Announcements

- Project 4 has been released, due Friday, November 16 at 11:59pm
  – Please get started early!

- Quiz on Friday
Linear subspaces

Classification can be expensive
- Must either search (e.g., nearest neighbors) or store large PDF’s

Suppose the data points are arranged as above
- Idea—fit a line, classifier measures distance to line

What does the $v_2$ coordinate measure?
- distance to line
- use it for classification—near 0 for orange pts

What does the $v_1$ coordinate measure?
- position along line
- use it to specify which orange point it is

Dimensionality reduction

We can represent the orange points with only their $v_1$ coordinates
- since $v_2$ coordinates are all essentially 0
- This makes it much cheaper to store and compare points
- A bigger deal for higher dimensional problems
Linear subspaces

Consider the variation along direction \( \mathbf{v} \) among all of the orange points:

\[
\text{var}(\mathbf{v}) = \sum_{\text{orange point} \ x} ||(\mathbf{x} - \overline{\mathbf{x}})^T \cdot \mathbf{v}||^2
\]

What unit vector \( \mathbf{v} \) minimizes var?
\[
\mathbf{v}_2 = \min_{\mathbf{v}} \{ \text{var}(\mathbf{v}) \}
\]

What unit vector \( \mathbf{v} \) maximizes var?
\[
\mathbf{v}_1 = \max_{\mathbf{v}} \{ \text{var}(\mathbf{v}) \}
\]

\[
\text{var}(\mathbf{v}) = \sum_{\mathbf{x}} ||(\mathbf{x} - \overline{\mathbf{x}})^T \cdot \mathbf{v}||^2
\]
\[
= \sum_{\mathbf{x}} \mathbf{v}^T (\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^T \mathbf{v}
\]
\[
= \mathbf{v}^T \left[ \sum_{\mathbf{x}} (\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^T \right] \mathbf{v}
\]
\[
= \mathbf{v}^T \mathbf{A} \mathbf{v} \quad \text{where} \quad \mathbf{A} = \sum_{\mathbf{x}} (\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^T
\]

Solution: \( \mathbf{v}_1 \) is eigenvector of \( \mathbf{A} \) with largest eigenvalue
\( \mathbf{v}_2 \) is eigenvector of \( \mathbf{A} \) with smallest eigenvalue

Principal component analysis

Suppose each data point is N-dimensional

- Same procedure applies:

\[
\text{var}(\mathbf{v}) = \sum_{\mathbf{x}} ||(\mathbf{x} - \overline{\mathbf{x}})^T \cdot \mathbf{v}||
\]
\[
= \mathbf{v}^T \mathbf{A} \mathbf{v} \quad \text{where} \quad \mathbf{A} = \sum_{\mathbf{x}} (\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^T
\]

- The eigenvectors of \( \mathbf{A} \) define a new coordinate system
  - eigenvector with largest eigenvalue captures the most variation among training vectors \( \mathbf{x} \)
  - eigenvector with smallest eigenvalue has least variation

- We can compress the data by only using the top few eigenvectors
  - corresponds to choosing a "linear subspace"
    - represent points on a line, plane, or "hyper-plane"
  - these eigenvectors are known as the principal components
The space of faces

An image is a point in a high dimensional space
- An $N \times M$ intensity image is a point in $\mathbb{R}^{NM}$
- We can define vectors in this space as we did in the 2D case

Dimensionality reduction

The set of faces is a “subspace” of the set of images
- Suppose it is $K$ dimensional
- We can find the best subspace using PCA
- This is like fitting a “hyper-plane” to the set of faces
  - spanned by vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K$
  - any face $\mathbf{x} \approx \bar{x} + a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \ldots + a_k\mathbf{v}_k$
Eigenfaces

PCA extracts the eigenvectors of $A$

- Gives a set of vectors $v_1, v_2, v_3, ...$
- Each one of these vectors is a direction in face space
  - what do these look like?

![Eigenfaces](image)

Projecting onto the eigenfaces

The eigenfaces $v_1, ..., v_K$ span the space of faces

- A face is converted to eigenface coordinates by

$$x ightarrow \left(\frac{(x - \bar{x}) \cdot v_1}{a_1}, \frac{(x - \bar{x}) \cdot v_2}{a_2}, ..., \frac{(x - \bar{x}) \cdot v_K}{a_K}\right)$$

$$x \approx \bar{x} + a_1 v_1 + a_2 v_2 + \ldots + a_K v_K$$

![Projecting onto eigenfaces](image)
Detection and recognition with eigenfaces

Algorithm
1. Process the image database (set of images with labels)
   • Run PCA—compute eigenfaces
   • Calculate the K coefficients for each image
2. Given a new image (to be recognized) $x$, calculate K coefficients
   $$x \rightarrow (a_1, a_2, \ldots, a_K)$$
3. Detect if $x$ is a face
   $$\|x - (\bar{x} + a_1 v_1 + a_2 v_2 + \ldots + a_K v_K)\| < \text{threshold}$$
4. If it is a face, who is it?
   • Find closest labeled face in database
     • nearest-neighbor in K-dimensional space

Choosing the dimension $K$

How many eigenfaces to use?
Look at the decay of the eigenvalues
• the eigenvalue tells you the amount of variance “in the direction” of that eigenface
• ignore eigenfaces with low variance
Issues: metrics

What’s the best way to compare images?
- need to define appropriate features
- depends on goal of recognition task

**exact matching**
complex features work well
(SIFT, MOPS, etc.)

**classification/detection**
simple features work well
(Viola/Jones, etc.)

Metrics

Lots more feature types that we haven’t mentioned
- moments, statistics
  - metrics: Earth mover’s distance, ...
- edges, curves
  - metrics: Hausdorff, shape context, ...
- 3D: surfaces, spin images
  - metrics: chamfer (ICP)
- ...

Issues: feature selection

If all you have is one image: non-maximum suppression, etc.

If you have a training set of images: AdaBoost, etc.

Issues: data modeling

Generative methods
- model the “shape” of each class
  - histograms, PCA, mixtures of Gaussians
  - graphical models (HMM’s, belief networks, etc.)
  - …

Discriminative methods
- model boundaries between classes
  - perceptrons, neural networks
  - support vector machines (SVM’s)
Generative vs. Discriminative

Generative Approach
model individual classes, priors

Discriminative Approach
model posterior directly

from Chris Bishop

Issues: dimensionality

What if your space isn’t flat?
- PCA may not help

Nonlinear methods
LLE, MDS, etc.
Moving forward

• Faces are pretty well-behaved
  – Mostly the same basic shape
  – Lie close to a low-dimensional subspace

• Not all objects are as nice

Different appearance, similar parts