Announcements

- HW 3 due Today.
- HW4 (face recognition) has been posted.

Recognition II

Introduction to Computer Vision
CSE 152
Lecture 18

Recognition

Given a database of objects and an image determine what, if any of the objects are present in the image.

Recognition Challenges

- Within-class variability
  - Different objects within the class have different shapes or different material characteristics
  - Deformable
  - Articulated
  - Compositional
- Pose variability:
  - 2-D Image transformation (translation, rotation, scale)
  - 3-D Pose Variability (perspective, orthographic projection)
- Lighting
  - Direction (multiple sources & type)
  - Color
  - Shadows
- Occlusion – partial
- Clutter in background -> false positives

Image parsing

- sky
- mountain
- building
- tree
- banner
- street lamp
- market
- people

Sketch of a Pattern Recognition Architecture

Image (window) --- Feature Extraction --- Feature Vector --- Classification --- Object Identity
Sliding window approaches

The Space of Images

- We will treat a 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

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

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 \( W \); \( a\mathbf{u} + b\mathbf{v} \in W \)
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.

Linear Subspaces & Linear Projection

- A \( d \)-pixel image \( \mathbf{x} \in \mathbb{R}^d \) can be projected to a low-dimensional feature space \( \mathbf{y} \in \mathbb{R}^k \) by
  \[ \mathbf{y} = W\mathbf{x} \]
  where \( W \) is an \( k \times d \) 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 1: Represent lighting variation w/o shadowing for a single individual as a 3-D linear subspace. \( n \) individuals are modeled as \( n \) 3-D 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 $x_i$ ($i=1, \ldots, n$) in $\mathbb{R}^d$. Write

$$
\mu = \frac{1}{n} \sum x_i,
$$

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

The unit eigenvectors of $\Sigma$ — which we write as $v_1, v_2, \ldots, v_k$, 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 $\{v_1, \ldots, v_k\}$ gives the $d$-dimensional set of linear features that preserve the most variance.

Algorithm 23.5: Principal components analysis identifies a collection of linear features that are independent, and capture as much variance as possible from a dataset.

Eigenfaces

- Modeling
  1. Given a collection of $n$ training images $x_i$, represent each one as $d$-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 $W = [v_1, v_2, \ldots, v_k]$ (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 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]
Basis Images for Variable Lighting

Distance to Linear Subspace

- An $n$-pixel image $x \in \mathbb{R}^d$ can be projected to a low-dimensional feature space $y \in \mathbb{R}^k$ by
  \[ y = Wx \]
- From $y \in \mathbb{R}^k$, the reconstruction of the point in $\mathbb{R}^d$ is $W^Ty = W^TWx$
- The error of the reconstruction, or the distance from $x$ to the subspace spanned by $W$ is:
  \[ ||x - W^TWx|| \]

Distance to Affine Subspace

- Represented by mean vector $\mu$ and basis images $W$
- An $n$-pixel image $x \in \mathbb{R}^d$ can be projected to a low-dimensional feature space $y \in \mathbb{R}^k$ by
  \[ y = W(x - \mu) \]
- From $y \in \mathbb{R}^k$, the reconstruction of the point in $\mathbb{R}^d$ is $W^Ty + \mu = W^TW(x - \mu) + \mu$
- The error of the reconstruction, or the distance from $x$ to the affine is:
  \[ ||x - W^TW(x - \mu) - \mu|| = ||(I - W^TW)(x - \mu)|| \]

An important footnote:
We don’t really implement PCA by constructing a covariance matrix!

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

Singular Value 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 $A^TA$
- $\Sigma$: $m$ by $n$, diagonal with non-negative entries ($\sigma_1, \sigma_2, \ldots, \sigma_s$) with $s=\text{min}(m,n)$ are called the called the singular values. SVD algorithm produces sorted singular values: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s$

Important property
  - Singular values are the square roots of Eigenvalues of both $AA^T$ and $A^TA$ & Columns of $U$ are corresponding Eigenvectors!!

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 $V$: Nullspace of $A$
- For some $d$ where $d < r$, 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 both $AA^T$ and $A^TA$ & Columns of $U$ are corresponding Eigenvectors
- And $\sum a_i a_i^T = [a_1 \ a_2 \ ... \ a_n] - AA^T$
- Covariance matrix is:

\[ \Sigma = \frac{1}{n} \sum_{i=1}^{n} (\bar{x}_i - \bar{\mu})(\bar{x}_i - \bar{\mu})^T \]

- So, ignoring $1/n$ subtract mean $\mu$ from each input image, create a $d \times n$ data matrix, and perform thin SVD on the data matrix. $D = \{x_1 - \mu \ | \ x_2 - \mu \ | \ ... \ | \ x_n - \mu \}$

Thin SVD

- Any $m \times n$ matrix $A$ may be factored such that $A = U \Sigma V^T$
- If $m > n$, then one can view $\Sigma$ as: (i.e., more pixels than images)

\[ \Sigma \rightarrow \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} \]

- Where $\Sigma = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_s)$ with $s = \min(m,n)$, and lower matrix is $(m-n \times m)$ 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!!

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?

3-D Linear subspace

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

$[\text{Moses 93}, \text{Nayar, Murase 96}, \text{Shashua 97}]$

$L = \{x \ | \ x = Bs, \forall s \in \mathbb{R}^3 \}$

where $B$ is a $n \times 3$

How do you construct the 3-D subspace?

$$
\begin{bmatrix}
\begin{array}{cccc}
X_1 & X_2 & X_3 & X_4 & X_5 \\
\end{array}
\end{bmatrix}
\rightarrow
\begin{bmatrix}
\begin{array}{c}
X_1 \\
X_2 \\
X_3 \\
X_4 \\
X_5
\end{array}
\end{bmatrix}
$$

With more than three images, perform least squares estimate of $B$ using PCA or Singular Value Decomposition (SVD)

Face Basis

Original Images

Basis Images

Span B
**Distance to Linear Subspace**

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

- From $y \in \mathbb{R}^k$, the reconstruction of the point in $\mathbb{R}^d$ is $W^T y = W^T Wx$

- The error of the reconstruction, or the distance from $x$ to the subspace spanned by $W$ is:
  \[ \| x - W^T Wx \| \]

**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 an $k \times d$ matrix.

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

- How do we choose a good $W$?

**PCA & Fisher’s Linear Discriminant**

- Between-class scatter
  \[ S_B = \sum_i |x_i - \mu_i \rangle \langle x_i - \mu_i | \]

- Within-class scatter
  \[ S_W = \sum_i \sum_{x_j \in C_i} (x_j - \mu_i)(x_j - \mu_i)^T \]

- Total scatter
  \[ S_T = \sum_i \sum_{x_j \in C_i} (x_j - \mu_i)(x_j - \mu_i)^T + S_B \]

- Where
  - $c$ is the number of classes
  - $\mu_i$ is the mean of class $y_i$
  - $|y_i|$ is number of samples of $y_i$

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 $W^T S W$.
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**
  \[ W_{FLD} = \arg \max_w \frac{W^T S_b W}{W^T S_w W} \]
  Maximizes ratio of projected between-class to projected within-class scatter

**Fisherfaces**

\[ W = W_{FLD} W_{PCA} \]
\[ W_{PCA} = \arg \max_w W^T S_w W \]
\[ W_{FLD} = \arg \max_w \frac{W^T S_b S_w W}{W^T S_w W} \]

- 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.

**Computing the Fisher Projection Matrix**

\[ W_{opt} = \arg \max_w \frac{W^T S_b W}{W^T S_w W} \]
\[ = [w_1 \quad w_2 \ldots \quad w_m] \]

where \( \{ w_i \}_{i=1, 2, \ldots, m} \) 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.,
\[ S_w w_i = \lambda_i S_b w_i \quad i = 1, 2, \ldots, m \]

- 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

**Harvard Face Database**

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

**Recognition Results: Lighting Extrapolation**