Data Mining
Classification: Bayesian Decision Theory

Lecture Notes for Chapter 2
R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. New York: Wiley, 2001.

Lecture Notes for Chapter 4
Introduction to Data Mining
by
Tan, Steinbach, Kumar

Machine Perception

- Build a machine that can recognize patterns:
  - Speech recognition
  - Fingerprint identification
  - OCR (Optical Character Recognition)
  - DNA sequence identification

Pattern Classification (Definition)

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

An Example

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

Species

Sea bass

Salmon

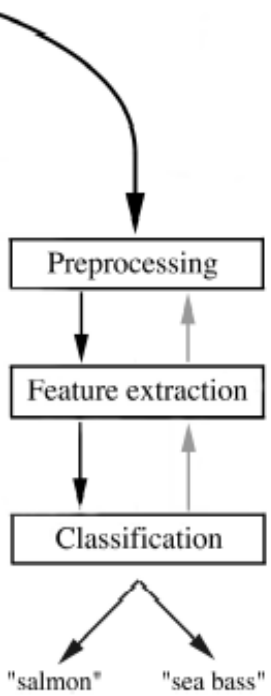
Problem Analysis

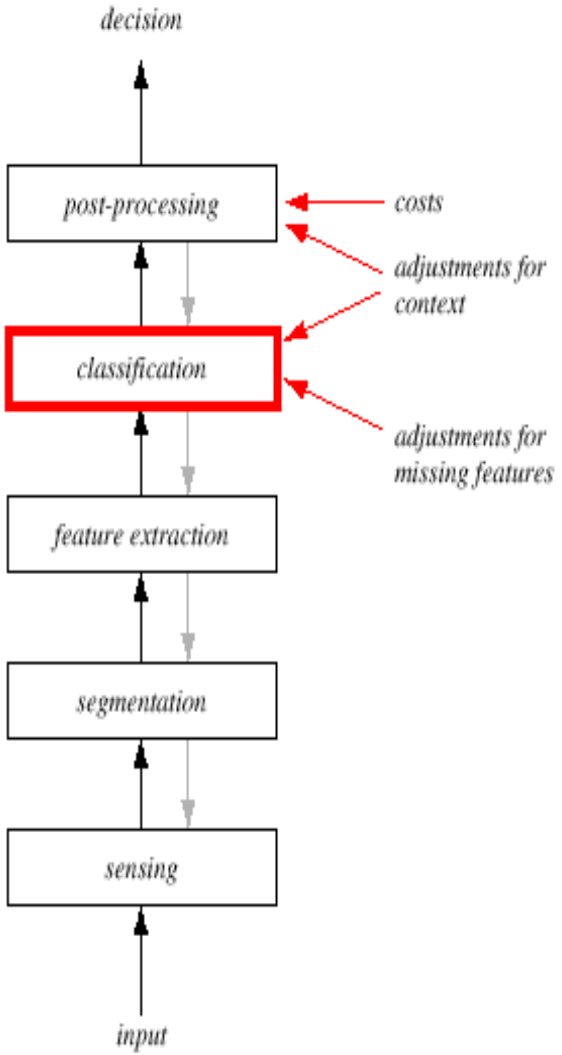
- Set up a camera and take some sample images to extract features
  - Length
  - Lightness
  - Width
  - Number and shape of fins
  - Position of the mouth, etc…
- This is the set of all suggested features to explore for use in our classifier!

Pattern Classification Systems

- Sensing
  - Use of a transducer (camera or microphone)
  - PR system depends of the bandwidth, the resolution sensitivity distortion of the transducer
- Segmentation and grouping
  - Patterns should be well separated and should not overlap
- Feature extraction
  - Discriminative features
  - Invariant features with respect to translation, rotation and scale.
- Classification
  - Use a feature vector provided by a feature extractor to assign the object to a category
- Post Processing
  - Exploit context input dependent information other than from the target pattern itself to improve performance

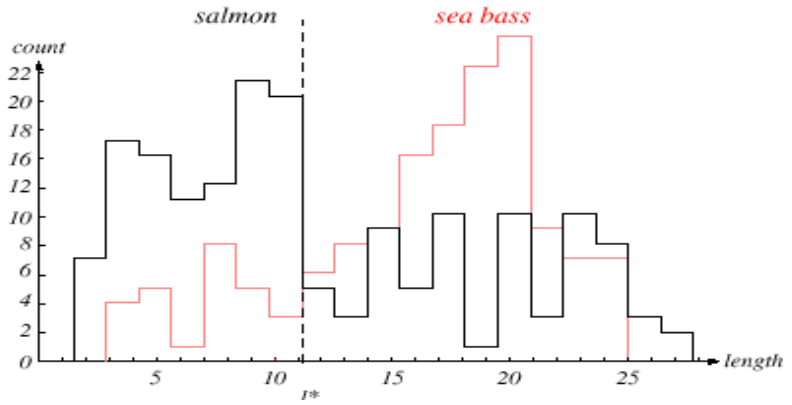
Pattern Classification Systems



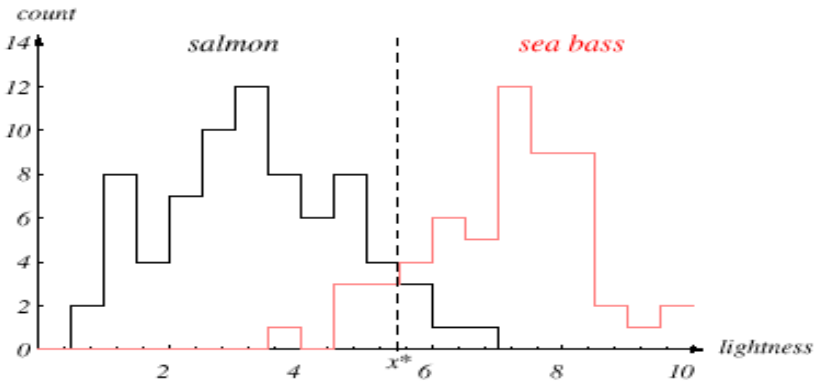


Classification

- Select the length of the fish as a possible feature for discrimination



The length is a poor feature alone! Select the lightness as a possible feature.

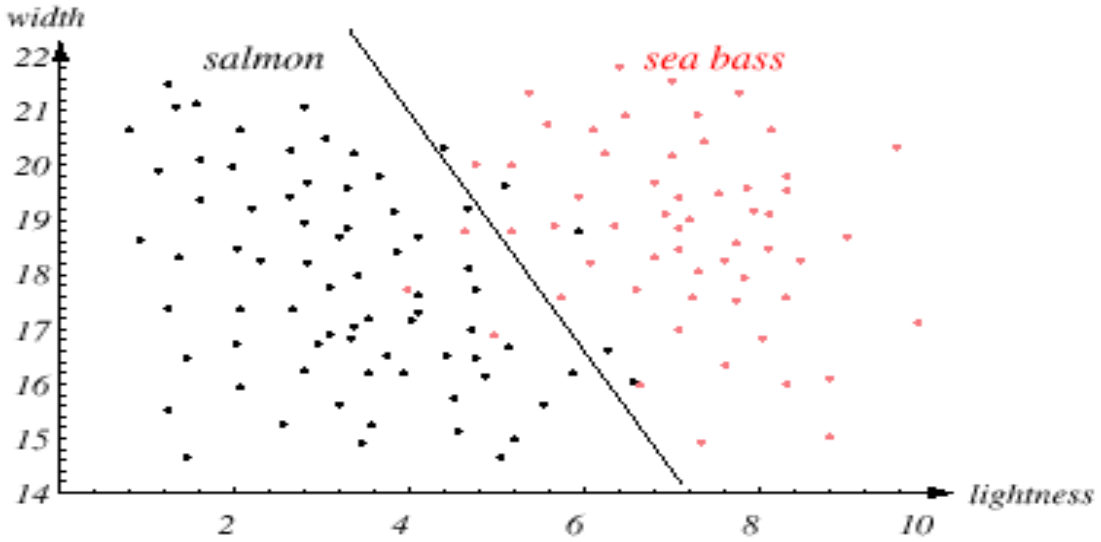


Two Features

- Adopt the lightness and add the width of the fish

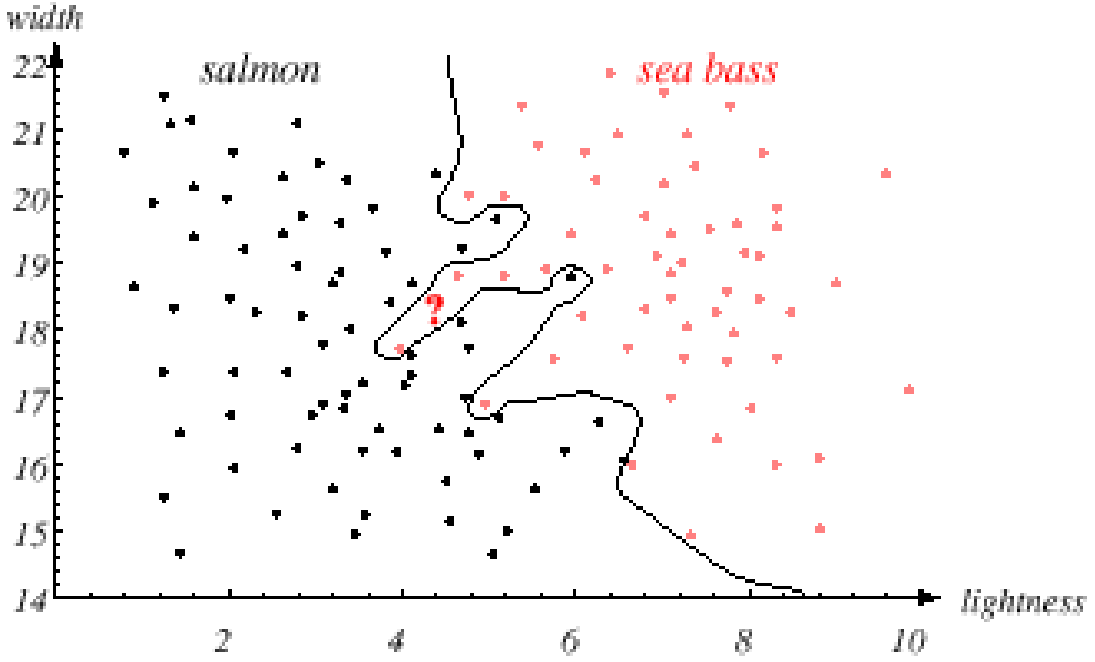
Fish $x^T = [x_1, x_2]$

Lightness **Width**



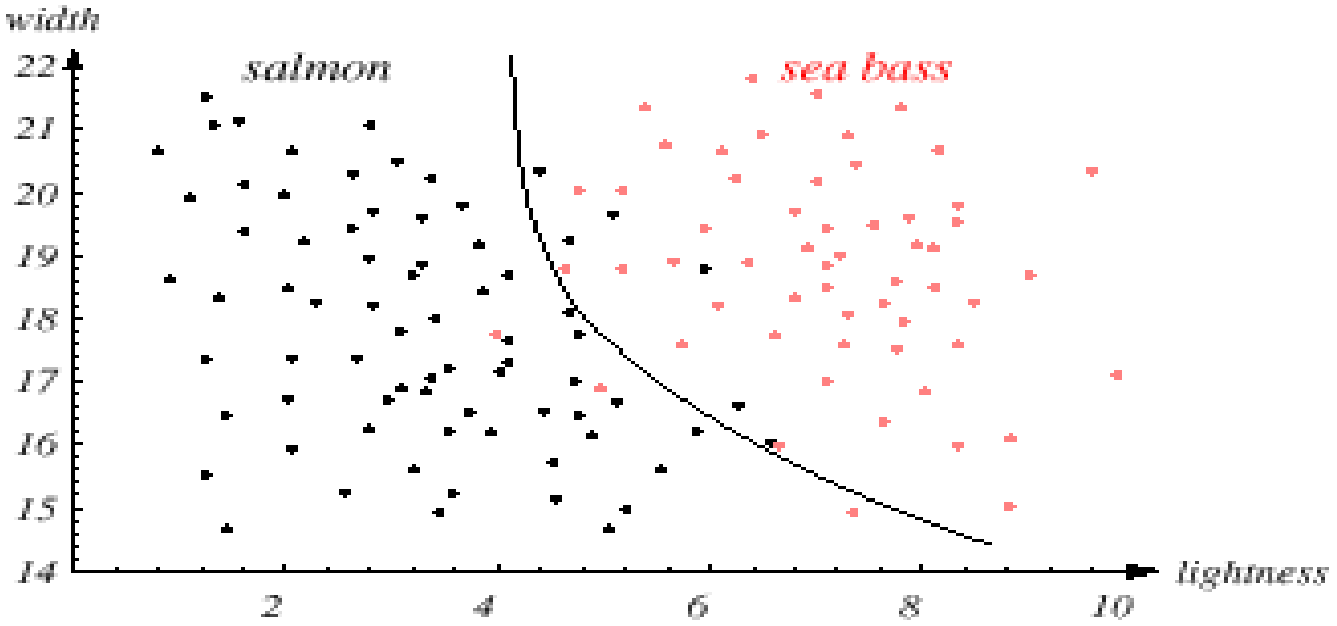
Overfitting Problem

- 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:



Generalization Ability

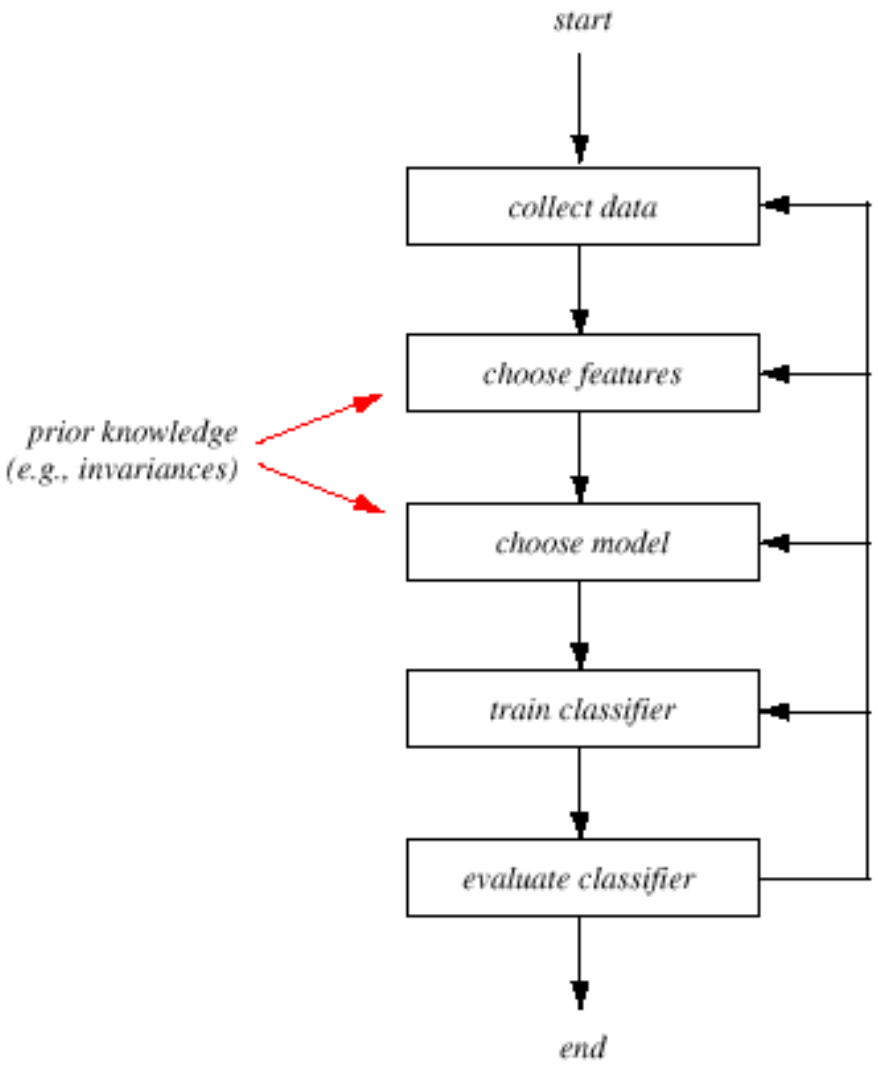
- However, our satisfaction is premature because the central aim of designing a classifier is to correctly classify novel (unseen or unknown test data) input



Overfitting avoidance (Occam's razor)

- Occam's razor: one should not use classifiers that are more complicated than are necessary, where "necessary" is determined by the quality of fit to the training data.
- One should prefer the simpler model over the more complex model.
- One should include model complexity when evaluating a model.
- Avoid overfitting by means of regularization (inclusion of penalty terms), pruning (parameters and structures of classifiers), minimizing a description length (minimize the sum of the model's algorithmic complexity and the description of the training data).

The Design Cycle



Bayesian Decision Theory

- Thomas Bayes was born in London. In 1719 he enrolled at the University of Edinburgh to study logic and theology.
  - http://en.wikipedia.org/wiki/Thomas_Bayes
- The Law of Total Probability and Bayes' rule

$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)},$$

$$p(x) = \sum_{j=1}^{2} p(x|\omega_j)P(\omega_j).$$

$$posterior = \frac{likelihood \times prior}{evidence}.$$

- Bayesian: http://www.authorstream.com/presentation/CoolDude26-13453-nips06-tutorial-Entertainment-ppt-powerpoint/

Example

- The sea bass/salmon example
  - State of nature, prior
    - State of nature is a random variable
    - The catch of salmon and sea bass is equiprobable
      - $P(\omega_1) = P(\omega_2)$ (uniform priors)
      - $P(\omega_1) + P(\omega_2) = 1$ (exclusivity and exhaustivity)
- Decision rule with only the prior information
  - Decide ω_1 if $P(\omega_1) > P(\omega_2)$ otherwise decide ω_2
- Use of the class –conditional information
- $P(x|\omega_1)$ and $P(x|\omega_2)$ describe the difference in lightness between populations of sea and salmon

Likelihood and Posterior

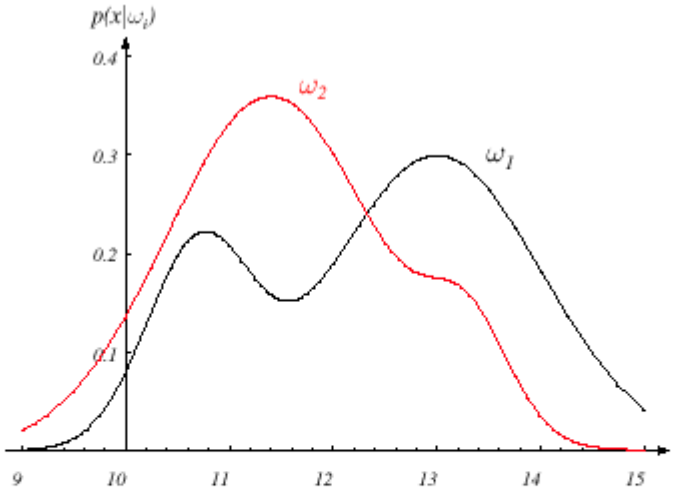


FIGURE 2.1. Hypothetical class-conditional probability density func probability density of measuring a particular feature value x given th category ω_i. If x represents the lightness of a fish, the two curves mig difference in lightness of populations of two types of fish. Density functi ized, and thus the area under each curve is 1.0. From: Richard O. Dud and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Inc.

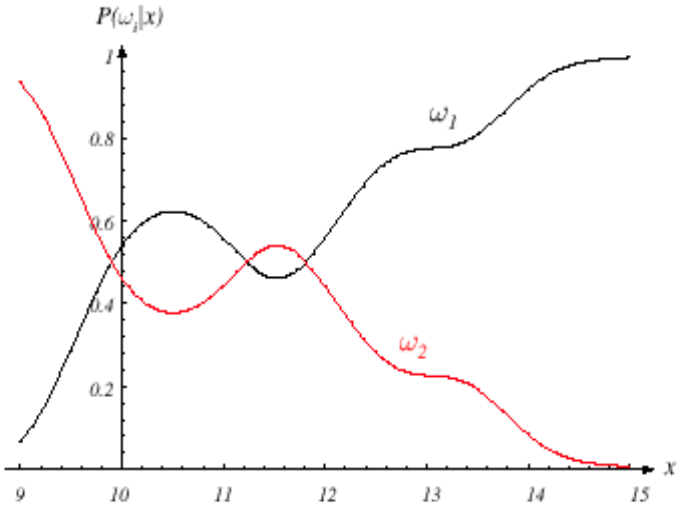


FIGURE 2.2. Posterior probabilities for the particular priors $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$ for the class-conditional probability densities shown in Fig. 2.1. Thus in this case, given that a pattern is measured to have feature value $x = 14$, the probability it is in category ω_2 is roughly 0.08, and that it is in ω_1 is 0.92. At every x, the posteriors sum to 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Bayes Decision

- Minimizing the probability of error

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

  Therefore:

$$P(error \mid x) = min\,[P(\omega_1 \mid x), P(\omega_2 \mid x)]$$

(Bayes decision)

The Gaussian Distribution

- Univariate density
  - Density which is analytically tractable
  - Continuous density
  - A lot of processes are asymptotically Gaussian
  - Handwritten characters, speech sounds are ideal or prototype corrupted by random process (central limit theorem)

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right],$$

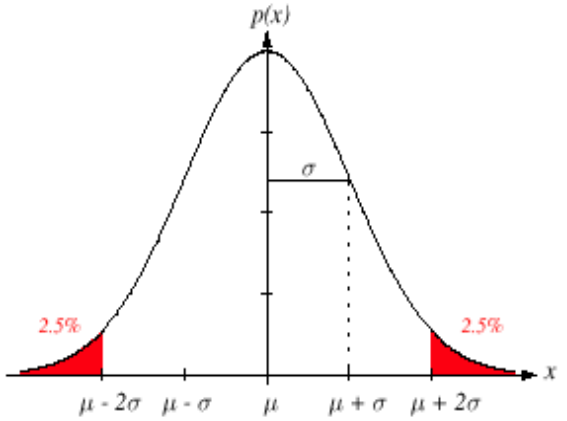


FIGURE 2.7. A univariate normal distribution has roughly 95% of its area in the range $|x-\mu| \leq 2\sigma$, as shown. The peak of the distribution has value $p(\mu) = 1/\sqrt{2\pi}\sigma$. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Multivariate Gaussian Distribution

The general multivariate normal density in d dimensions is written as

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right], \tag{37}$$

where $\mathbf{x}$ is a d-component column vector, $\boldsymbol{\mu}$ is the d-component *mean vector*, $\boldsymbol{\Sigma}$ is the d-by-d *covariance matrix*, $|\boldsymbol{\Sigma}|$ and $\boldsymbol{\Sigma}^{-1}$ are its determinant and inverse, respectively, and $(\mathbf{x}-\boldsymbol{\mu})^t$ is the transpose of $\mathbf{x}-\boldsymbol{\mu}$.* Our notation for the *inner product* is

$$\mathbf{a}^t\mathbf{b} = \sum_{i=1}^{d} a_i b_i, \tag{38}$$

and often called a *dot product*.

Discriminant Functions

- We saw that the minimum error-rate classification can be achieved by the discriminant function

$$g_i(x) = \ln P(x \mid \omega_i) + \ln P(\omega_i)$$

(Avoiding underflow !! Logarithm and sum!)

- Case of multivariate Gaussian

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln |\boldsymbol{\Sigma}_i| + \ln P(\omega_i).$$

Case $\Sigma_i = \sigma^2 I$ (I is the identity matrix)

$g_i(x) = w_i^t x + w_{i0}$ (linear discriminant function)

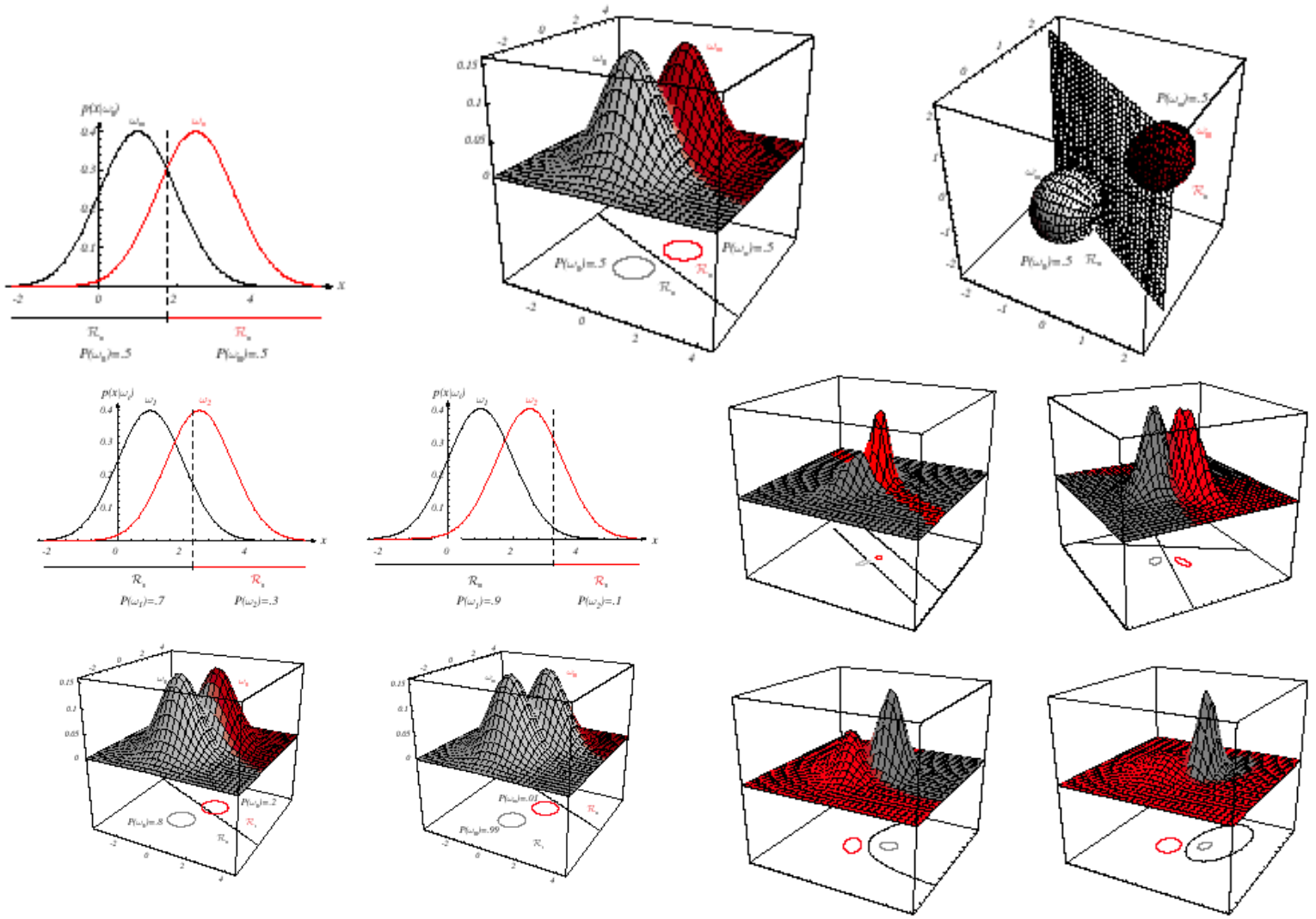
where:

$$w_i = \frac{\mu_i}{\sigma^2}; \quad w_{i0} = -\frac{1}{2\sigma^2}\mu_i^t \mu_i + \ln P(\omega_i)$$

(ω_{i0} is called the threshold for the ***i***th category!)

A classifier that uses linear discriminant functions is called “a linear machine”.

Decision Boundary



Multi-class Problem

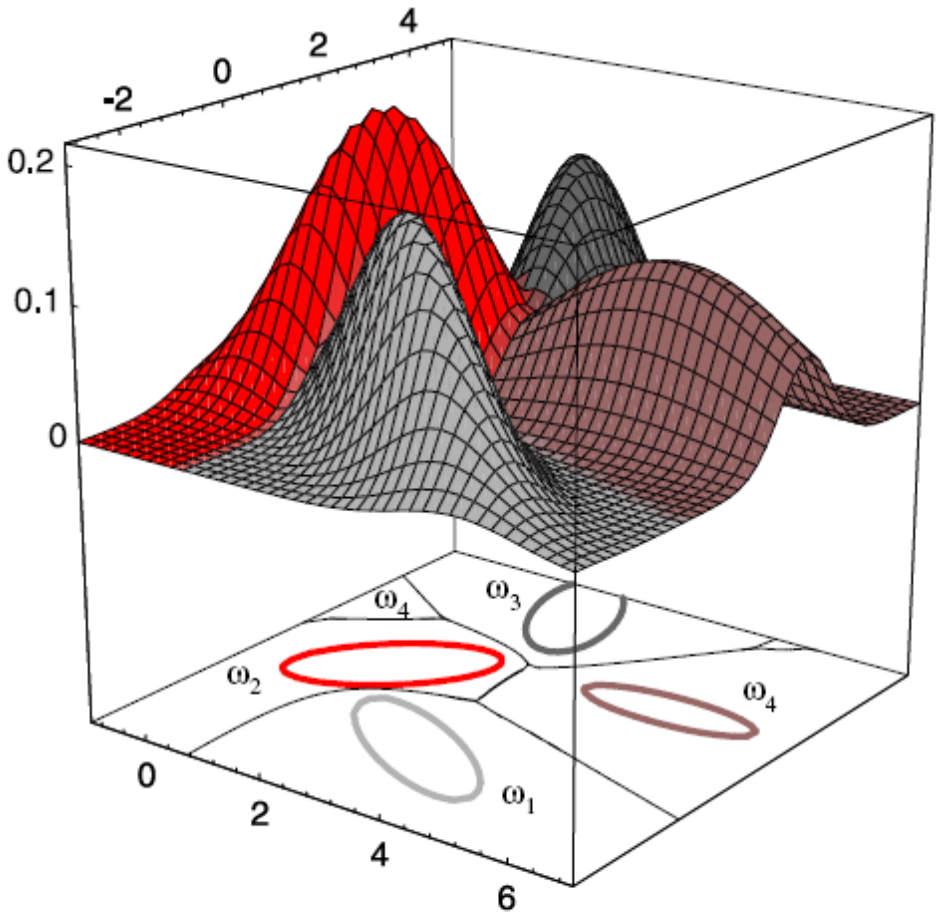


Figure 2.16: The decision regions for four normal distributions. Even with such a low number of categories, the shapes of the boundary regions can be rather complex.

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Metrics for Performance Evaluation…

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any class 1 example

Cost Matrix

	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	**Class=Yes**	**Class=No**
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

C(i|j): Cost of misclassifying class j example as class i

Computing Cost of Classification

Cost Matrix	PREDICTED CLASS		
	C(i|j)	+	-
ACTUAL CLASS	+	-1	100
	-	1	0

Model M_1	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	150	40
	-	60	250

Accuracy = 80%

Cost = 3910

Model M_2	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	250	45
	-	5	200

Accuracy = 90%

Cost = 4255

Cost vs Accuracy

Count	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

Cost	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	p	q
	Class=No	q	p

Accuracy is proportional to cost if
1. C(Yes|No)=C(No|Yes) = q
2. C(Yes|Yes)=C(No|No) = p

$N = a + b + c + d$

$\text{Accuracy} = (a + d)/N$

$$\text{Cost} = p\,(a + d) + q\,(b + c)$$
$$= p\,(a + d) + q\,(N - a - d)$$
$$= q\,N - (q - p)(a + d)$$
$$= N\,[q - (q-p) \times \text{Accuracy}]$$

Cost-Sensitive Measures

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F-measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

Model Evaluation

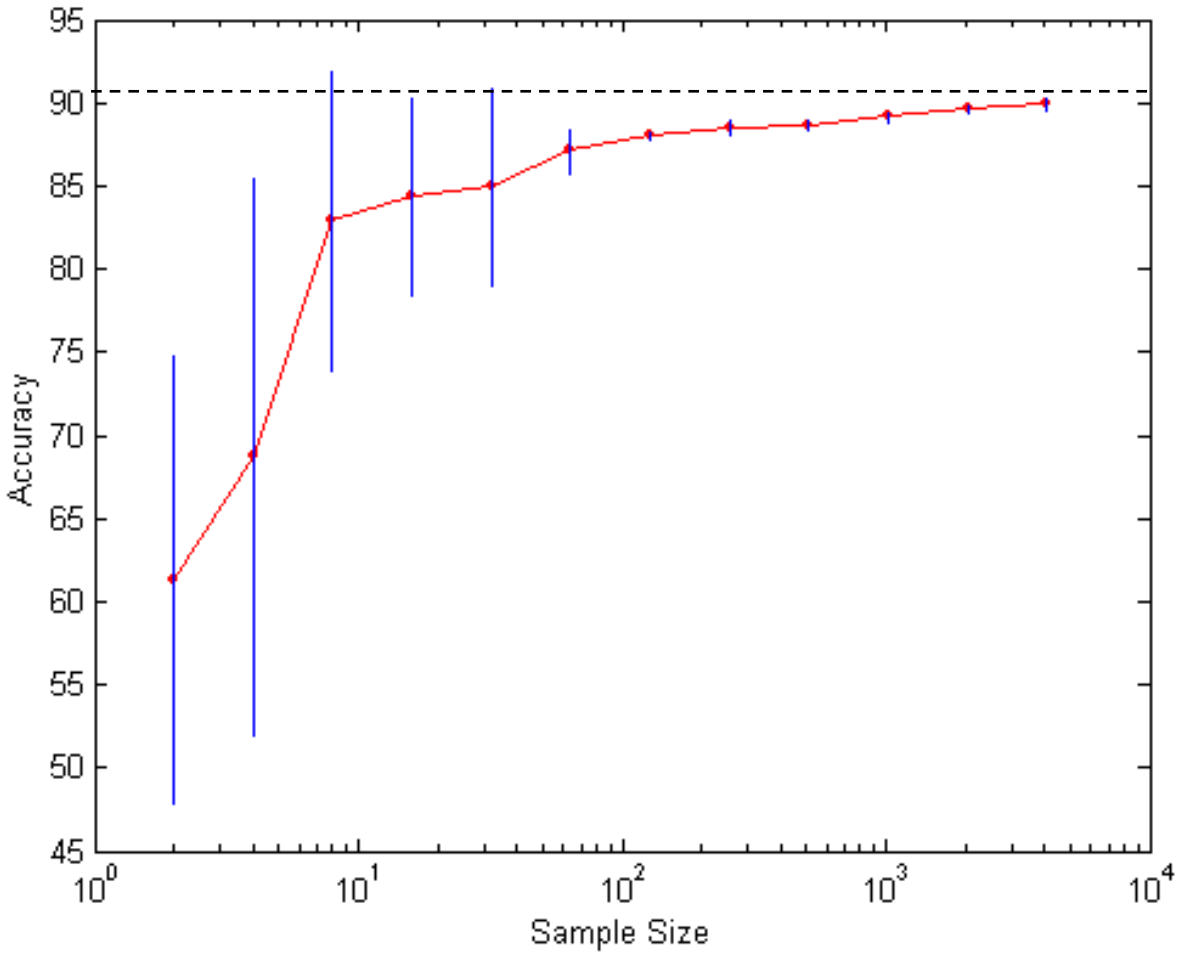
- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Methods for Performance Evaluation

- How to obtain a reliable estimate of performance?

- Performance of a model may depend on other factors besides the learning algorithm:
  - Class distribution
  - Cost of misclassification
  - Size of training and test sets

Learning Curve



- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve:
  - Arithmetic sampling (Langley, et al)
  - Geometric sampling (Provost et al)

Effect of small sample size:

- Bias in the estimate
- Variance of estimate

Methods of Estimation

- Holdout
  - Reserve 2/3 for training and 1/3 for testing
- Random subsampling
  - Repeated holdout
- Cross validation
  - Partition data into k disjoint subsets
  - k-fold: train on k-1 partitions, test on the remaining one
  - Leave-one-out: k=n
- Stratified sampling
  - oversampling vs undersampling
- Bootstrap
  - Sampling with replacement

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

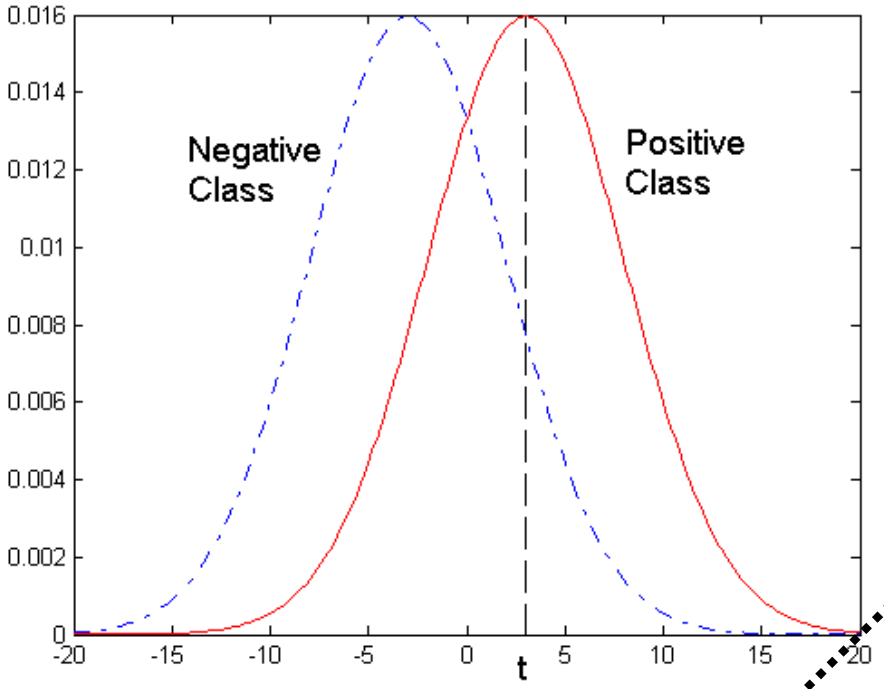
ROC (Receiver Operating Characteristic)

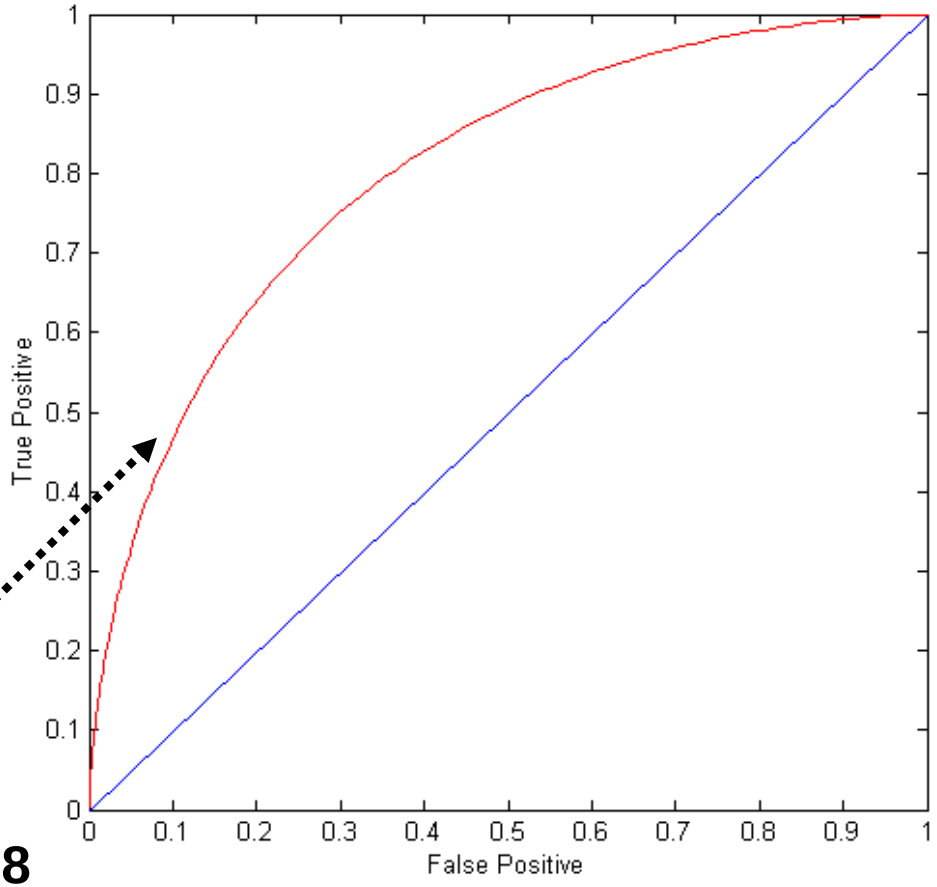
- Developed in 1950s for signal detection theory to analyze noisy signals
  - Characterize the trade-off between positive hits and false alarms
- ROC curve plots TP (on the y-axis) against FP (on the x-axis)
- Performance of each classifier represented as a point on the ROC curve
  - changing the threshold of algorithm, sample distribution or cost matrix changes the location of the point

ROC Curve

- 1-dimensional data set containing 2 classes (positive and negative)

- any points located at x > t is classified as positive





At threshold t:

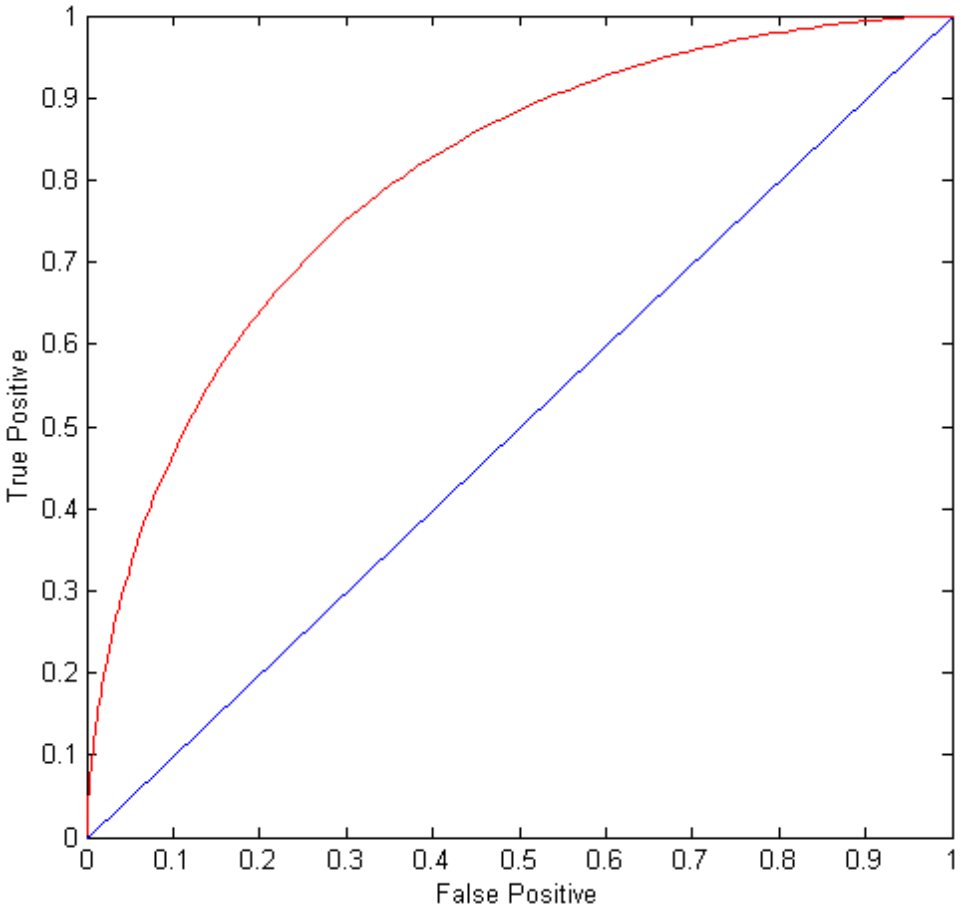
TP=0.5, FN=0.5, FP=0.12, FN=0.88

ROC Curve

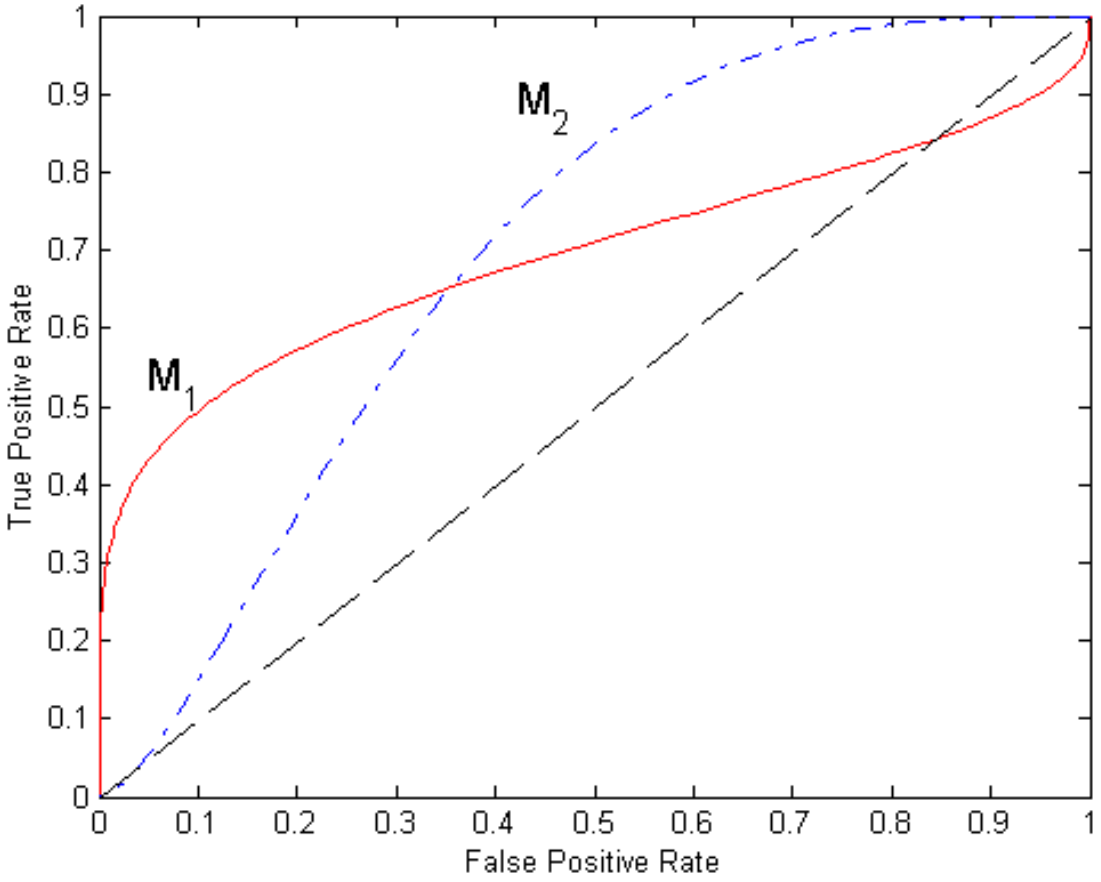
(TP,FP):

- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal

- Diagonal line:
  - Random guessing
  - Below diagonal line:
    - prediction is opposite of the true class



Using ROC for Model Comparison



- No model consistently outperform the other
  - M_1 is better for small FPR
  - M_2 is better for large FPR
- Area Under the ROC curve
  - Ideal:
    - Area = 1
  - Random guess:
    - Area = 0.5

How to Construct an ROC curve

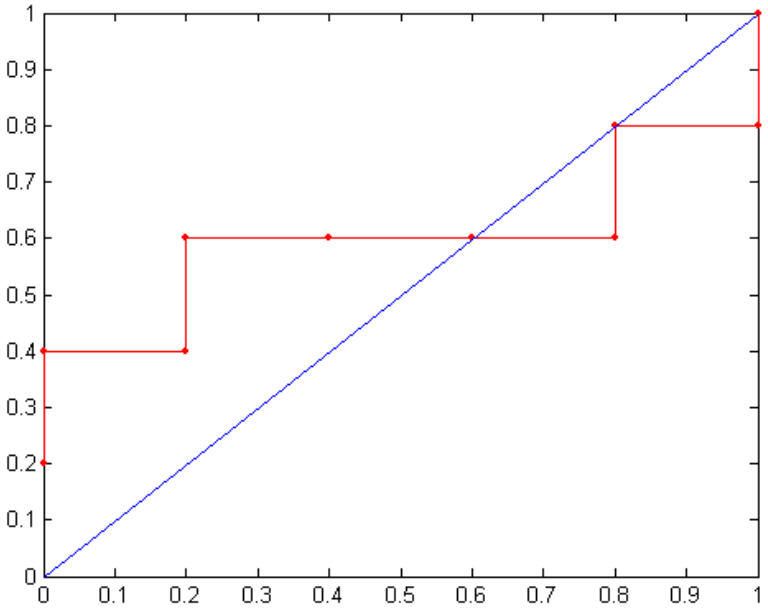
Instance	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

- Use classifier that produces posterior probability for each test instance P(+|A)
- Sort the instances according to P(+|A) in decreasing order
- Apply threshold at each unique value of P(+|A)
- Count the number of TP, FP, TN, FN at each threshold
- TP rate, TPR = TP/(TP+FN)
- FP rate, FPR = FP/(FP + TN)

How to construct an ROC curve

	Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=		0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
	TP	5	4	4	3	3	3	3	2	2	1	0
	FP	5	5	4	4	3	2	1	1	0	0	0
	TN	0	0	1	1	2	3	4	4	5	5	5
	FN	0	1	1	2	2	2	2	3	3	4	5
→	TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
→	FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

ROC Curve:



Test of Significance

- Given two models:
  - Model M1: accuracy = 85%, tested on 30 instances
  - Model M2: accuracy = 75%, tested on 5000 instances

- Can we say M1 is better than M2?
  - How much confidence can we place on accuracy of M1 and M2?
  - Can the difference in performance measure be explained as a result of random fluctuations in the test set?

Confidence Interval for Accuracy

- Prediction can be regarded as a Bernoulli trial
  - A Bernoulli trial has 2 possible outcomes
  - Possible outcomes for prediction: correct or wrong
  - Collection of Bernoulli trials has a Binomial distribution:
    - $x \sim Bin(N, p)$ x: number of correct predictions
    - e.g: Toss a fair coin 50 times, how many heads would turn up?
      Expected number of heads = $N \times p = 50 \times 0.5 = 25$

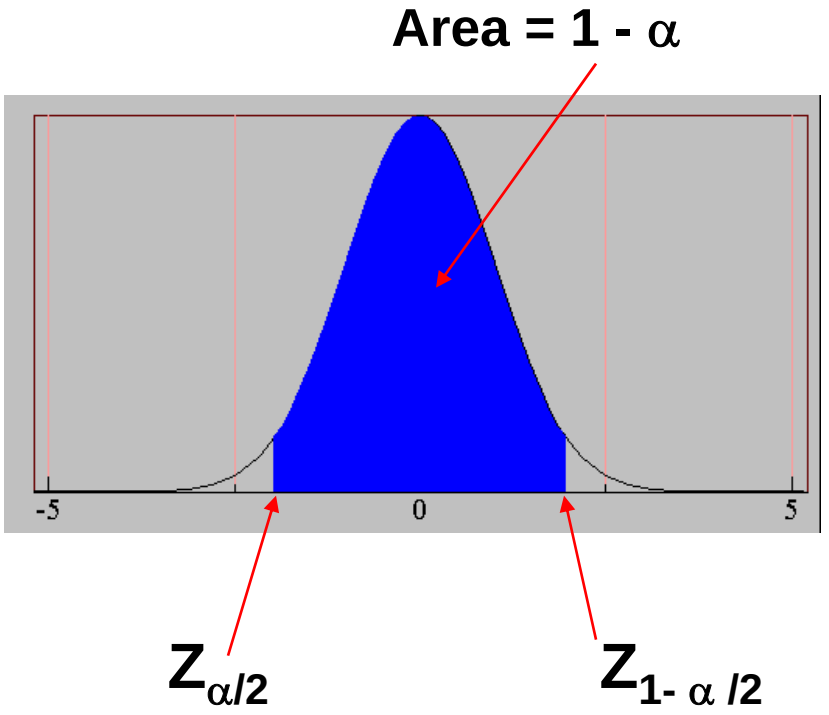
- Given x (# of correct predictions) or equivalently, acc=x/N, and N (# of test instances),

  Can we predict p (true accuracy of model)?

Confidence Interval for Accuracy

- For large test sets (N > 30),
  - acc has a normal distribution with mean p and variance p(1-p)/N

$$P(Z_{\alpha/2} < \frac{acc - p}{\sqrt{p(1-p)/N}} < Z_{1-\alpha/2}) = 1 - \alpha$$



- Confidence Interval for p:

$$p = \frac{2 \times N \times acc + Z^2_{\alpha/2} \pm \sqrt{Z^2_{\alpha/2} + 4 \times N \times acc - 4 \times N \times acc^2}}{2(N + Z^2_{\alpha/2})}$$

Confidence Interval for Accuracy

- Consider a model that produces an accuracy of 80% when evaluated on 100 test instances:
  - N=100, acc = 0.8
  - Let $1-\alpha = 0.95$ (95% confidence)
  - From probability table, $Z_{\alpha/2}=1.96$

1-α	Z
0.99	2.58
0.98	2.33
0.95	1.96
0.90	1.65

N	50	100	500	1000	5000
p(lower)	0.670	0.711	0.763	0.774	0.789
p(upper)	0.888	0.866	0.833	0.824	0.811

Comparing Performance of 2 Models

- Given two models, say M1 and M2, which is better?
  - M1 is tested on D1 (size=n1), found error rate = e_1
  - M2 is tested on D2 (size=n2), found error rate = e_2
  - Assume D1 and D2 are independent
  - If n1 and n2 are sufficiently large, then

$$e_1 \sim N(\mu_1, \sigma_1)$$
$$e_2 \sim N(\mu_2, \sigma_2)$$

  - Approximate: $\hat{\sigma}_i = \frac{e_i(1-e_i)}{n_i}$

Comparing Performance of 2 Models

- To test if performance difference is statistically significant: d = e1 – e2
  - d ~ $N(d_t,\sigma_t)$ where d_t is the true difference
  - Since D1 and D2 are independent, their variance adds up:

$$\sigma_t^2 = \sigma_1^2 + \sigma_2^2 \cong \hat{\sigma}_1^2 + \hat{\sigma}_2^2$$

$$= \frac{e1(1-e1)}{n1} + \frac{e2(1-e2)}{n2}$$

  - At (1-α) confidence level, $d_t = d \pm Z_{\alpha/2}\hat{\sigma}_t$

An Illustrative Example

- Given: M1: n1 = 30, e1 = 0.15
  M2: n2 = 5000, e2 = 0.25
- d = |e2 – e1| = 0.1 (2-sided test)

$$\hat{\sigma}_d = \frac{0.15(1-0.15)}{30} + \frac{0.25(1-0.25)}{5000} = 0.0043$$

- At 95% confidence level, $Z_{\alpha/2}$=1.96

$$d_t = 0.100 \pm 1.96 \times \sqrt{0.0043} = 0.100 \pm 0.128$$

=> Interval contains 0 => difference may not be statistically significant

Comparing Performance of 2 Algorithms

- Each learning algorithm may produce k models:
  - L1 may produce M11 , M12, …, M1k
  - L2 may produce M21 , M22, …, M2k
- If models are generated on the same test sets D1,D2, …, Dk (e.g., via cross-validation)
  - For each set: compute $d_j = e_{1j} – e_{2j}$
  - d_j has mean d_t and variance σ_t
  - Estimate:

$$\hat{\sigma}_t^2 = \frac{\sum_{j=1}^{k} (d_j - \bar{d})^2}{k(k-1)}$$

$$d_t = d \pm t_{1-\alpha,k-1} \hat{\sigma}_t$$