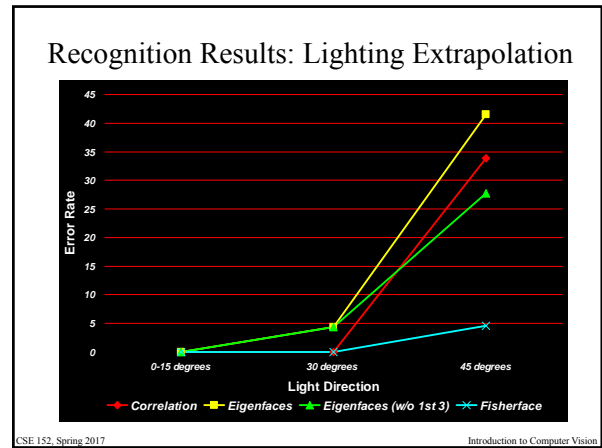
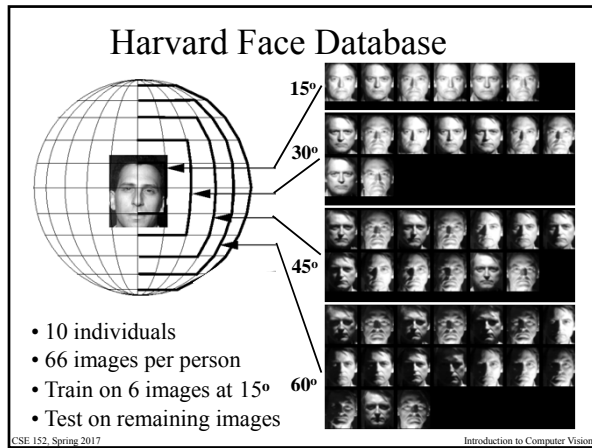
Introduction to Computer Vision

PCA vs. FLD



CSE 152, Spring 2017

Introduction to Computer Vision



Next Lecture

- Recognition, detection, and classification
- Reading:
 - Chapter 15: Learning to Classify
 - Chapter 16: Classifying Images
 - Chapter 17: Detecting Objects in Images

CSE 152, Spring 2017 Introduction to Computer Vision