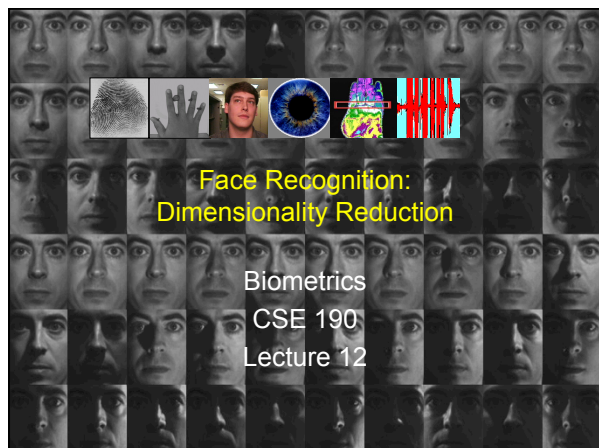


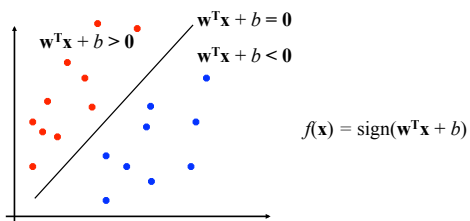


- HW2 due Today

CSE190, Winter 2011

Perceptron Revisited: Linear Separators

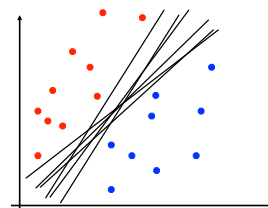
- Binary classification can be viewed as the task of separating classes in feature space:



3

Linear Separators

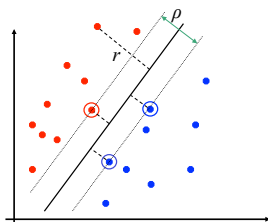
- Which of the linear separators is optimal?



4

Classification Margin

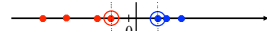
- Distance from example x_i to the separator is $r = \frac{w^T x_i + b}{\|w\|}$
- Training examples closest to the hyperplane are **support vectors**.
- **Margin** ρ of the separator is the distance from the separator to support vectors.



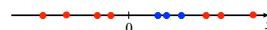
5

Non-linear SVMs

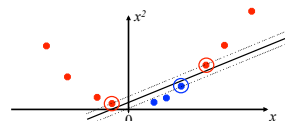
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?



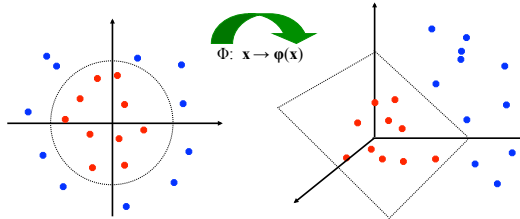
- How about... mapping data to a higher-dimensional space:



6

Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



7

The “Kernel Trick”

- The linear classifier relies on inner product between vectors $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
- If every datapoint is mapped into high-dimensional space via some transformation $\Phi: \mathbf{x} \rightarrow \phi(\mathbf{x})$, the inner product becomes:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$
- A *kernel function* is a function that is equivalent to an inner product in some feature space.
- Example:
 2-dimensional vectors $\mathbf{x} = [x_1 \ x_2]$; let $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2$.
 Need to show that $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2 = 1 + x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 + 2 x_{i1} x_{j1} + 2 x_{i2} x_{j2} =$$

$$= [1 \ x_{i1}^2 \ \sqrt{2} x_{i1} x_{i2} \ x_{i2}^2 \ \sqrt{2} x_{i1} \ \sqrt{2} x_{i2}]^T [1 \ x_{j1}^2 \ \sqrt{2} x_{j1} x_{j2} \ x_{j2}^2 \ \sqrt{2} x_{j1} \ \sqrt{2} x_{j2}] =$$

$$= \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j), \text{ where } \phi(\mathbf{x}) = [1 \ x_1^2 \ \sqrt{2} x_1 x_2 \ x_2^2 \ \sqrt{2} x_1 \ \sqrt{2} x_2]$$
- Thus, a kernel function *implicitly* maps data to a high-dimensional space (without the need to compute each $\phi(\mathbf{x})$ explicitly).

8

Examples of Kernel Functions

- Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
 - Mapping $\Phi: \mathbf{x} \rightarrow \phi(\mathbf{x})$, where $\phi(\mathbf{x})$ is $\mathbf{x}$ itself
- Polynomial of power p : $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$
 - Mapping $\Phi: \mathbf{x} \rightarrow \phi(\mathbf{x})$, where $\phi(\mathbf{x})$ has $\binom{d+p}{p}$ dimensions
- Gaussian (radial-basis function): $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$
 - Mapping $\Phi: \mathbf{x} \rightarrow \phi(\mathbf{x})$, where $\phi(\mathbf{x})$ is *infinite-dimensional*: every point is mapped to a *function* (a Gaussian); combination of functions for support vectors is the separator.
- Higher-dimensional space still has *intrinsic* dimensionality d (the mapping is not *onto*), but linear separators in it correspond to *non-linear* separators in original space.

9

Non-linear SVMs Mathematically

- Dual problem formulation:

Find $\alpha_1 \dots \alpha_n$ such that
 $Q(\mathbf{a}) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ is maximized and
 (1) $\sum \alpha_i y_i = 0$
 (2) $\alpha_i \geq 0$ for all α_i

- The solution is:

$$f(\mathbf{x}) = \sum \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b$$

- Optimization techniques for finding α_i 's remain the same!

10

Face



CSE190, Winter 2011

Face Recognition

- Face is the most common biometric used by humans
- Applications range from static, mug-shot verification to a dynamic, uncontrolled face identification in a cluttered background
- Challenges:
 - automatically locate the face
 - recognize the face from a general view point under different illumination conditions, facial expressions, and aging effects

© Jain, 2004

Authentication vs Identification

- Face Authentication/Verification (1:1 matching)



- Face Identification/recognition (1:N matching)



© Jain, 2004

Applications

- Access Control



www.viisage.com



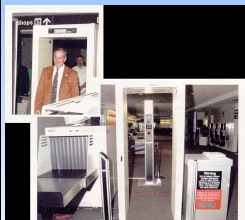
www.visionics.com

© Jain, 2004

Applications

- Video Surveillance (On-line or off-line)

Face Scan at Airports



The St. Petersburg-Clearwater Airport installed facial recognition systems at two security checkpoints in January. The first full-screen system forces travelers that stop pictures of passengers as they pass through inspection centers. The passengers' faces instantly are compared for a database of images of wanted criminals. Sheriff's Office (see above left) was one of the first people to pass through the new security system.

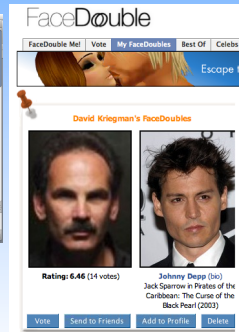
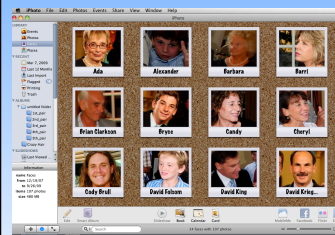


www.facesnap.de

© Jain, 2004

Applications

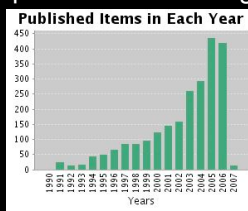
- Tracking people in games (Kinnect)



© Jain, 2004

Face Recognition Home Page

<http://www.face-rec.org/>



Graph shows how many items on face recognition were published between 1991 and 2006.

CSE190, Winter 2011

Why is Face Recognition Hard?

Many faces of Madonna



UCSD Jacobs Computer Science and Engineering

Who are these people?



[Sinha and Poggio 1996]

Why is Face Recognition Hard?



Face Recognition Difficulties

- Identify similar faces (**inter-class similarity**)
- Accommodate **intra-class variability** due to:
 - head pose
 - illumination conditions
 - expressions
 - facial hair
 - aging effects
 - glasses
 - partial occlusion
 - camera variability
 - Artistic renderings

© Jain, 2004

Inter-class Similarity

- Different persons may have very similar **appearance**



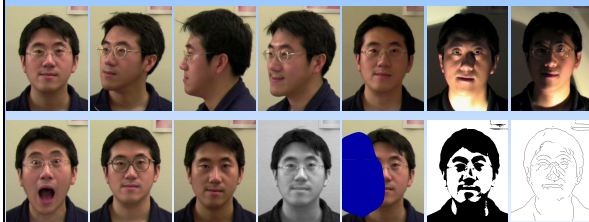
Twins

Father and son

© Jain, 2004

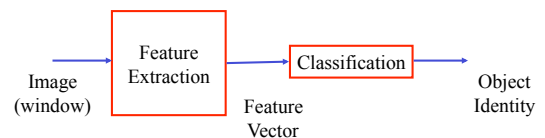
Intra-class Variability

- Faces with intra-subject variations in pose, illumination, expression, accessories, color, occlusions, and brightness



© Jain, 2004

Sketch of a Pattern Recognition Architecture



CS252A, Winter 2011

Computer Vision I

Example: Face Detection

- Scan window over image.
- Classify window as either:
 - Face
 - Non-face



CS252A, Winter 2011

Computer Vision I

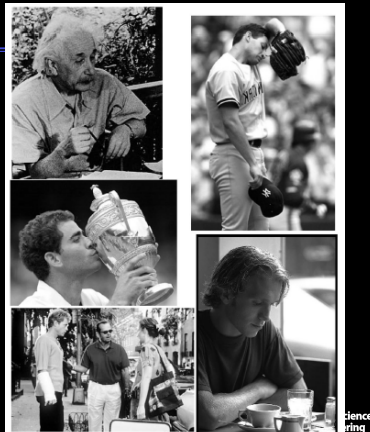
Detection Test Sets

Data Set	Location	Description
MIT Test Set [154]	http://www.cs.cmu.edu/~har	Two sets of high and low resolution gray scale images with multiple faces in complex background.
CMU Test Set [128]	http://www.cs.cmu.edu/~har	130 gray scale images with a total of 507 frontal faces.
CMU Profile Face Test Set [141]	ftp://eyes.ius.cs.cmu.edu/aar20/ ftp://testing_face_images.tar.gz	208 gray scale images with faces in profile views.
Kodak Data Set [94]	Eastman Kodak Corporation	Faces of multiple size, pose and under varying illumination in color images. Designed for face detection and recognition.

UCSD Jacobs Computer Science and Engineering

Profile views

Schneiderman's Test set



Face Detection: Experimental Results

Test sets: two CMU benchmark data sets
 Test set 1: 125 images with 483 faces
 Test set 2: 20 images with 136 faces

Method	Test Set 1		Test Set 2	
	Detection Rate	False Detections	Detection Rate	False Detections
Distribution based [154]	N/A	N/A	81.9%	13
Neural network [128]	92.5%	862	90.3%	42
Naive Bayes classifier [140]	93.0%	88	91.2%	12
Kullback relative information [24]	98.0%	12758	N/A	N/A
Support vector machine [107]	N/A	N/A	74.2%	20
Mixture of factor analyzers [175]	92.3%	82	89.4%	3
Fisher linear discriminant [175]	93.6%	74	91.5%	1
SNoW with primitive features [176]	94.2%	84	93.6%	3
SNoW with multi-scale features [176]	94.8%	78	94.1%	3
Inductive learning [38]	90%	N/A	N/A	N/A

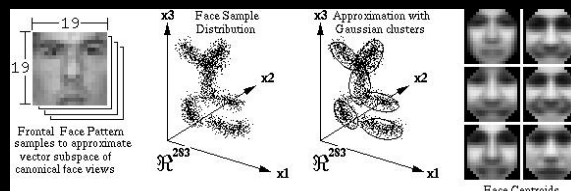
[See also work by Viola & Jones, Rehg, Schneiderman]

UCSD Jacobs Computer Science and Engineering

Generative model

- Learn about faces from examples: $P(\text{Window} | \text{Face})$
- Challenge: High dimensional data, multi-modal
- Detection: Scan window over image, and classify window if local maximum of $P(\text{Window} | \text{Face})$ and $P(\text{Window} | \text{Face}) > \text{Threshold}$

E.g., mixture of factor analyzers [Yang, Kriegman, Ahuja, 2001]

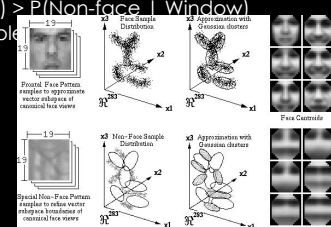


[Sung and Poggio 98]

UCSD Jacobs Computer Science and Engineering

Discriminative Model

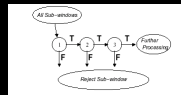
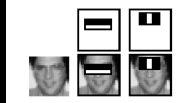
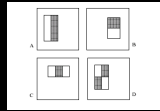
- Learn face and nonface models from examples $P(| \text{Window} | \text{Face})$ and $P(| \text{Window} | \text{Non-face})$
- Cluster samples of each class to create subclasses, and project the examples to a lower dimensional space based on multi-discriminant analysis.
- Detect faces in lower-dimensional space when $P(\text{Face} | \text{Window}) > P(\text{Non-face} | \text{Window})$
- Add non-face example using bootstrapping



[Sung and Poggio 98]

"State of the Art Method:" Viola Jones

- [Viola and Jones CVPR 01]:
 - A face is modeled as a set of Harr-like features
 - A fast way to compute simple rectangle features
 - Use Adaboost to focus on a small set of features
 - Cascade of simple classifiers
- Error rate comparable to the best
- Fast: 15 Fps, 700Mhz Pentium, half resolution video images



UCSD Computer Science
Jacobs and Engineering