Recognition (Part 2)

Introduction to Computer Vision
CSE 152
Lecture 17

Announcements

• Homework 3 is due May 29, 11:59 PM
• Homework 4 is due next week
• Final exam will be a take home exam

• Reading:
  – Section 14.2.1 Eigenfaces
  – Section 14.4.1 Bag of words

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 $W$?

PCA & Fisher’s Linear Discriminant

• Between-class scatter

$$S_B = \sum_{i=1}^{c} |\mathbf{X}_i|\mathbf{S}_i$$

• Within-class scatter

$$S_W = \sum_{i=1}^{c} \sum_{x_i \in \mathbf{X}_i} (x_i - \mu_i)(x_i - \mu_i)^T$$

• Total scatter

$$S_T = \sum_{i=1}^{c} \sum_{x_i \in \mathbf{X}_i} (x_i - \mu_i)(x_i - \mu_i)^T = S_B + S_W$$

• Where

  – $c$ is the number of classes
  – $\mu_i$ is the mean of class $\mathbf{X}_i$
  – $|\mathbf{X}_i|$ is number of samples of $\mathbf{X}_i$

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 $W S W^T$.

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

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

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

The unit eigenvectors of $\Sigma$ — which we write as $v_1, v_2, \ldots, v_d$ 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 $\{v_1, \ldots, v_d\}$ 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 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.
Performing PCA with SVD

• Singular values of $A$ are the square roots of eigenvalues of both $A^TA$ and $AA^T$
• Columns of $U$ are corresponding Eigenvectors of $A^TA$
• And $\sum_{i=1}^{n} a_i x_i = [a_1 \ a_2 \ \cdots \ a_i \ \cdots \ a_n] = A^T$
• Covariance matrix is:

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

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

Computing the Fisher Projection Matrix

$$W_{opt} = \arg \max_w \frac{w^T S_w w}{w^T S_b w}$$

$$= [w_1 \ w_2 \ \cdots \ w_m]$$

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 $eig$ in Matlab

Fisherfaces

$$W = W_{opt} W_{PC}$$

$$W_{PC} = \arg \max_W \frac{W^T S_b W}{W^T S_w W}$$

$$W_{opt} = \arg \max_W \frac{W^T S_b W}{W^T S_w W}$$

• 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.

Appearance-Based Vision for Instances Level Recognition:
A Pattern Classification Viewpoint
1. Feature Space + Nearest Neighbor
2. Dimensionality Reduction
3. Bayesian Classification
4. Appearance Manifolds

Sketch of a Pattern Recognition Architecture

Bayesian Classification
Example: Sorting

- “Sorting incoming Fish on a conveyor according to species using optical sensing”

Fish

\[ x^T = [x_1, x_2] \]

![](fish.png)

Adopt the lightness and add the width of the fish
Basic ideas in classifiers

- Loss
  - some errors may be more expensive than others
    - e.g. a fatal disease that is easily cured by a cheap medicine with no side-effects -> false positives in diagnosis are better than false negatives
  - We discuss two class classification: \( L(1 \rightarrow 2) \) is the loss caused by calling 1 a 2
- Total risk of using classifier \( s \)

\[
R(s) = Pr(1 \rightarrow 2 | \text{using } s) L(1 \rightarrow 2) + Pr(2 \rightarrow 1 | \text{using } s) L(2 \rightarrow 1)
\]

Some loss may be inevitable: the minimum risk (shaded area) is called the Bayes risk

Finding a decision boundary is not the same as modelling a conditional density.

Plug-in classifiers

- Assume that class conditional distributions \( P(x|\omega_i) \) have some parametric form - now estimate the parameters from the data.
  - Common:
    - assume a normal distribution with shared covariance, different means; use usual estimates
    - Normal distribution but with different covariances
  - Issue: parameter estimates that are “good” may not give optimal classifiers.
Example: Finding skin

- Skin has a very small range of (intensity independent) colors, and little texture
  - Compute an intensity-independent color measure, check if color is in this range, check if there is little texture (median filter)
  - See this as a classifier - we can set up the tests by hand, or learn them.
  - get class conditional densities (histograms), priors from data (counting)

- Classifier
  - if $p($skin$|r) > \theta$, classify as skin
  - if $p($skin$|r) < \theta$, classify as not skin
  - if $p($skin$|r) = \theta$, choose classes uniformly at random

Receiver Operating Curve

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?

Limitations of these approaches

- Object must be segmented from background (How would one do this in non-trivial situations?)
- Occlusion?
- The variability (dimension) in images is large, so is sampling feasible?
- How can one generalize to classes of objects?
Appearance-Based Vision: Lessons

Strengths

• Posing the recognition metric in the image space rather than a derived representation is more powerful than expected.
• Modeling objects from many images is not unreasonable given hardware developments.
• The data (images) may provide a better representations than abstractions for many tasks.

Weaknesses

• Segmentation or object detection is still an issue.
• To train the method, objects have to be observed under a wide range of conditions (e.g. pose, lighting, shape deformation).
• Limited power to extrapolate or generalize (abstract) to novel conditions.

Bag-of-features models

Object → Bag of ‘words’

Origin 1: Texture recognition

• Texture is characterized by the repetition of basic elements or textons.
• For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters.

Origin 2: Bag-of-words models


Which US President?
Franklin D. Roosevelt, John F. Kennedy, George W. Bush

Bag-of-features steps
1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies (histogram) of “visual words”
5. Recognition using histograms as input to classifier
Feature extraction

- Regular grid or interest regions