Announcements

- Homework 3 is due May 18, 11:59 PM
- Homework 4 will be assigned this week
- Reading:
  - Chapter 15: Learning to Classify
  - Chapter 16: Classifying Images
  - Chapter 17: Detecting Objects in Images

A Rough Recognition Spectrum

Appearance-Based Recognition

Increasing Generality

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

Classifier

Feature Space

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

The Space of Images

- We will treat an d-pixel image as a point in an d-dimensional space, \( x \in \mathbb{R}^d \).
- 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)

Feature Space

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

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

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 \( W \)?

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
Eigenfaces: Principal Component Analysis (PCA)

Assume we have a set of n feature vectors \( \mathbf{x}_i \) (i = 1, ..., n) in \( \mathbb{R}^m \). Write
\[
\mu = \frac{1}{n} \sum \mathbf{x}_i \\
\mathbf{E} = \frac{1}{n-1} \sum (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T
\]
The unit eigenvectors of \( \Sigma \) — which we write as \( \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \), where the order is given by the size of the eigenvalue and \( \lambda_1 \) has the largest eigenvalue — give a set of features with the following properties:
- They are independent.
- Projection onto the basis \( \{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \) gives the \( i \)-dimensional set of linear features that preserves the most variance.

Algorithm 23.5: Principal component analysis identifies a collection of linear features that are independent, and captures as much variance as possible from a dataset. Eigen decomposition of covariance matrix.

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

---

**SVD Properties**

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

---

**Performing PCA with SVD**

- Singular values of \( \mathbf{U} \) are the square roots of eigenvalues of \( \mathbf{A} \mathbf{A}^T \) (and \( \mathbf{A}^T \mathbf{A} \))
- Columns of \( \mathbf{U} \) are corresponding Eigenvectors of \( \mathbf{A} \mathbf{A}^T \)
- And \( \sum a_i^2 = [a_1 \ a_2 \ldots \ a_1 \ a_2 \ldots a_1] \mathbf{A}^T \)
- Covariance matrix is:
\[
\Sigma = \frac{1}{n-1} \sum (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T
\]
- 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. \( \mathbf{D} = [x_1 - \mu \ | \ x_2 - \mu \ | \ldots \ x_n - \mu] \)

---

**Economy SVD**

- Any \( m \) by \( n \) matrix \( \mathbf{A} \) may be factored such that
\[
\mathbf{A} = \mathbf{U} \mathbf{S} \mathbf{V}^T
\]
- If \( m > n \), then one can view \( \Sigma \) as: (i.e., more pixels than images)
\[
\Sigma = \begin{bmatrix}
\Sigma^T \\
0
\end{bmatrix}
\]
- Where \( \Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s) \) with \( s = \min(m,n) \), and lower matrix is \( (n-m \times m) \) of zeros.
- Alternatively, you can write:
\[
\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T
\]
- In Matlab, economy SVD is: \( [\mathbf{U} \; \mathbf{S} \; \mathbf{V}] = \text{svd}(\mathbf{A}, \text{econ}) \)

---

**PCA Example**

First Principal Component
Direction of Maximum Variance
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 \( W = [u_1, u_2, \ldots, u_k] \) (Or perform using SVD)
   - Note that the Eigenvectors are images
4. Project the training images to the \( k \)-dimensional Eigenspace. \( y_i = W x_i \)

Recognition
1. Given a test image \( x \), project the vectorized image to the Eigenspace by \( y = W x \)
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 the sum of the squared distances from each datapoint to the the subspace.

Eigenfaces: Training Images

[ Turk, Pentland 91]

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)

**Maximizes projected total scatter**

\[
W_{PC} = \arg \max_{W} \|W^T S_W W\|
\]

### Fisher’s Linear Discriminant

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

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

### Computing the Fisher Projection Matrix

\[
W_{FLD} = \arg \max_{W} \frac{\|W^T S_B W\|}{\|W^T S_W W\|} = \left[ w_1, w_2, \ldots, w_m \right]
\]

\( W_{FLD} \) is the set of generalized eigenvectors of \( S_W \) and \( S_B \) corresponding to the \( m \) largest generalized eigenvalues \( \lambda_i \), \( i = 1, 2, \ldots, m \), 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

- 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
  \[
  S_B = \sum_{i=1}^{c} \sum_{x \in \mathcal{C}_i} (x - \mu_i)(x - \mu_i)^T
  \]
- Within-class scatter
  \[
  S_W = \sum_{i=1}^{c} \sum_{x \in \mathcal{C}_i} (x - \mu_i)(x - \mu_i)^T
  \]
- Total scatter
  \[
  S_T = \sum_{i=1}^{c} \sum_{x \in \mathcal{C}_i} (x - \mu)(x - \mu)^T = S_W + S_B
  \]

- Where
  - \( c \) is the number of classes
  - \( \mu_i \) is the mean of class \( \mathcal{C}_i \)
  - \( | \mathcal{C}_i | \) is number of samples of \( \mathcal{C}_i \)
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