Recognition
(Part 3)
Introduction to Computer Vision
CSE 152
Lecture 16
Announcements

• Homework 4 is due today at 11:59 PM
• Homework 5 will be assigned today
  – Due Sat, Jun 9, 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
- Shape Contexts
- Geometric Invariants
- Local Features + Spatial Relations
- 3-D Model-Based Recognition
- Image Abstractions / Volumetric Primitives
- Function

Increasing Generality
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
The Space of Images

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

\{ R_j \} are set of training images.

\[ ID = \arg \min_j dist(R_j, I) \]

Variation of this: \( k \) nearest neighbors
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 $\mathbb{R}^d$
- Faces of an individual are approximated by a linear subspace
- Faces and objects lie on or near a manifold in the space of images
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 $0$ is in $W$
2. If $u$ and $v$ are elements of $W$, then any linear combination of $u$ and $v$ is an element of $W$; $au + bv \in W$
3. If $u$ is an element of $W$ and $c$ is a scalar, then the scalar product $cu \in W$

- 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 $\mathbb{R}^3$ spanned by $y_1$ and $y_2$
Linear Subspaces & Linear Projection

- A $d$-pixel image $x \in \mathbb{R}^d$ can be projected to a low-dimensional feature space $y \in \mathbb{R}^k$ by
  \[ y = Wx \]
  where $W$ is an $k$ by $d$ matrix
- Each training image is projected to the subspace
- Recognition is performed in $\mathbb{R}^k$ using, for example, nearest neighbor
- How do we choose a good $W$?

Example: A 2-D linear subspace of $\mathbb{R}^3$ spanned by $y_1$ and $y_2$
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
Eigenfaces: Principal Component Analysis (PCA)

Assume we have a set of $n$ feature vectors $\mathbf{x}_i (i = 1, \ldots, n)$ 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 $\Sigma$ — which we write as $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_d$, 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 $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ 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.

**Eigen decomposition of covariance matrix.**

**Alternative:** singular value decomposition of (mean-deviation form of) data matrix.
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 \( A A^T \)

• \( V \): \( n \) by \( n \), orthogonal matrix,
  – Columns are the eigenvectors of \( A^T A \)

• \( \Sigma \): \( m \) by \( n \), diagonal with non-negative entries \( (\sigma_1, \sigma_2, \ldots, \sigma_s) \) with \( s = \min(m,n) \) are called the singular values
  – Singular values are the square roots of eigenvalues of both \( A A^T \) and \( A^T A \)
  – Result of SVD algorithm: \( \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s \)
SVD Properties

• $r = \text{Rank}(A) = \# \text{ of non-zero singular values.}$
• $U, V$ 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.*
Performing PCA with SVD

• Singular values of $A$ are the square roots of eigenvalues of $AA^T$ (and $A^TA$)

• Columns of $U$ are corresponding Eigenvectors of $AA^T$

• And $\sum_{i=1}^{n} a_i a_i^T = [a_1 \ a_2 \ \cdots \ a_n][a_1 \ a_2 \ \cdots \ a_n]^T = AA^T$

• Covariance matrix is:

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^\top$$

• So, ignoring $1/(n-1)$, subtract mean image $\mu$ from each input image, create a $d$ by $n$ data matrix, and perform thin SVD on the data matrix. $D=[x_1-\mu \ | \ x_2-\mu \ | \ \cdots \ | \ x_n-\mu]$
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 $\Sigma$ as: (i.e., more pixels than images)
  $$\begin{bmatrix}
  \Sigma' \\
  0
  \end{bmatrix}$$
- Where $\Sigma' = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s)$ with $s=n$, and lower submatrix is $(m-n \times n)$ of zeros.
- Alternatively, you can write:
  $$A = U'\Sigma' V^T$$
PCA Example

First Principal Component
Direction of Maximum Variance

Mean

$\mu$

$\mathbf{v}_1$

$\mathbf{v}_2$
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 $W^T=[u_1, u_2, ..., u_k]$ (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
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 sum of the squared distances from each datapoint to the subspace.
Eigenfaces: Training Images

[ Turk, Pentland 91]
Eigenfaces

Mean Image

Basis Images
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.
Alternative projections
Fisherfaces: Class specific linear projection


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

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

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

- Recognition is performed using nearest neighbor in $\mathbb{R}^k$

- How do we choose a good $\mathbf{W}$?
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**
  \[ S_W = \sum_{i=1}^{c} \sum_{x_k \in \chi_i} (x_k - \mu_i)(x_k - \mu_i)^T \]
- **Total scatter**
  \[ S_T = \sum_{i=1}^{c} \sum_{x_k \in \chi_i} (x_k - \mu)(x_k - \mu)^T = S_B + S_W \]
- **Where**
  - \( c \) is the number of classes
  - \( \mu_i \) is the mean of class \( \chi_i \)
  - \( |\chi_i| \) is number of samples of \( \chi_i \).

If the data points \( x_i \) are projected by \( y_i=Wx_i \) and the scatter of \( x_i \) is \( S \), then the scatter of the projected points \( y_i \) is \( W^T S W \).
PCA & Fisher’s Linear Discriminant

- **PCA (Eigenfaces)**
  \[ W_{PCA} = \arg \max_W \left| W^T S_T W \right| \]
  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
Computing the Fisher Projection Matrix

\[ W_{opt} = \arg \max_W \frac{W^T S_B W}{W^T S_W W} \]

\[ = [w_1 \ w_2 \ \cdots \ w_m] \tag{4} \]

where \( \{w_i \mid i = 1, 2, \ldots, m\} \) is the set of generalized eigenvectors of \( S_B \) and \( S_W \) corresponding to the \( m \) largest generalized eigenvalues \( \{\lambda_i \mid i = 1, 2, \ldots, m\} \), i.e.,

\[ S_B w_i = \lambda_i S_W w_i, \quad i = 1, 2, \ldots, m \]

• There are at most \( c-1 \) non-zero generalized Eigenvalues, so \( m \leq c-1 \)
Fisherfaces

\[ W = W_{fld} W_{PCA} \]

\[ W_{PCA} = \arg \max_W \left| W^T S_T W \right| \]

\[ W_{fld} = \arg \max_W \frac{W^T W_{PCA} S_B W_{PCA} W}{W^T W_{PCA} S_W W_{PCA} W} \]

- 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.
PCA vs. FLD

![Chart comparing PCA and FLD](chart.png)
Harvard Face Database

- 10 individuals
- 66 images per person
- Train on 6 images at 15°
- Test on remaining images
Recognition Results: Lighting Extrapolation

![Graph showing error rates for different lighting conditions and methods: Correlation, Eigenfaces, Eigenfaces (w/o 1st 3), Fisherface. The x-axis represents light direction (0-15 degrees, 30 degrees, 45 degrees), and the y-axis represents error rate.]
Next Lecture

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