Pattern Recognition

Dr. Terence Sim

Example: face detection in cameras

Example: optical character recognition

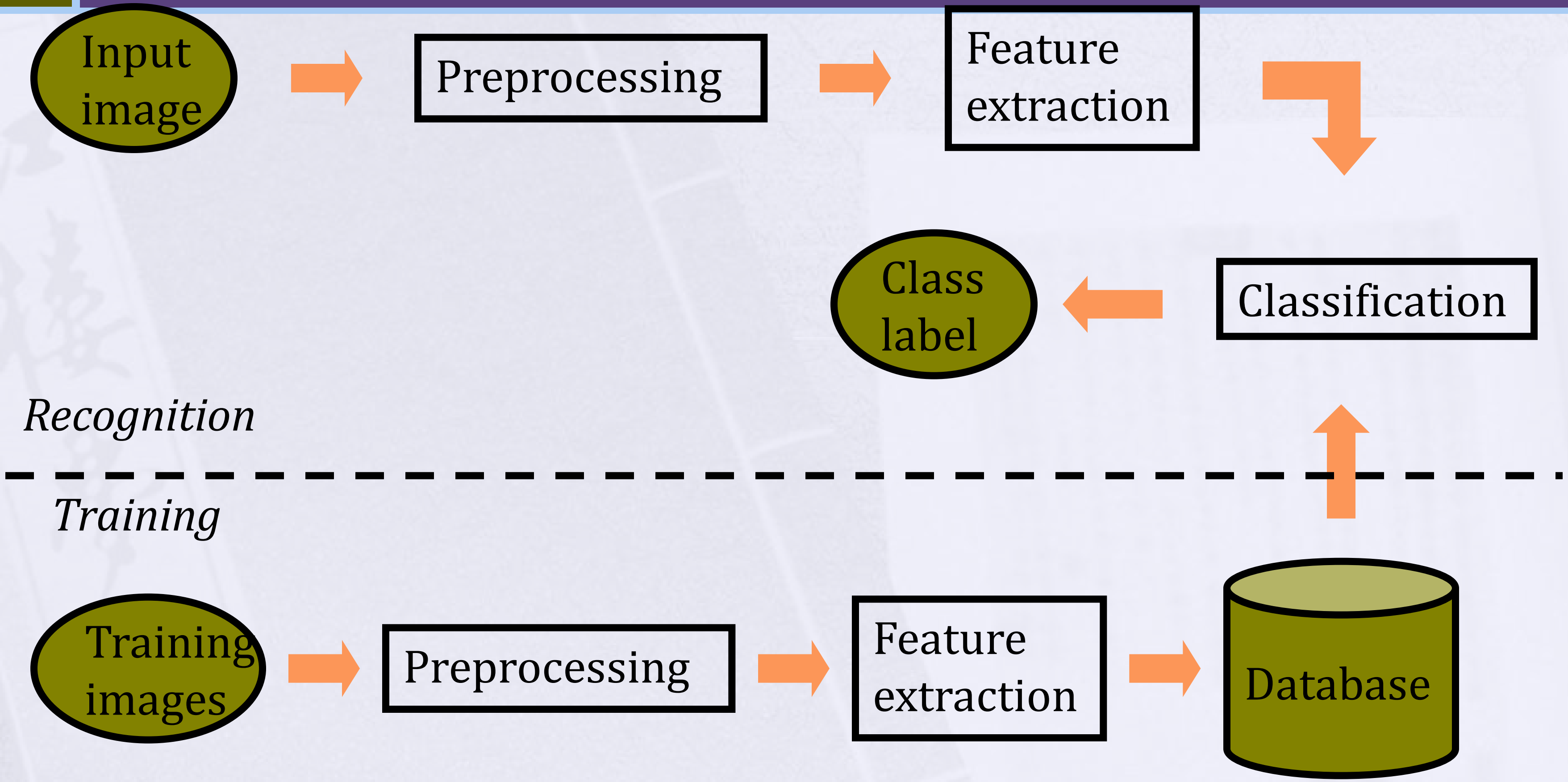
Basic Ideas: Definition

- Let $S = \{\omega_1, \omega_2 \ldots \omega_C\}$ be the set of pre-defined C classes
  - e.g. {face, non-face}, {a,b,c,d ...}

- Let $\boldsymbol{x}$ be the feature vector in $\boldsymbol{R^n}$

- Classifier is a function $f : \boldsymbol{R^n} \rightarrow S$
  - We say that a classifier *assigns a class label* to the feature vector (pattern)

Basic Ideas: Typical Image PR pipeline

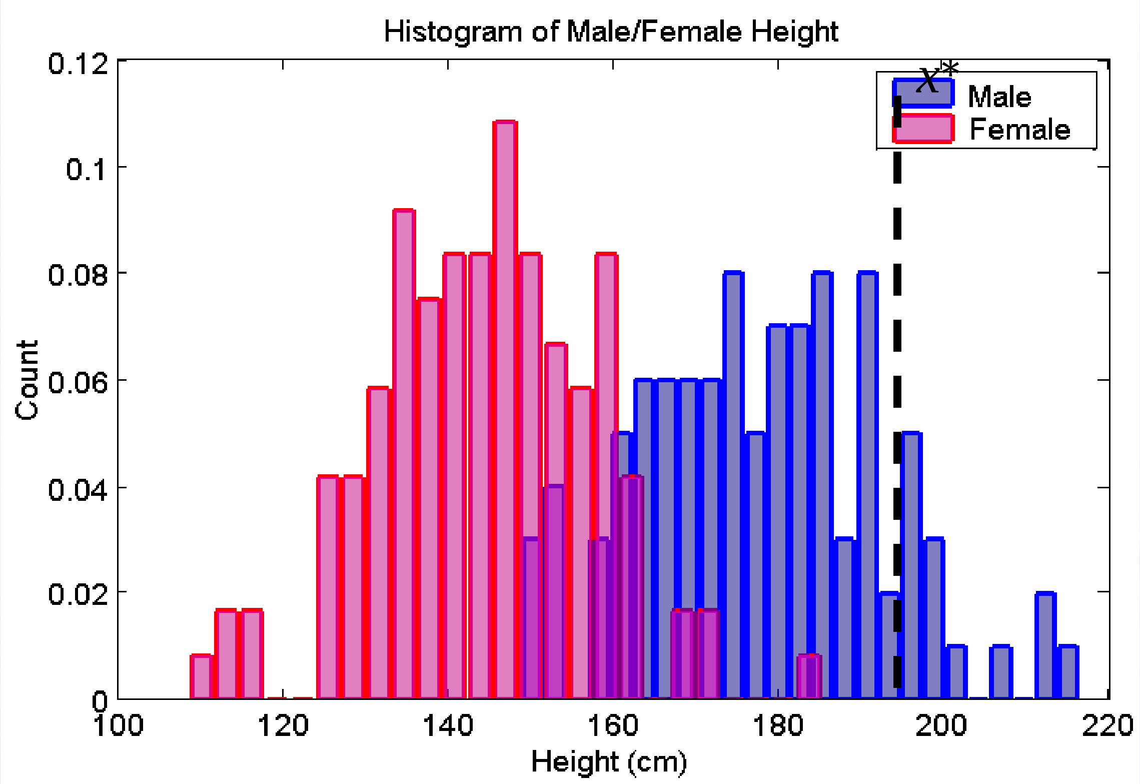


3 Important Questions

- What features are best?
  - _domain_______ knowledge
  - Ask the expert
  - Guess
  - Learn from training data
- Given features, how to design classifier?
  - What type of classifier?
  - How to find decision boundary?
- How good is the classifier?
  - How to evaluate performance?

Gender classification

- What features to use?
- Try height
  - Idea: males are generally taller than females
  - Therefore, a large value of height implies male
  - How true is this?

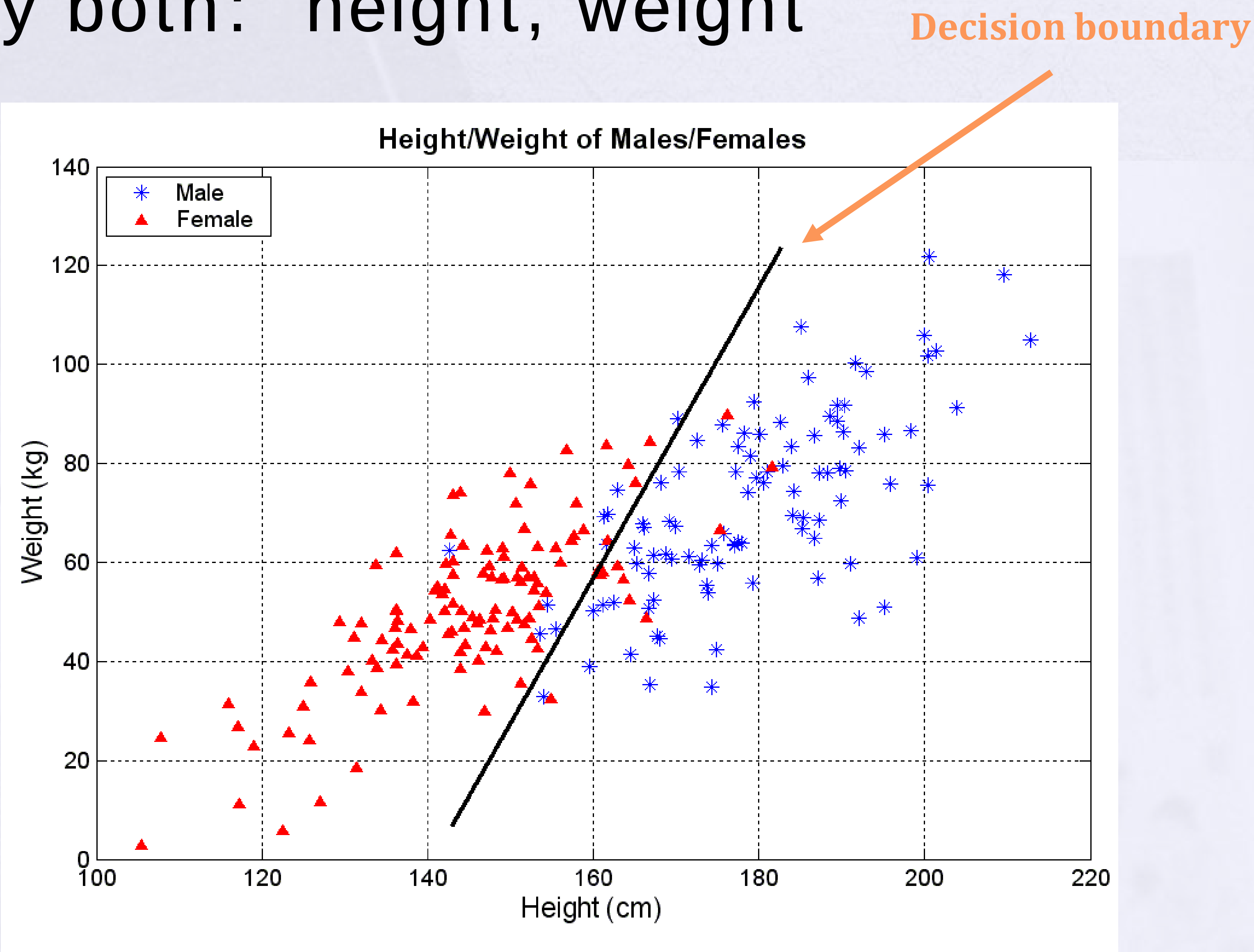


Decision boundary

- Boundary between 2 classes: x*
- Decision rule:
  - If x < x* then decide *Female*
  - Else If x > x* then decide *Male*
  - Else flip a coin

Features

- Try both: height, weight



2 features

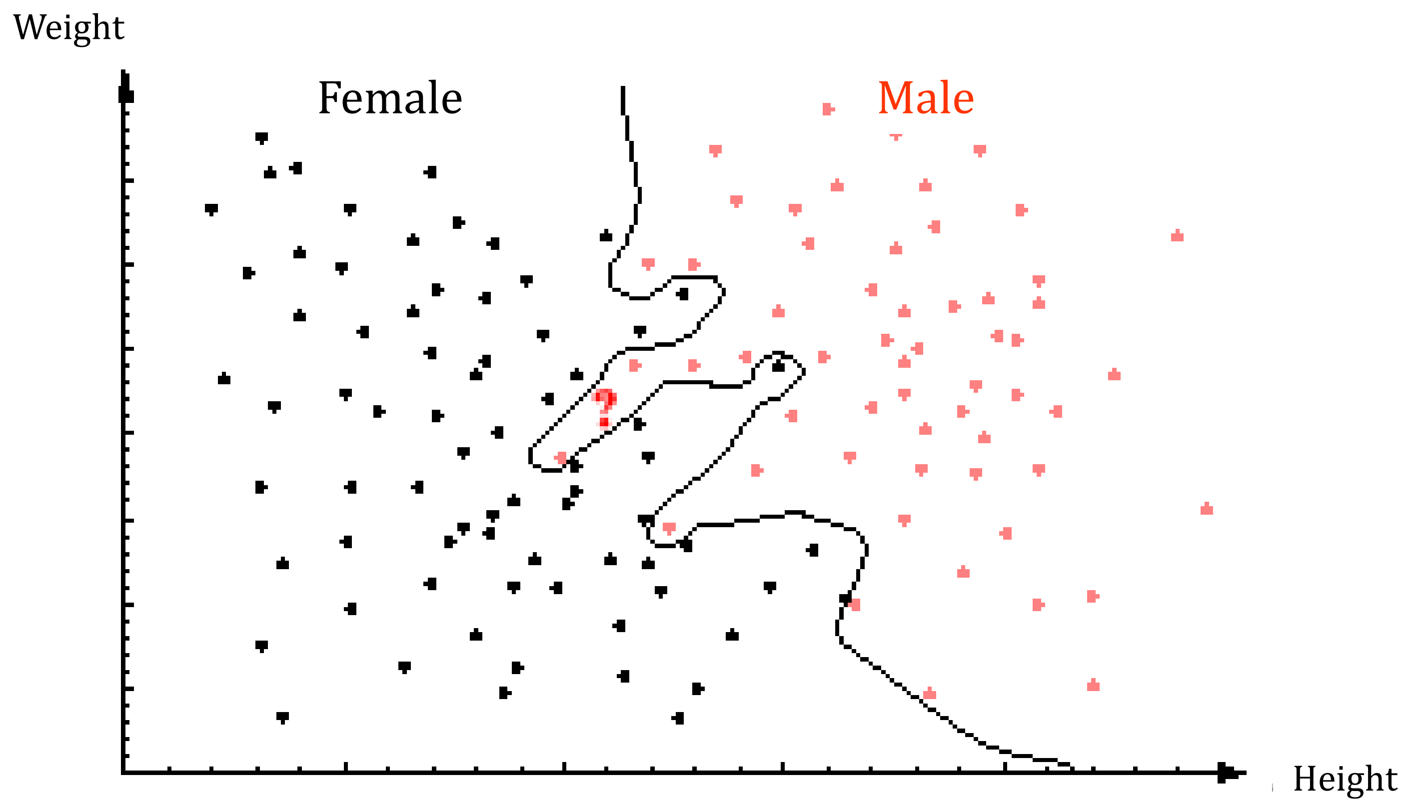
- $x = [\text{height}, \text{weight}]^T$
- Decision boundary is a line
- Decision rule:
  - If x lies above line, then decide *Male*
  - Else If x lies below line, then decide *Female*
  - Else flip a coin

- But still some errors ...

More features?

- We might add other features that are not correlated with the ones we already have.
  - A precaution should be taken not to reduce the performance by adding such “noisy features”
- Ideally, the best decision boundary should be the one which provides an optimal performance such as in the following figure:

Perfect Decision Boundary?

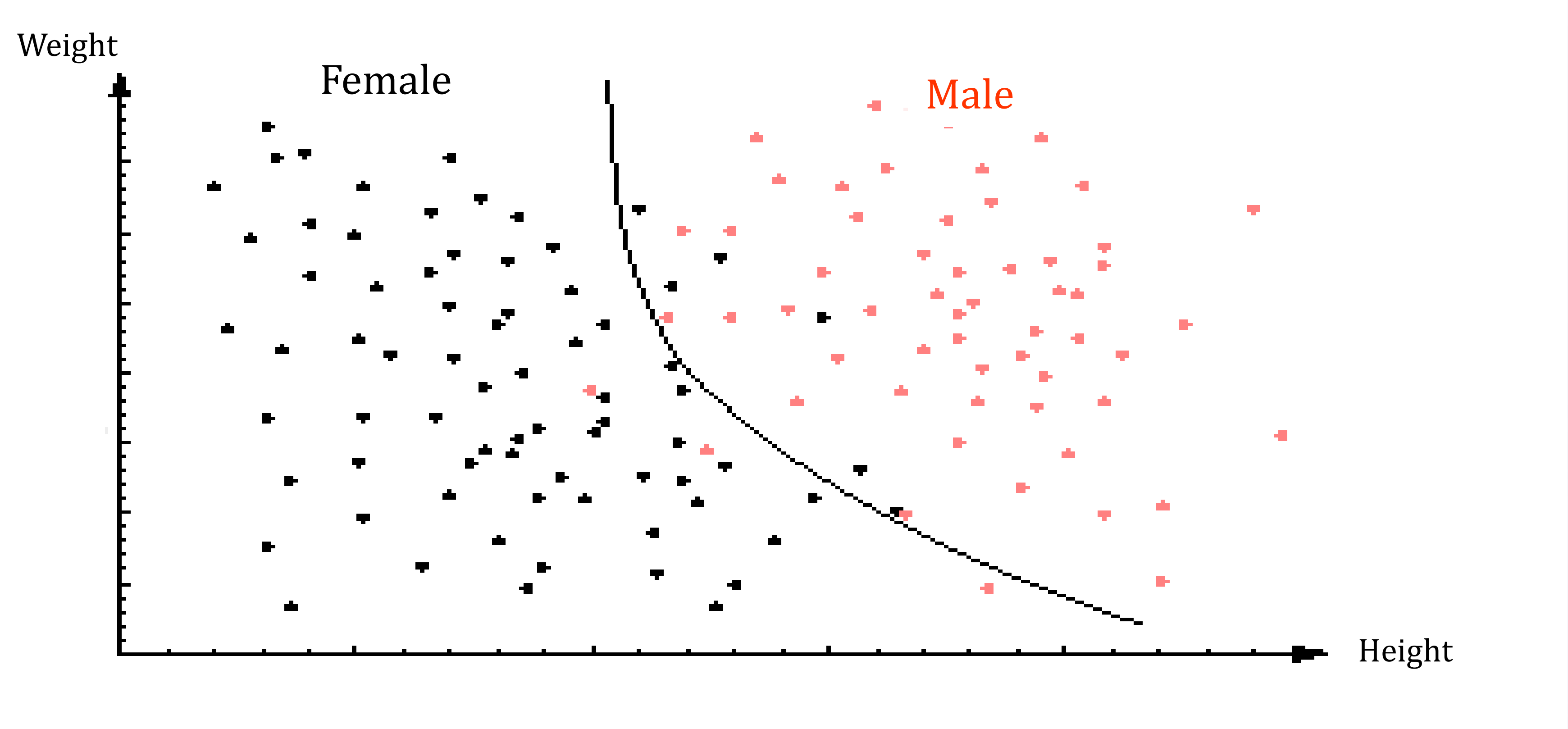


Generalization

- However, our satisfaction is premature because the central aim of designing a classifier is to correctly classify ***novel*** input

Issue of generalization!

Non-linear boundary



Choices, choices, choices

Feature

- Edges
- Color
- Shape
- Texture
- Histogram of Oriented Gradients (HOG)
- Local Binary Pattern (LBP)
- Wavelets
- FFT

Classifier

- Bayes' classifier
- Support Vector Machine (SVM)
- Artificial Neural Network (ANN)
- k-nearest neighbor (kNN)
- Decision Tree
- Adaboost
- Bayesian Network

Theoretically Optimal Classifier

BAYES' CLASSIFIER

Statistical PR

- Suppose you have no observation
  - How to classify?
  - You only know the prior probabilities, e.g. males in population = 50.85%

- Decision rule with only the prior information
  - Decide ω_1 if $P(\omega_1) > P(\omega_2)$ otherwise decide ω_2

Bayes' Classifier

- Now suppose you observed X
- How to classify?
- Bayes' classifier says:

Maximum A Posteriori

$$\omega^* = \arg\max_{\omega_j} P(\omega_j \mid x)$$

- That is, assign *x* to label ω_j such that $P(\omega_j \mid x)$ is largest among all $P(\omega_i \mid x)$

Bayes' Classifier

- Bayes' Rule: $$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

- So

$$\omega^* = \arg\max_{\omega_j} P(\omega_j | x)$$

Posterior

Likelihood

Prior

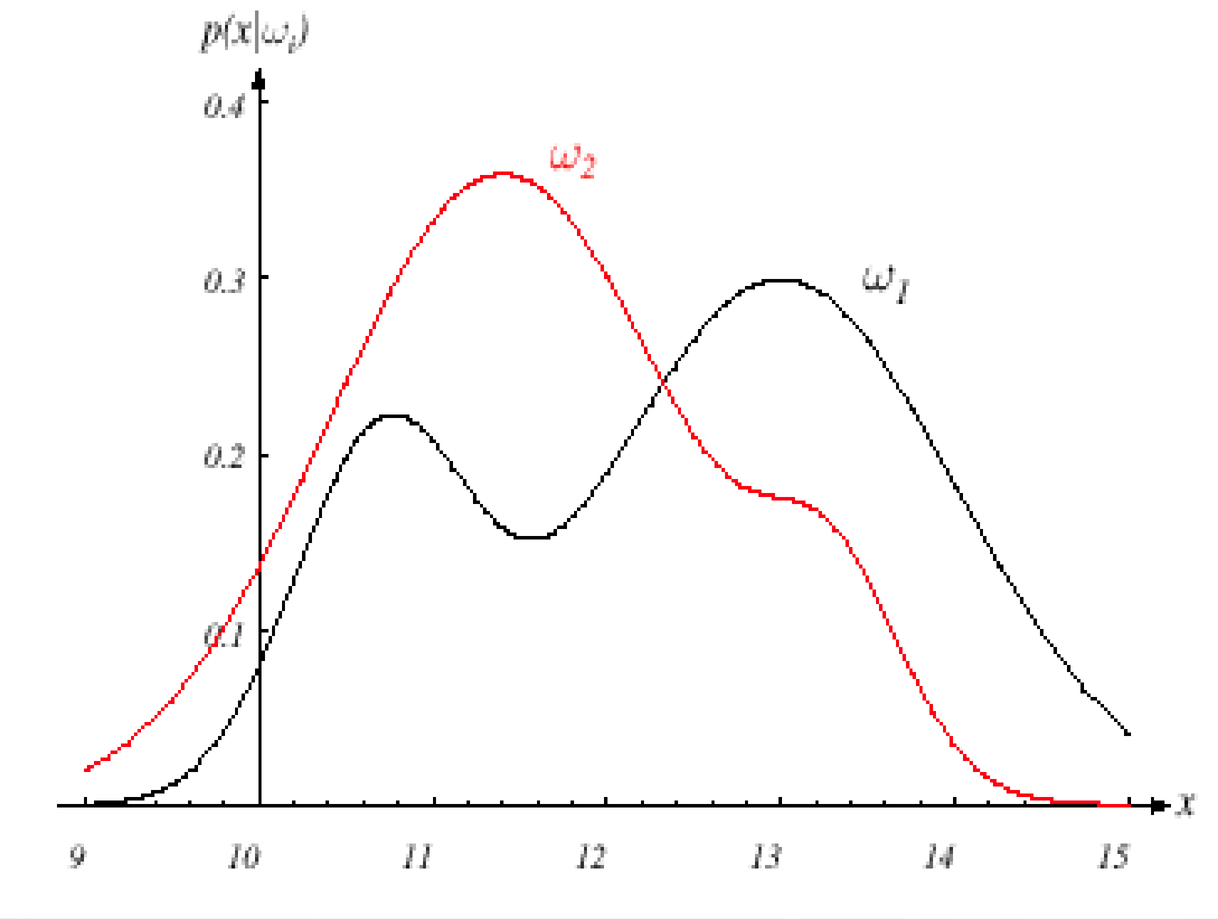
$$= \arg\max_{\omega_j} \frac{P(x|\omega_j) \bullet P(\omega_j)}{P(x)}$$

Evidence

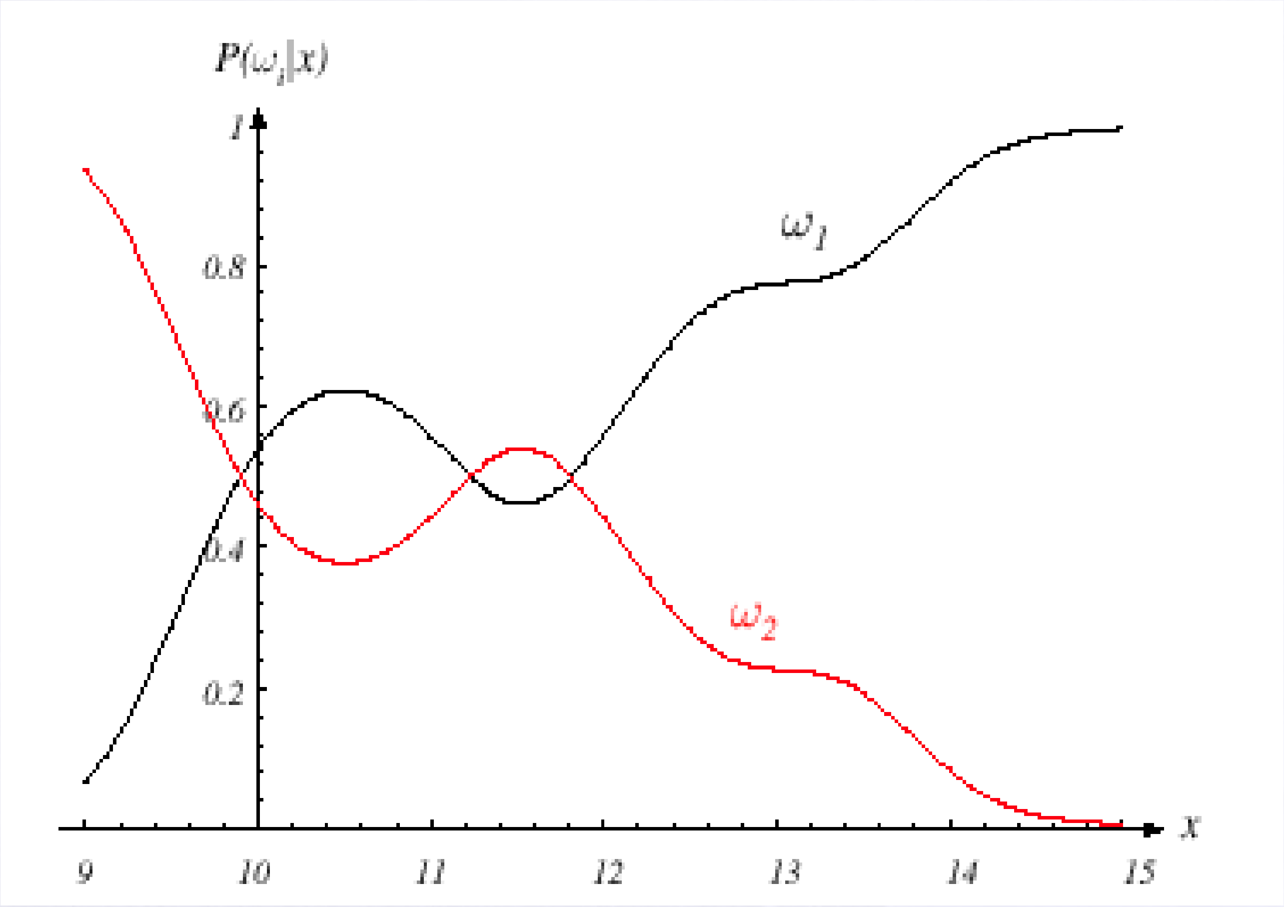
$$= \arg\max_{\omega_j} P(x|\omega_j) \bullet P(\omega_j)$$

Likelihood: learn from training data

a.k.a. class-conditional probability



Maximum A Posteriori



Special case

- Equal priors $P(\omega_1) = P(\omega_2) = \cdots = P(\omega_C) = \frac{1}{C}$

- Then $$\omega^* = \arg\max_{\omega_j} P(x \mid \omega_j) \bullet \cancel{P(\omega_j)}$$

Maximum Likelihood

Special case: only 2 classes

- Decide ω_1 *if* $P(\omega_1 \mid x) > P(\omega_2 \mid x)$; otherwise decide ω_2

Alternatively:

- Decide ω_1 *if* $g(x) > 0$ otherwise decide ω_2
- Where $g(x) = P(\omega_1 \mid x) - P(\omega_2 \mid x)$
  - *g(x) is called a* ***Discriminant Function***

Bayes' with cost

Let $\{\omega_1, \omega_2, \ldots, \omega_c\}$ be the set of C classes

Let λ_{ij} be the loss incurred for deciding ω_i when the class is ω_j

Likelihood Ratio

Then Bayes' rule that minimizes risk (expected loss) is:

$$if \ \frac{P(x \mid \omega_1)}{P(x \mid \omega_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \cdot \frac{P(\omega_2)}{P(\omega_1)}$$

Then decide ω_1

Otherwise decide ω_2

Note: right-hand side independent of input x

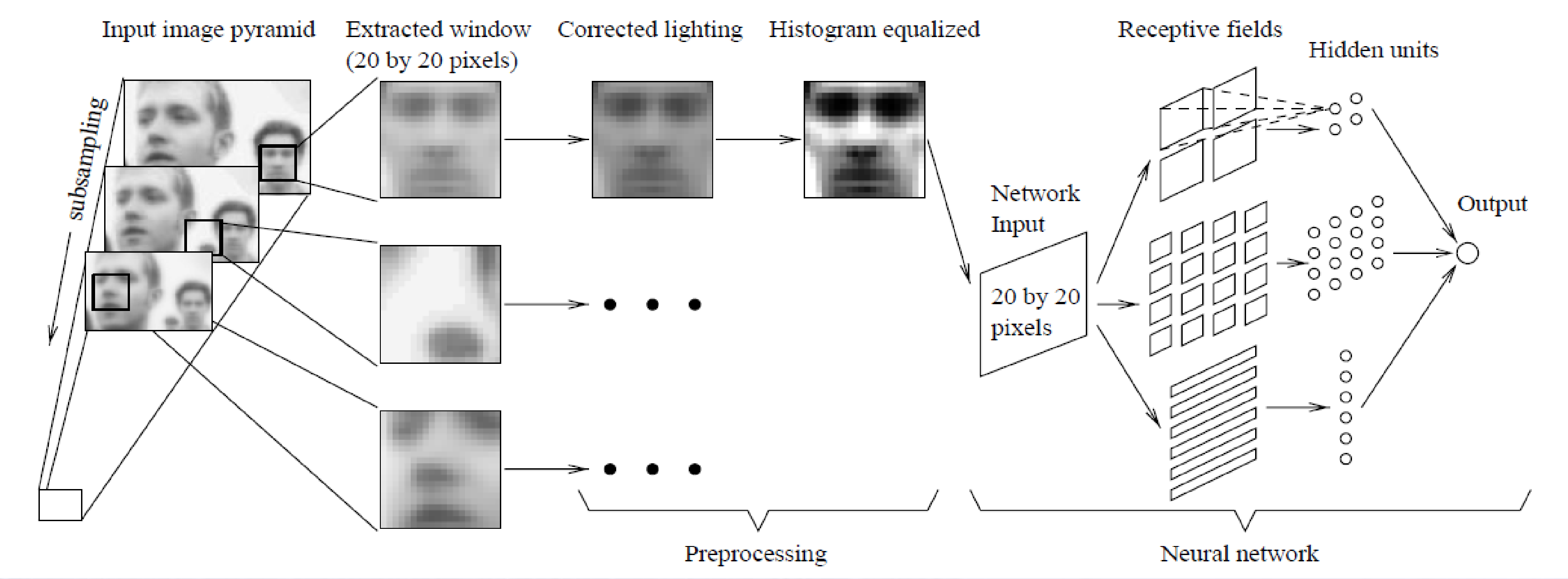
Note: if $\lambda_{21} = \lambda_{12} = 1$ and $\lambda_{11} = \lambda_{22} = 0$, then MAP!

Case Study

VIOLA-JONES FACE DETECTION

Prior face detector

- Using ANN, by Sung Kah Kay (MIT), and also by Henry Rowley (CMU)



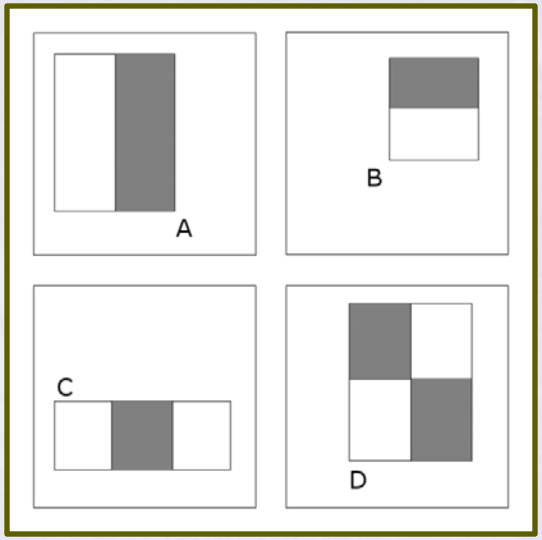
Viola Jones Technique Overview

Slides from Prof. Padhraic Smyth, UC Irvine

- Three major contributions/phases of the algorithm :
  - Feature extraction
  - Classification using boosting
  - Multi-scale detection algorithm
- Feature extraction and feature evaluation.
  - Rectangular features are used, with a new image representation their calculation is very fast.
- Classifier training and feature selection using a slight variation of a method called AdaBoost.
- A combination of simple classifiers is very effective
- Paper: Robust Real-Time Object Detection, IJCV 2001.

Features

- Four basic types.
  - They are easy to calculate.
  - The white areas are subtracted from the black ones.
  - A special representation of the sample called the **integral image** makes feature extraction faster.

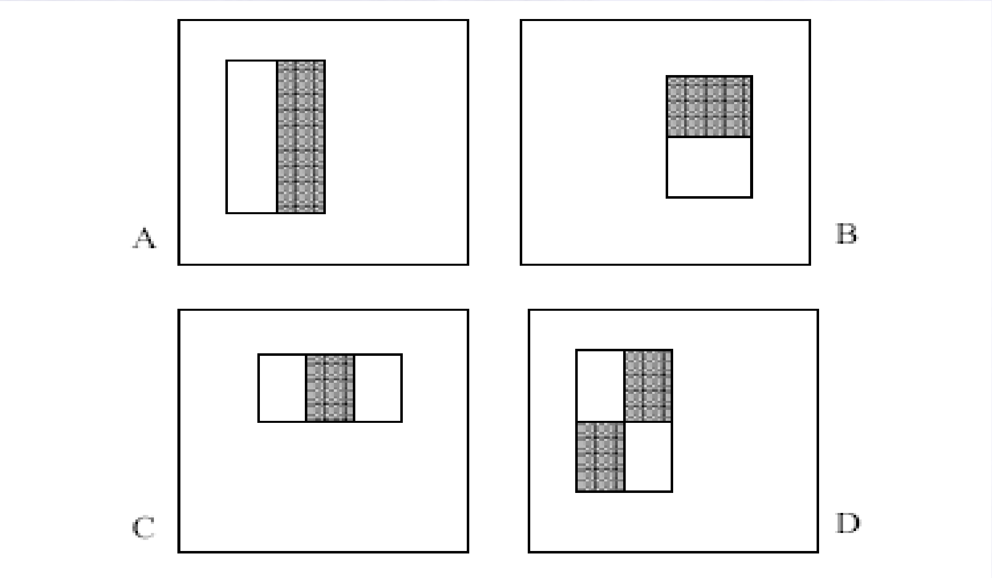


Slides from Prof. Padhraic Smyth, UC Irvine

Feature Extraction

Slides from Prof. Padhraic Smyth, UC Irvine

- Features are extracted from sub windows of a sample image.
  - The base size for a sub window is 24 by 24 pixels.
  - Each of the four feature types are scaled and shifted across all possible combinations
    - In a 24 pixel by 24 pixel sub window there are ~160,000 possible features to be calculated.



Boosting with Single Feature Perceptrons

- Viola-Jones version of Boosting:
  - "simple" (weak) classifier = single-feature perceptron
    - see last slide
  - With K features (e.g., K = 160,000) we have 160,000 different single-feature perceptrons
  - At each stage of boosting
    - given reweighted data from previous stage
    - Train all K (160,000) single-feature perceptrons
    - Select the single best classifier at this stage
    - Combine it with the other previously selected classifiers
    - Reweight the data
    - Learn all K classifiers again, select the best, combine, reweight
    - Repeat until you have T classifiers selected
  - Hugely computationally intensive
    - Learning K perceptrons T times
    - E.g., K = 160,000 and T = 1000

Slides from Prof. Padhraic Smyth, UC Irvine

How is classifier combining done?

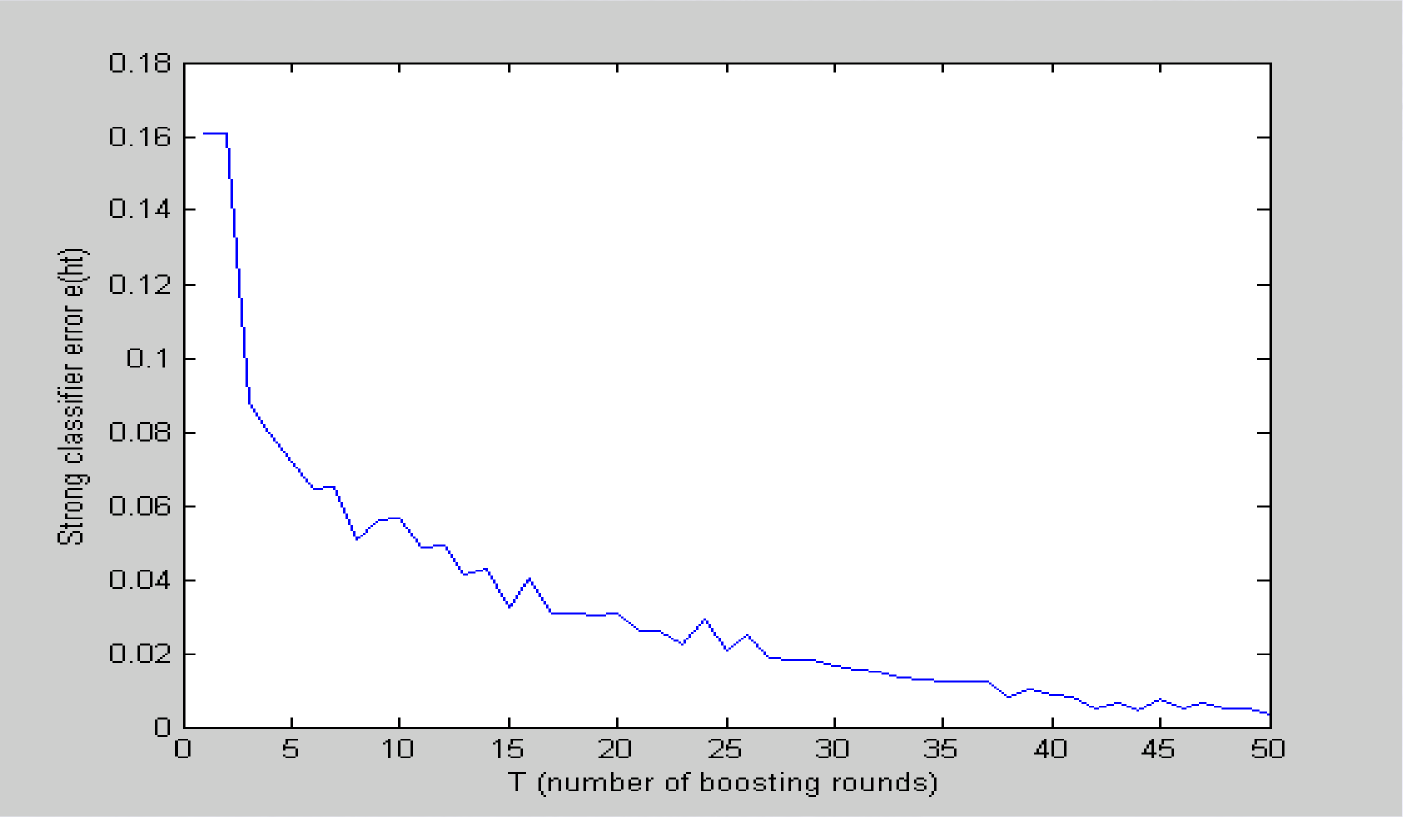
Slides from Prof. Padhraic Smyth, UC Irvine

- At each stage we select the best classifier on the current iteration and combine it with the set of classifiers learned so far

- How are the classifiers combined?
  - Take the weight*feature for each classifier, sum these up, and compare to a threshold (very simple)
  - Boosting algorithm automatically provides the appropriate weight for each classifier and the threshold
  - This version of boosting is known as the AdaBoost algorithm
  - Some nice mathematical theory shows that it is in fact a very powerful machine learning technique

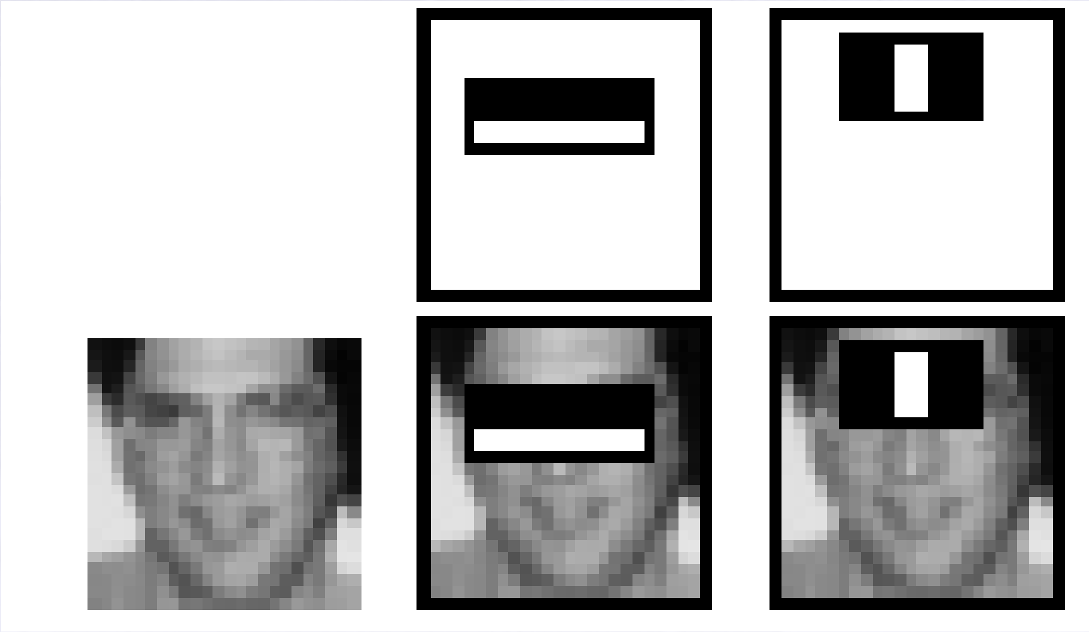
Reduction in Error as Boosting adds Classifiers

Slides from Prof. Padhraic Smyth, UC Irvine



Useful Features Learned by Boosting

Slides from Prof. Padhraic Smyth, UC Irvine



Detection in Real Images

Slides from Prof. Padhraic Smyth, UC Irvine

- Basic classifier operates on 24 x 24 subwindows
- Scaling:
  - Scale the detector (rather than the images)
  - Features can easily be evaluated at any scale
  - Scale by factors of 1.25
- Location:
  - Move detector around the image (e.g., 1 pixel increments)
- Final Detections
  - A real face may result in multiple nearby detections
  - Postprocess detected subwindows to combine overlapping detections into a single detection

Training

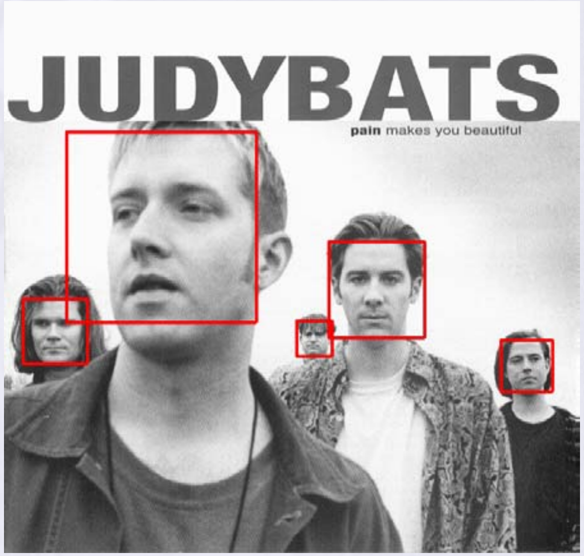
Slides from Prof. Padhraic Smyth, UC Irvine

- In paper, 24x24 images of faces and non faces (positive and negative examples).

Sample results using the Viola-Jones Detector

Slides from Prof. Padhraic Smyth, UC Irvine

- Notice detection at multiple scales



Slides from Prof. Padhraic Smyth, UC Irvine

More Detection Examples

Practical implementation

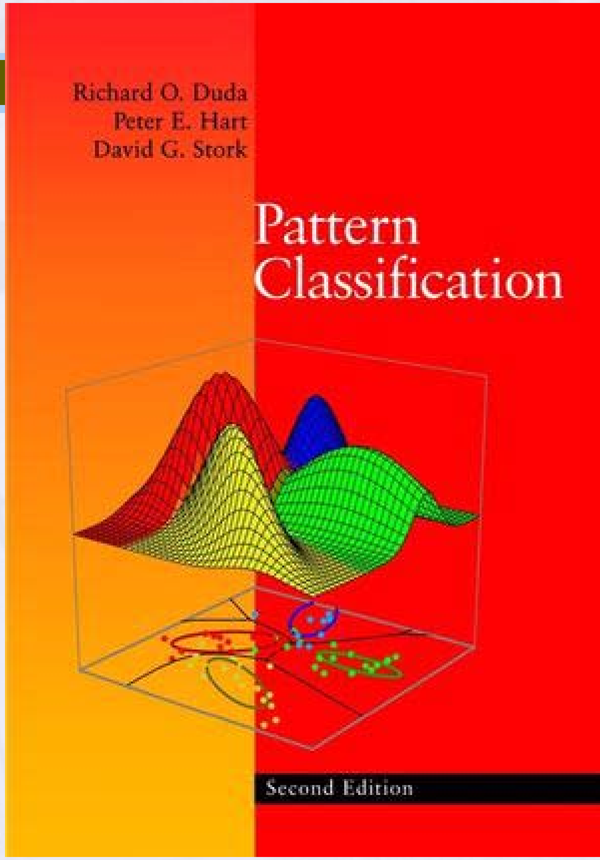
Slides from Prof. Padhraic Smyth, UC Irvine

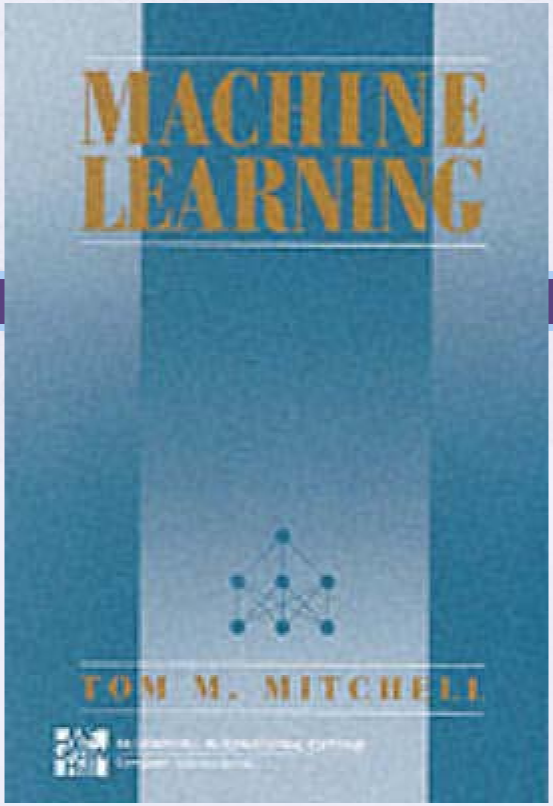
- Details discussed in Viola-Jones paper
- Training time = weeks (with 5k faces and 9.5k non-faces)
- Final detector has 38 layers in the cascade, 6060 features
- 700 Mhz processor:
  - Can process a 384 x 288 image in 0.067 seconds (in 2003 when paper was written)

Summary

- Pattern Recognition or Classification means assigning class label to input pattern.
- Choosing features is an art!
- Given the right features, many classifiers work equally well.
  - Some classifiers require long learning time
- Evaluating a classifier on a test set is an important part of determining its performance.

Books





- Machine Learning, Tom Mitchell, McGraw Hill, 1997
- http://www.cs.cmu.edu/~tom/mlbook.html

- Pattern Classification, 2nd Ed., R. Duda, P. Hart, D. Stork, 2000
- http://rii.ricoh.com/~stork/DHS.html