Announcements

• Homework 3 is due November 23, 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

Example: Face Detection

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

Feature Space

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

The Space of Images

- We will treat an n-pixel image as a point in an n-dimensional space, \( x \in \mathbb{R}^n \).
- Each pixel value is a coordinate of \( x \).

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)
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

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

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 a \( k \) 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 \( W \)?

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

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

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 \( y_2 \)
Eigenfaces: Principal Component Analysis (PCA)

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

\[
\mu = \frac{1}{n} \sum x_i
\]

\[
E = \frac{1}{n-1} \sum (x_i - \mu)(x_i - \mu)^T
\]

The unit eigenvectors of \( E \) — which we write as \( \pi_1, \pi_2, \ldots, \pi_n \), where the order is given by the size of the eigenvalue and \( \pi_1 \) has the largest eigenvalue — give a set of features with the following properties:

- They are independent.
- Projection onto the basis \( \{\pi_1, \ldots, \pi_n\} \) gives the \( d \)-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.

Eigendecomposition of covariance matrix.

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

SVD Properties

- In Matlab \( [U, S, V] = \text{svd}(A) \), and you can verify that: \( A = U \Sigma V^T \)
- \( r = \text{rank}(A) \) \( \neq 0 \) of non-zero singular values.
- \( U, V \) give orthonormal bases for the subspaces of \( A \):
  - First \( r \) columns of \( U \): Column space of \( A \)
  - Last \( m - r \) columns of \( U \): Left nullspace of \( A \)
  - First \( r \) columns of \( V \): Row space of \( A \)
  - Last \( n - r \) 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
  \[
  \sum a_i a_i^T = \begin{bmatrix} a_1 & a_2 & \cdots & a_d \end{bmatrix} \begin{bmatrix} a_1 \ a_2 \ \cdots \ a_d \end{bmatrix}^T = AA^T
  \]
- Covariance matrix is:
  \[
  \Sigma = \frac{1}{n-1} \sum (x_i - \bar{x})(x_i - \bar{x})^T
  \]
- 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. \( D = [x_1 - \mu \ | \ x_2 - \mu \ | \ \ldots \ | \ x_n - \mu] \)

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)
  \[
  \Sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 & \cdots & \sigma_s \end{bmatrix}
  \]
  \( \Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s) \) with \( s = \text{min}(m,n) \), and lower matrix is \( (n-m \times m) \) of zeros.
- Alternatively, you can write:
  \[
  A = U \Sigma V^T
  \]
- In Matlab, economy SVD is \([U, S, V] = \text{svd}(A, \text{econ})\)

PCA Example

First Principal Component
Direction of Maximum Variance

Mean

\( v_1 \)

\( v_2 \)
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 the \( k \) Eigenvectors of the covariance matrix corresponding to the \( k \) largest Eigenvalues and form matrix \( W = [u_1, u_2, ..., u_k] \) (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 the subspace.

Eigenfaces: Training Images

Eigenfaces

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 \( x \in \mathbb{R}^d \) can be projected to a low-dimensional feature space \( y \in \mathbb{R}^k \) by

\[
y = Wx
\]

where \( W \) is a \( k \times d \) matrix

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

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

\[
W_{FLD} = \arg \max_{W} \frac{\|W^T S_B W \|}{\|W^T S_W W \|}
\]

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

If the data points \( x_i \) are projected by \( y_i = Wx_i \), and the scatter of \( x_i \) is \( S_i \), then the scatter of the projected points \( y_i \) is \( WSW \).

Computing the Fisher Projection Matrix

\[
W_{FLD} = \arg \max_{W} \frac{\|W^T S_B W \|}{\|W^T S_W W \|}
\]

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

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

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

Fisherfaces

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

\[
W_{PCA} = \arg \max_{W} \|W^T S_B W \|
\]

\[
W_{FLD} = \arg \max_{W} \frac{\|W^T W_{PCA} S_B W_{PCA} W \|}{\|W^T W_{PCA} S_W W_{PCA} W \|}
\]

• 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

- Between-class scatter

\[
S_B = \sum_{i} \sum_{x \in \mathbb{C}_i} (x - \mu_i)(x - \mu_i)^T
\]

- Within-class scatter

\[
S_W = \sum_{i} \sum_{x \in \mathbb{C}_i} (x - \mu)(x - \mu)^T
\]

- Total scatter

\[
S_T = \sum_{i} \sum_{x \in \mathbb{C}_i} (x - \mu_i)(x - \mu_i)^T + S_B
\]

Where
- \( c \) is the number of classes
- \( \mu_i \) is the mean of class \( \mathbb{C}_i \)
- \( | \mathbb{C}_i | \) is number of samples of \( \mathbb{C}_i \).
Harvard Face Database

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

Recognition Results: Lighting Extrapolation

Next Lecture

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