Recognition III

Introduction to Computer Vision
CSE 152
Lecture 19

Announcements

- HW 4 due Friday
- Final Exam: Tuesday, 6/7 at 8:00-11:00

CSE Peer Mentoring Program
seeking volunteers

- Was your first quarter at UCSD hard?
- Would you like to help others through this time?
- We are seeking volunteers for a new peer mentoring program
  - Each mentor will work with a few new majors
    - just a couple hours per week
  - Mentors will be advised by CSE graduate students
  - Mentorship will look great on your resume
- Visit this URL to fill out a short form, and we'll contact you over the summer:
  http://goo.gl/xLAAj
- Questions? Contact Bill Griswold: wgg@cs.ucsd.edu

Object Recognition: The Problem

Given: A database D of "known" objects and an image I:

1. Determine which (if any) objects in D appear in I
2. Determine the pose (rotation and translation) of the object

Sketch of a Pattern Recognition Architecture

Example: Face Detection

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

Window → Classifier → Face → Non-face
The Space of Images

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

Appearance-based (View-based)

- Face Space:
  - A set of face images construct a face space in \( \mathbb{R}^n \).
  - Appearance-based methods analyze the distributions of individual faces in face space.

Some questions:
1. How are images of an individual, under all conditions, distributed in this space?
2. How are the images of all individuals distributed in this space?

Nearest Neighbor Classifier

\( \{ R_j \} \) are set of training images.

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

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:
  1. The zero 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 \( W \): \( \mathbf{u} + \mathbf{v} \in W \)
  3. If \( \mathbf{u} \) is an element of \( W \) and \( c \) is a scalar from \( \mathbb{K} \), then the scalar product \( c \mathbf{u} \in W \)

A \( d \)-dimensional subspace is spanned by \( d \) linearly independent vectors.
It is spanned by a \( d \)-dimensional orthogonal basis.

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

Linear Subspaces & Linear Projection

- 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
  \[ \mathbf{y} = \mathbf{Wx} \]
where \( \mathbf{W} \) is an \( m \times n \) matrix.
- Recognition is performed in \( \mathbb{R}^m \) using, for example, nearest neighbor.
- How do we choose a good \( \mathbf{W} \)?

Linear Subspaces & Recognition

1. Approximate all training images as a single linear subspace (Eigenfaces).
2. Represent lighting variation w/o shadowing for a single individual as a 3-D linear subspace. A collection of \( n \) individuals is modeled as an \( n \) 3-D linear subspaces.
3. Represent lighting variation w/ shadowing for a single individual as a 9-D linear subspace. A collection of \( n \) individuals is modeled as an \( n \) 9-D linear subspaces.
4. Project all training images to a single subspace that enhances discriminability (Fisherfaces).
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

\[
\mathbf{\mu} = \frac{1}{n} \sum \mathbf{x}_i, \\
\Sigma = \frac{1}{n-1} \sum (\mathbf{x}_i - \mathbf{\mu})(\mathbf{x}_i - \mathbf{\mu})^T
\]

The unit eigenvectors of \( \Sigma \) — which we write as \( \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k \), 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 preserve the most variance.

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

**Eigenfaces**

- **Modeling**
  1. Given a collection of \( n \) labeled training images,
  2. Compute mean image and covariance matrix.
  3. Compute \( k \) Eigenvectors (note that these are images) of covariance matrix corresponding to \( k \) largest Eigenvalues. (Or perform using SVD!!)
  4. Project the training images to the \( k \)-dimensional Eigenspace.

- **Recognition**
  1. Given a test image, project vectorized image to Eigenspace.
  2. Perform classification to the projected training images.

**Distance to Subspace**

- 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

\[
\mathbf{y} = \mathbf{Wx}
\]

- From \( \mathbf{y} \in \mathbb{R}^m \), the reconstruction of the point in \( \mathbb{R}^n \) is \( \mathbf{W}^T\mathbf{y} = \mathbf{W}^T\mathbf{Wx} \)

- The error of the reconstruction, or the distance from \( \mathbf{x} \) to the subspace spanned by \( \mathbf{W} \) is:

\[
||\mathbf{x} - \mathbf{W}^T\mathbf{Wx}||
\]

**3-D Linear subspace**

*The set of images of a Lambertian surface with no shadowing is a subset of 3-D linear subspace.*

\[
\mathbb{L} = \{ \mathbf{x} \mid \mathbf{x} = \mathbf{ Bs}, \mathbf{V s} \in \mathbb{R}^3 \}
\]

where \( \mathbf{B} \) is a \( n \) by \( 3 \)

**Illumination Variability**

- How does the set of images of an object in fixed pose vary as lighting changes?
- How can we recognize people across all lighting conditions without having to see the person every way?
Distance to Linear Subspace

- From a collection of images of person $i$ under variable lighting, compute basis $B_i$ images of a 3-D linear subspace using PCA or SVD.
- Recognize individual by finding subspace with shortest distance from a test image to the subspace. 
  $$d_i = \| x - B_i B_i^T x \|$$

What about shadows?

- The set of images of a Lambertian surface with no shadowing is a subset of 3-D linear subspace. [Moses 93], [Nayar, Murase 96], [Shashua 97]
- The set of images of a convex Lambertian surfaces is a convex cone in the spaces of images. [Belhumeur, Kriegman 1998]
- The set of images of a convex Lambertian surfaces is well-approximated by a 9-D linear subspace. [Basri, Jacobs 2001], [Ramamoorthi, Hanrahan 2001]
- The 9-D linear subspace approximating the set of images of a convex Lambertian surfaces is approximately spanned by nine images taken under 9 well-chosen light sources. [Lee, Ho, Kriegman 2005]

Acquiring Subspace for Recognition

Lee, Ho, Kriegman, PAMI 2005

- Can we find a way to gather real images that span a good subspace for recognition

An important footnote:

We don't really implement PCA by constructing a covariance matrix!

Why?
1. How big is $\Sigma$?
   - $n \times n$ where $n$ is the number of pixels in an image!!
2. You only need the first $k$ Eigenvectors

Singular Value Decomposition

- Any $m \times n$ matrix $A$ may be factored such that 
  $$A = U \Sigma V^T$$
  $[m \times n] = [m \times m][m \times n][n \times n]$
- $U$: $m \times m$, orthogonal matrix
  - Columns of $U$ are the eigenvectors of $AA^T$
- $V$: $n \times n$, orthogonal matrix,
  - columns are the eigenvectors of $A^TA$
- $\Sigma$: $m \times n$, diagonal with non-negative entries ($\sigma_1$, $\sigma_2$, ..., $\sigma_s$) 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 $A^TA$ & Columns of $U$ are corresponding Eigenvectors!!
  - Result of SVD algorithm: $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_s$

SVD Properties

- In Matlab $[u \ s \ v] = svd(A)$, and you can verify that: $A = u^* \Sigma^* v$
- $r=\text{Rank}(A) = \# \text{ of non-zero singular values}$.
- $U$, $V$ 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 $A$
  - Last $n-r$ columns of $V$: 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.
Thin 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:
  \[ \Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s) \] with \( s = \min(m, n) \), and lower matrix is \((n-m \times m)\) of zeros.
- Alternatively, you can write:
  \[ A = U' \Sigma' V^T \]

In Matlab, thin SVD is:
\[ [U S V] = \text{svds}(A) \]
This is what you should use!!

Performing PCA with SVD

- Singular values of A are the square roots of eigenvalues of both \( AA^T \) and \( A^T A \) & Columns of \( U \) are corresponding Eigenvectors
- And \( \sum_{i=1}^{n} a_i \alpha_i = [a_1 a_2 \ldots a_1 a_2 \ldots a_n] = AA^T \)
- Covariance matrix is:
  \[ \Sigma = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T \]
- So, ignoring \( 1/n \) subtract mean \( \mu \) from each input image, create data matrix, and perform thin SVD on the data matrix.

Alternative projections

Fisherfaces: Class specific linear projection


- An n-pixel image \( x \in \mathbb{R}^n \) can be projected to a low-dimensional feature space \( y \in \mathbb{R}^m \) by
  \[ y = Wx \]
where \( W \) is an \( n \times m \) matrix.
- Recognition is performed using nearest neighbor in \( \mathbb{R}^m \).
- How do we choose a good \( W \)?

PCA & Fisher’s Linear Discriminant

- Between-class scatter \( S_b = \sum_{c} \sum_{x \in c} (x - \mu_c)(x - \mu_c)^T \)
- Within-class scatter \( S_w = \sum_{c} \sum_{x \in c} (x - \mu_c)(x - \mu_c)^T \)
- Total scatter \( S_T = \sum_{c} \sum_{x \in c} (x - \mu_c)(x - \mu_c)^T + S_b \)
- Where
  - \( c \) is the number of classes
  - \( \mu_c \) is the mean of class \( y_c \)
  - \( |X_c| \) is number of samples of \( y_c \)
- If the data points are projected by \( y = Wx \) and scatter of points is \( S \), then the scatter of the projected points is \( W^T S W \)

PCA & Fisher’s Linear Discriminant

- PCA (Eigenfaces)
  \[ W_{PCA} = \arg \max_w |W^T S_w W| \]
  Maximizes projected total scatter
- Fisher’s Linear Discriminant
  \[ W_{FLD} = \arg \max_w |W^T S_b W| \]
  Minimizes ratio of between-class to projected within-class scatter
Computing the Fisher Projection Matrix

\[ W_{w_i} = \arg \max_{w} \frac{w^{T}S_{w}w}{w^{T}S_{b}w} \]

\[ w_{1}, w_{2}, \ldots, w_{m} \]

where \( \{w_{i}\} \) is the set of generalized eigenvectors of \( S_{b} \) and \( S_{w} \) corresponding to the \( m \) largest generalized eigenvalues \( \{\lambda_{i}\} \) of \( S_{b} \), i.e.,

\[ S_{w}w_{i} = \lambda_{i}S_{b}w_{i}, \quad i = 1, 2, \ldots, m \]

- The \( w_{i} \) are orthonormal
- There are at most \( c-1 \) non-zero generalized Eigenvalues, so \( m \leq c-1 \)
- Can be computed with \texttt{eig} in Matlab

Fisherfaces

- 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.

Harvard Face Database

- 10 individuals
- 66 images per person
- Train on 6 images at 15°
- Test on remaining images

Support Vector Machines

- Optimal separating hyperplane maximizes \( \frac{1}{w^{T}w} \)

\[ \text{Minimize} \quad \frac{1}{2}w^{T}w \]

\[ \text{Subject to} \quad y_{i}(w \cdot x_{i} + b) \geq 1, \quad i = 1, 2, \ldots, N \]

Optimal separating hyperplane (OSH)

Variability:
- Camera position
- Illumination
- Internal parameters

Within-class variations
Appearance manifold approach
- for every object
  1. sample the set of viewing conditions
  2. Crop & scale images to standard size
  3. Use as feature vector
- apply a PCA over all the images
- keep the dominant PCs
- Set of views for one object is represented as a manifold in the projected space
- Recognition: What is nearest manifold for a given test image?

An example: input images

An example: basis images

An example: surfaces of first 3 coefficients

Parameterized Eigenspace

Employ spatial relations

Figure from “Local grayvalue invariants for image retrieval,” by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997; copyright 1997, IEEE.
Finding faces using relations

• Strategy:
  – Face is eyes, nose, mouth, etc. with appropriate relations between them
  – Build a specialised detector for each of these (template matching) and look for groups with the right internal structure
  – Once we've found enough of a face, there is little uncertainty about where the other bits could be

Finding faces using relations

• Strategy: compare

$P(\text{true face at } \mathbf{X}_{\text{true}} = \mathbf{x}_1, \mathbf{X}_2 = \mathbf{x}_2, \mathbf{X}_3 = \mathbf{x}_3, \mathbf{X}_4 = \mathbf{x}_4, \text{all other responses})$

with

$P(\text{no face})\mathbf{x}_1, \mathbf{X}_2 = \mathbf{x}_2, \mathbf{X}_3 = \mathbf{x}_3, \mathbf{X}_4 = \mathbf{x}_4, \text{all other responses})$

Notice that once some facial features have been found, the position of the rest is quite strongly constrained.