Artificial Neural Networks

Bayesian Classification

Outline

Main Classification Problem

Feature Selection

Feature Extraction

Bayesian Classification

Error Analysis

Machine Learning as Pattern Recognition

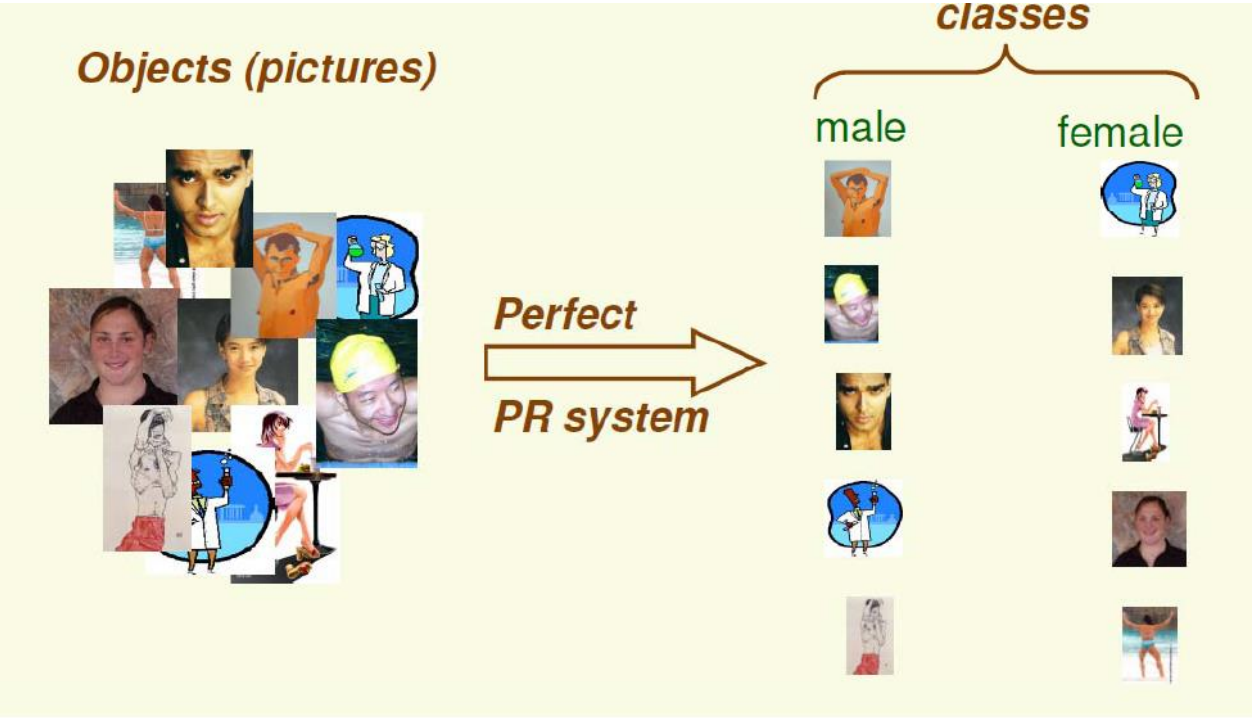
- Machine learning is defined as improving the performance of a system through experiments
- Question:
- Which information from an experiment can help in doing so?
- Answer:
- Characteristics of objects (patterns) which can be used for identifying them

Pattern Classification

- The performance metric in machine learning is better classification of objects using their characteristics (patterns)
- Therefore, we can say that the classifier assigns patterns to classes (instead of objects)

What is Pattern Recognition?

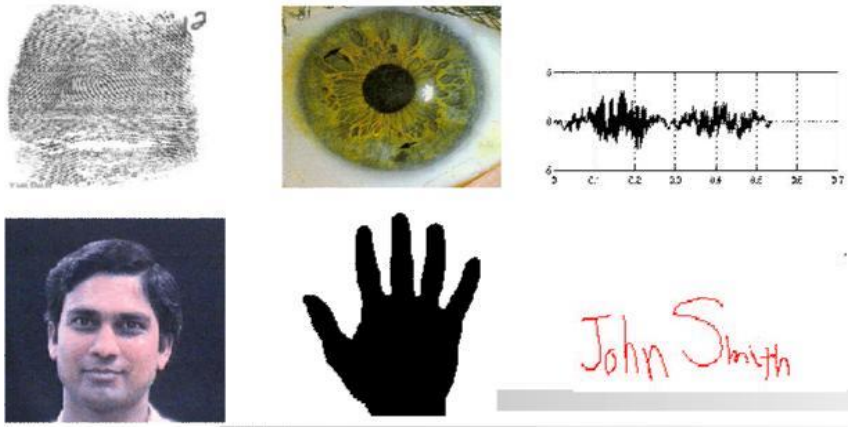
•Assigning an unknown pattern to one of several known categories (or classes).



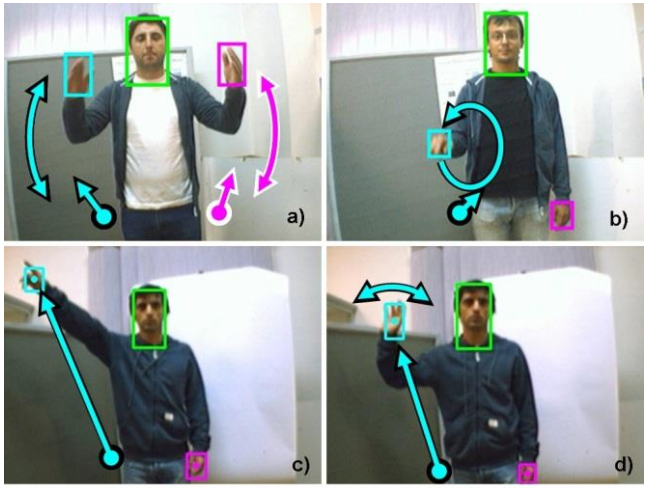
What is a Pattern?

- A pattern is a set of features representing an object or an event.

biometric patterns



hand gesture patterns



Pattern Class

- A class is a collection of "similar" objects.

Female



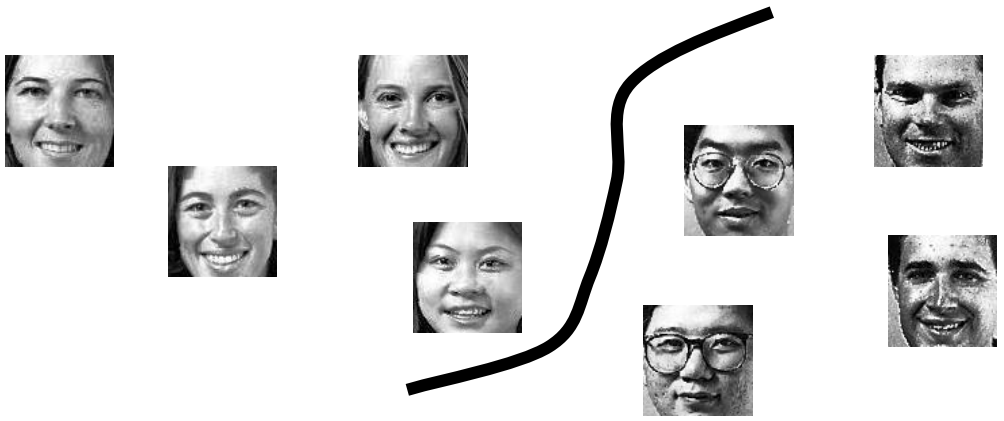

Male

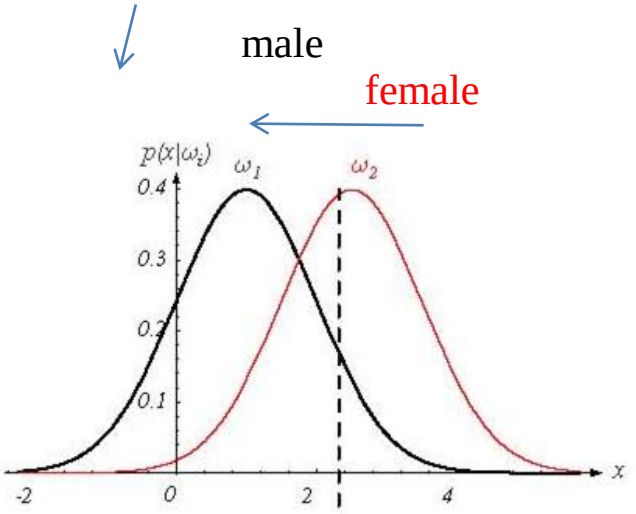



How do we model a Pattern Class?

- Typically, using a statistical model.
  - probability density function (e.g., Gaussian)

Gender Classification





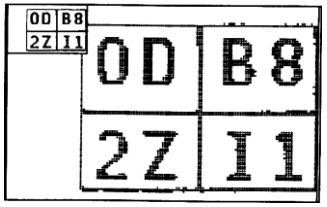
How do we model a Pattern Class? (cont'd)

- Key challenges:

  – Intra-class variability

The letter "T" in different typefaces

  – Inter-class variability



Letters/Numbers that look similar

Pattern Recognition: Main Objectives

- Hypothesize the models that describe each pattern class (e.g., recover the process that generated the patterns).

- Given a novel pattern, choose the best-fitting model for it and then assign it to the pattern class associated with the model.

Main Classification Approaches

***x**: input vector (pattern)*

***y**: class label (class)*

- *Generative*
  - –Model the joint probability, p(x, y)
  - –Make predictions by using Bayes rules to calculate p(ylx)
  - –Pick the most likely label y
- *Discriminative*
  - –Estimate p(ylx) directly (e.g., learn a direct map from inputs x to the class labels y)
  - –Pick the most likely label y

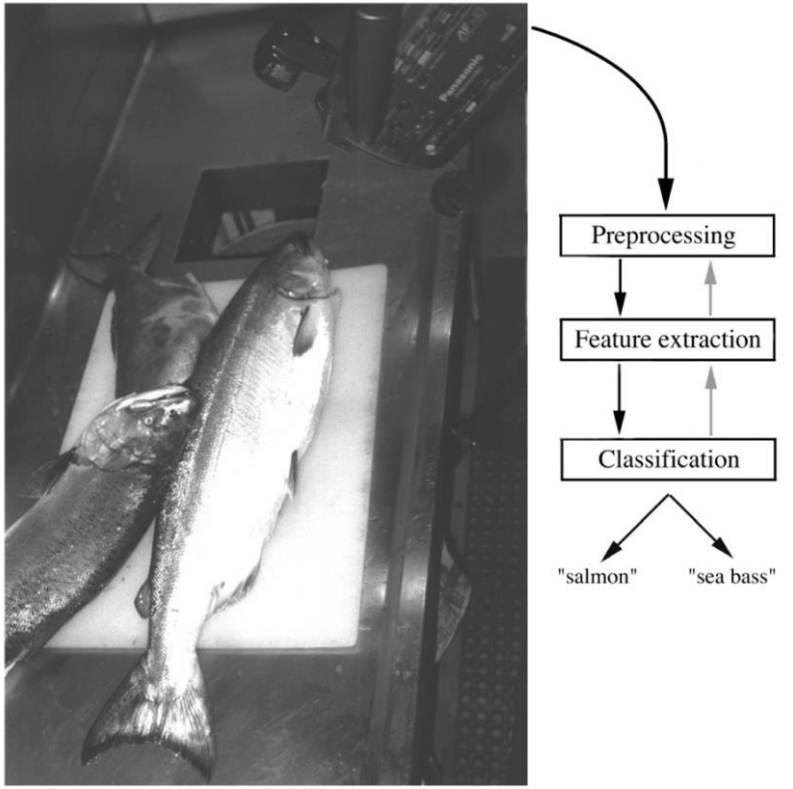
Complexity of PR – An Example

Problem: Sorting incoming fish on a conveyor belt.

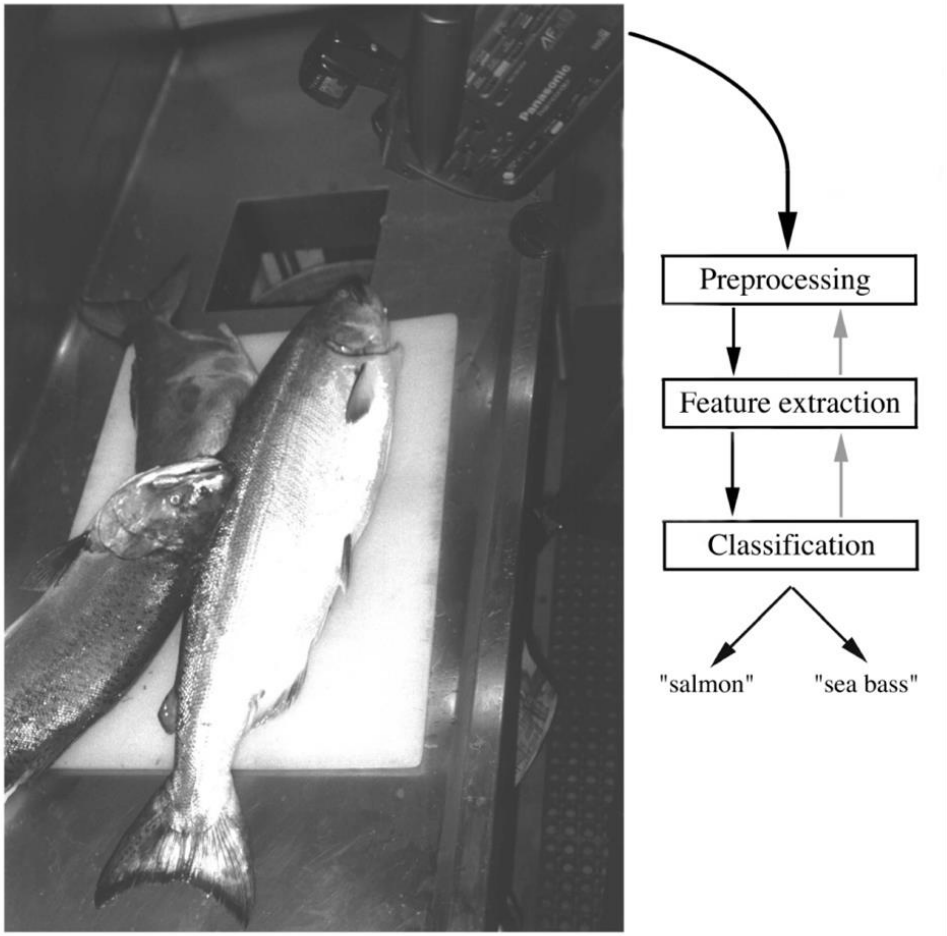
Assumption: Two kind of fish:

(1) sea bass

(2) salmon



Pre-processing Step



Example

(1) Image enhancement

(2) Separate touching or occluding fish

(3) Find the boundary of each fish

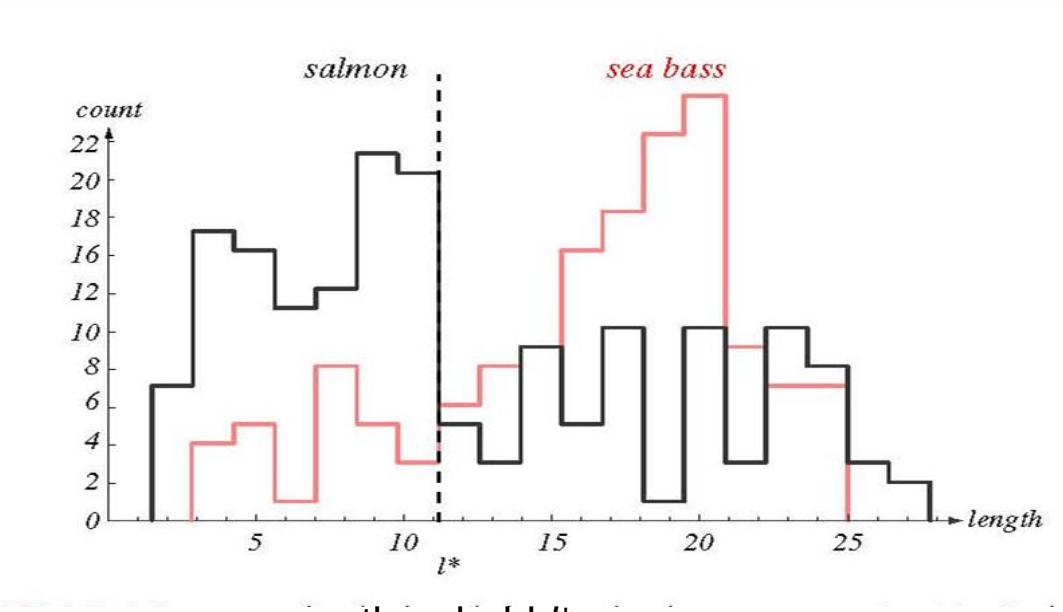
Feature Extraction

- Assume a fisherman told us that a sea bass is generally longer than a salmon.

- We can use length as a feature and decide between sea bass and salmon according to a threshold on length.

- How should we choose the threshold?

“Length” Histograms

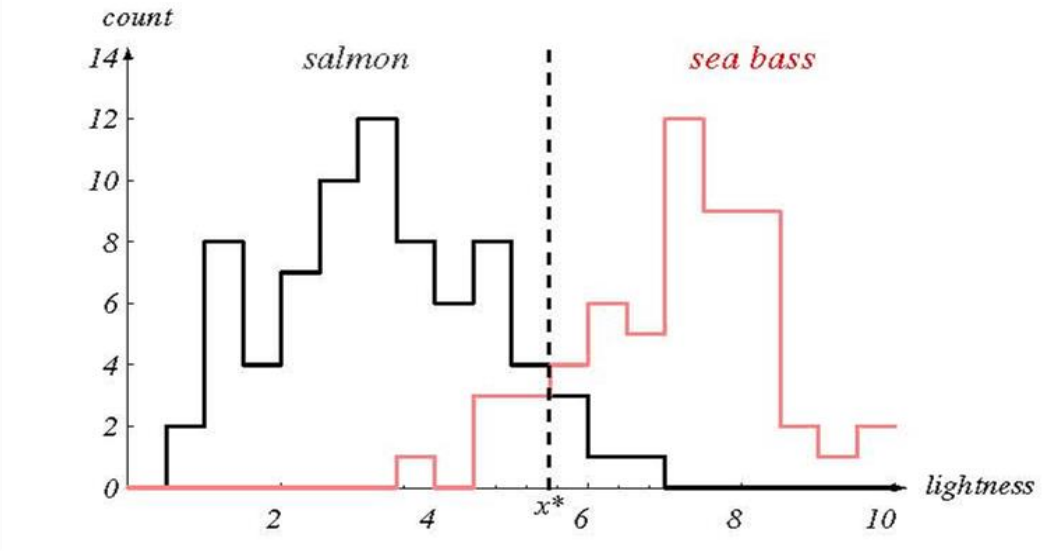


threshold *l**

•Even though sea bass is longer than salmon on the average, there are many examples of fish where this observation does not hold.

“Average Lightness” Histograms

•Consider a different feature such as “average lightness”



threshold *x**

•It seems easier to choose the threshold x^* but we still cannot make a perfect decision.

Multiple Features

• To improve recognition accuracy, we might have to use more than one features at a time.

–Single features might not yield the best performance.

–Using combinations of features might yield better performance.

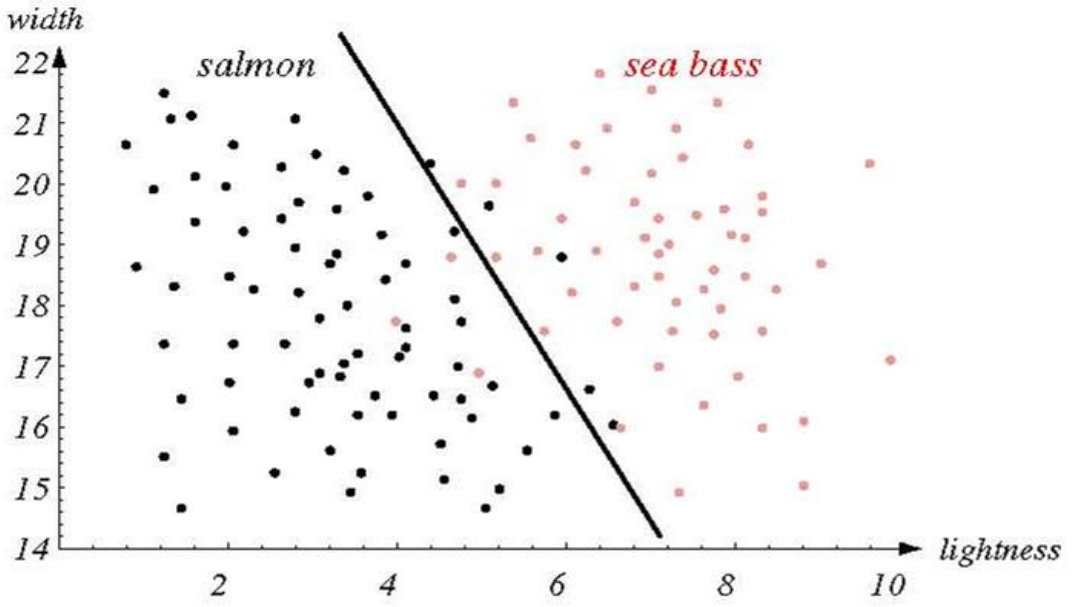
$$x_1 : lightness$$

$$x_2 : width$$

•How many features should we choose?

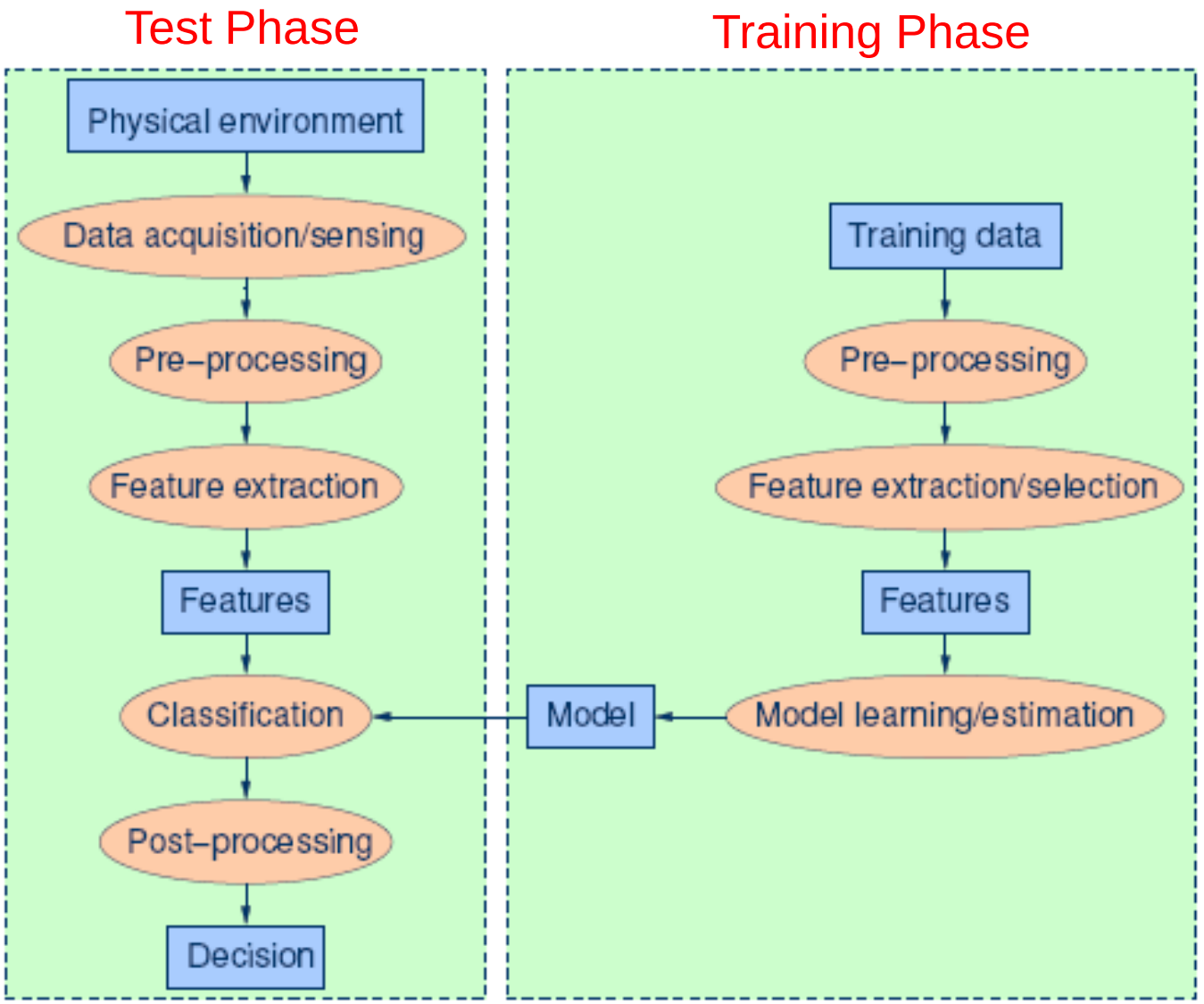
Classification

•Partition the *feature space* into two regions by finding the decision boundary that minimizes the error.



•How should we find the optimal decision boundary?

PR System – Two Phases

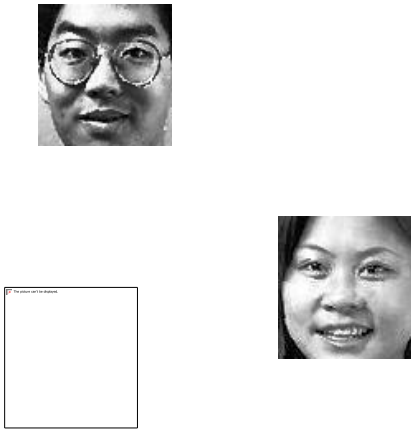


Training/Test data

- How do we know that we have collected an adequately large and representative set of examples for training/testing the system?

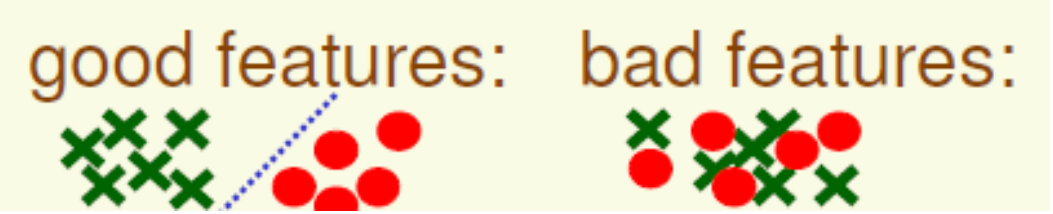
Training Set

Test Set ?



Feature Extraction

- How to choose a good set of features?
  - Discriminative features



  - Invariant features (e.g., translation, rotation and scale)
- Are there ways to automatically learn which features are best ?

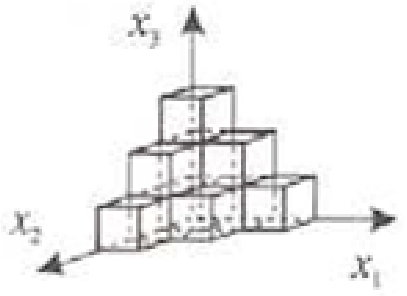
How Many Features?

- Does adding more features always improve performance?
  - It might be difficult and computationally expensive to extract certain features.
  - Correlated features might not improve performance.
  - "Curse" of dimensionality.

Curse of Dimensionality

- Adding too many features can, paradoxically, lead to a worsening of performance.

–Divide each of the input features into a number of intervals, so that the value of a feature can be specified approximately by saying in which interval it lies.



–If each input feature is divided into **M** divisions, then the total number of cells is $\mathbf{M}^{\mathbf{d}}$ (***d:*** # of features).

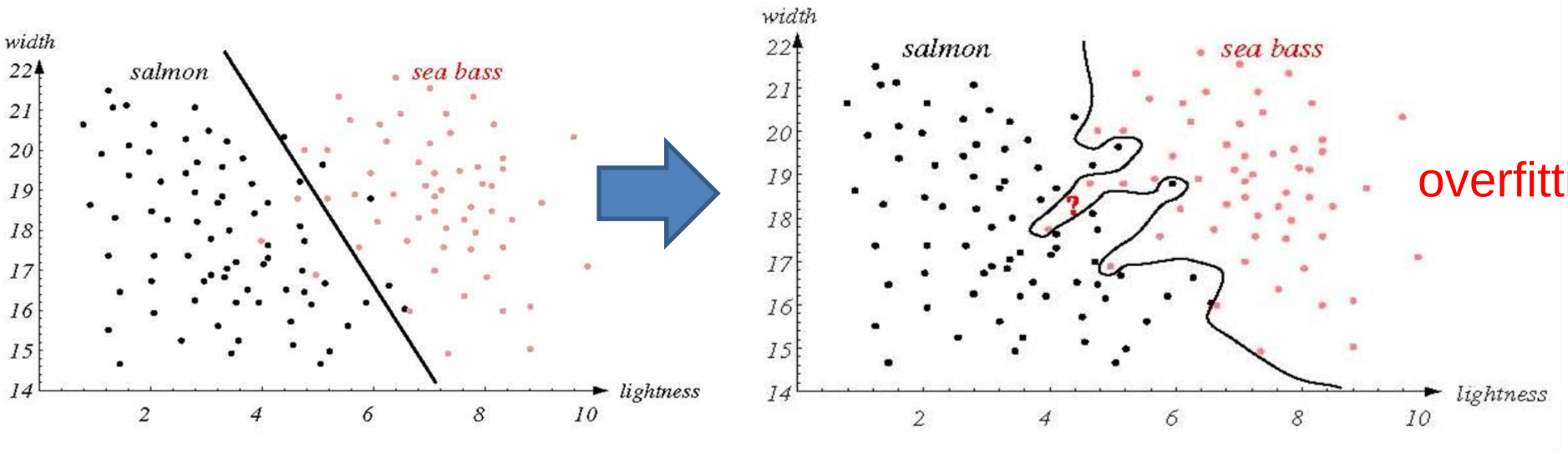
–Since each cell must contain at least one point, the number of training data grows exponentially with ***d***.

Missing Features

- Certain features might be missing (e.g., due to occlusion).
- How should we train the classifier with missing features ?
- How should the classifier make the best decision with missing features ?

Complexity

- We can get perfect classification performance on the training data by choosing complex models.
- Complex models are tuned to the particular training samples, rather than on the characteristics of the true model.



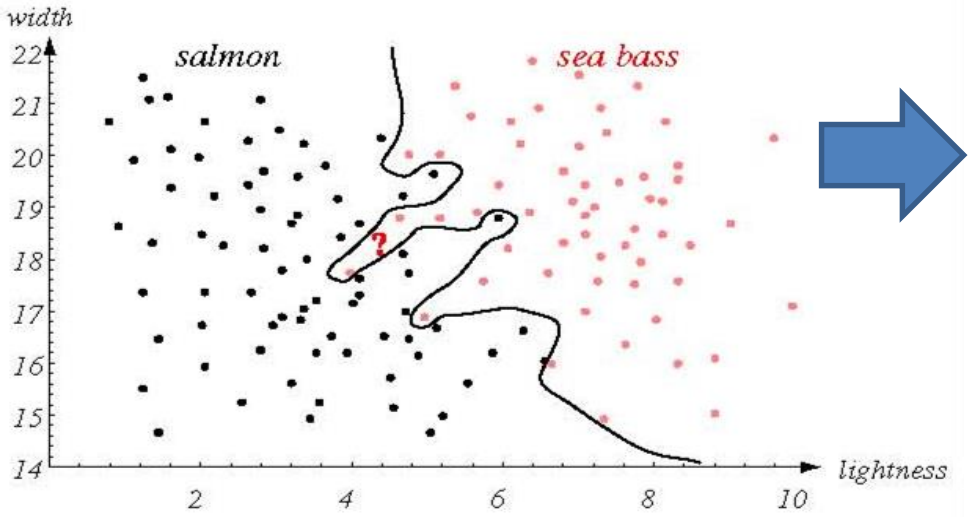
overfitting

How well can the model generalize to unknown samples?

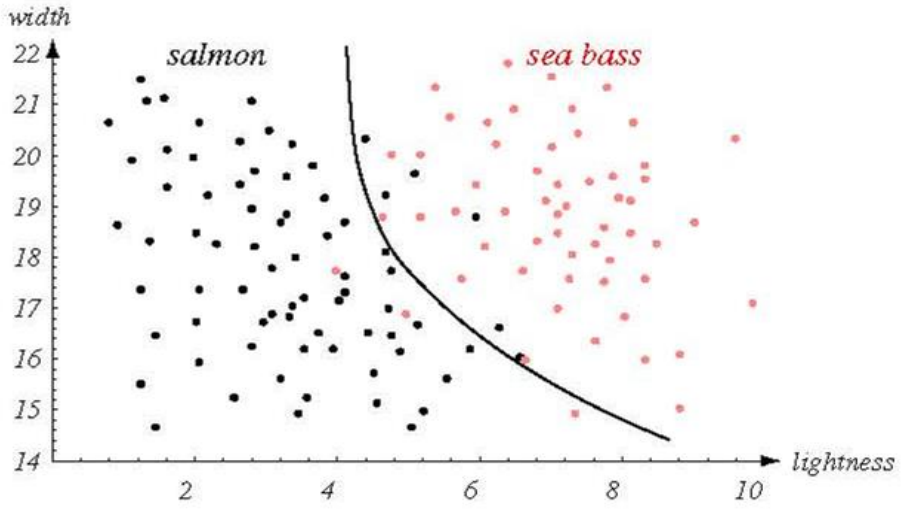
Generalization

- Generalization is defined as the ability of a classifier to produce correct results on novel patterns.
- How can we improve generalization performance ?
  - More training examples (i.e., better model estimates).
  - Simpler models usually yield better performance.

complex model



simpler model

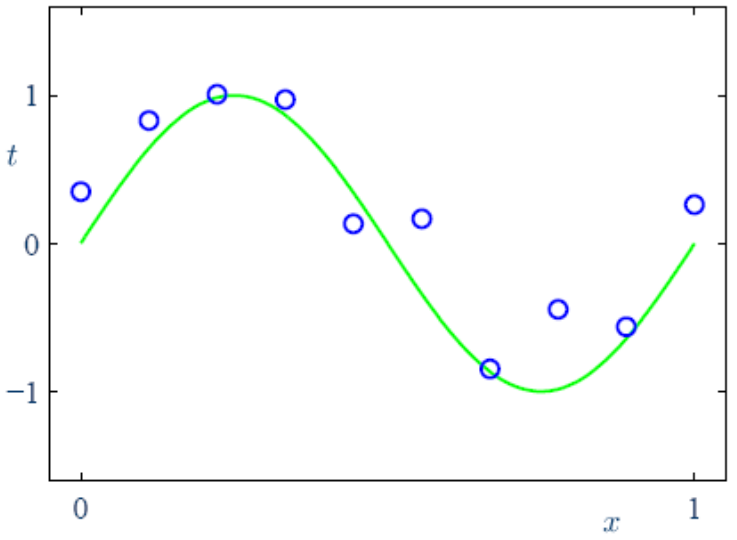


More on model complexity

•Consider the following 10 sample points (blue circles) assuming some noise.

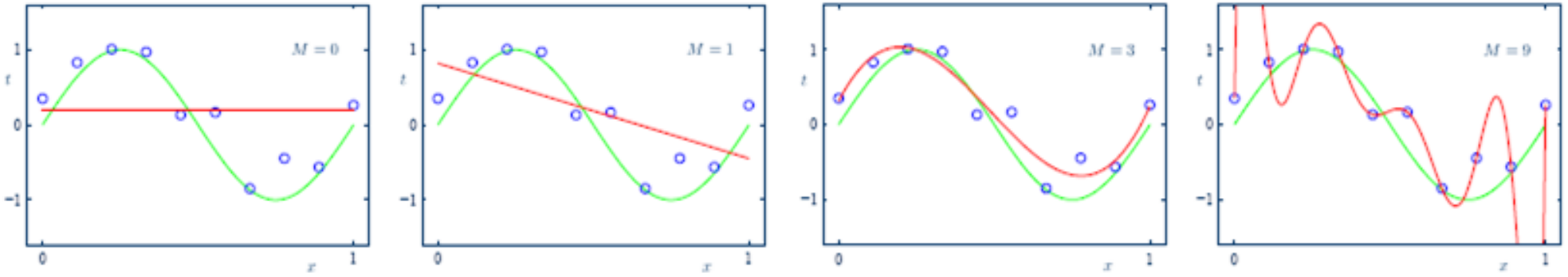
•Green curve is the true function that generated the data.

•Approximate the true function from the sample points.



More on model complexity (cont'd)

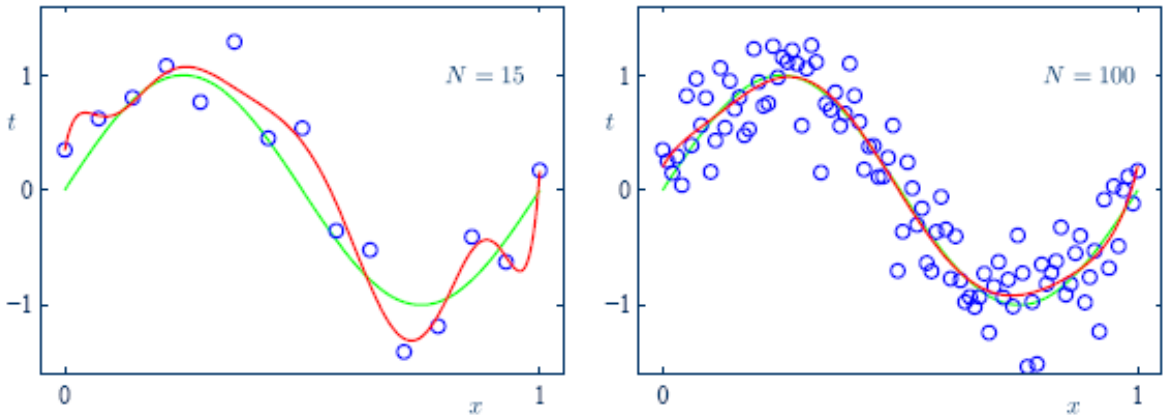
Polynomial curve fitting: polynomials having various orders, shown as red curves, fitted to the set of 10 sample points.



(a) 0'th order polynomial (b) 1'st order polynomial (c) 3'rd order polynomial (d) 9'th order polynomial

More on complexity (cont'd)

Polynomial curve fitting: 9'th order polynomials fitted to 15 and 100 sample points.



(a) 15 sample points (b) 100 sample points

Cost of miss-classifications

- Consider the fish classification example; there are two possible classification errors:

  (1) Deciding the fish was a sea bass when it was a salmon.

  (2) Deciding the fish was a salmon when it was a sea bass.

- Are both errors equally important ?

Cost of miss-classifications (cont'd)

- Suppose the fish packing company knows that:
  - Customers who buy salmon will object vigorously if they see sea bass in their cans.
  - Customers who buy sea bass will not be unhappy if they occasionally see some expensive salmon in their cans.

- How does this knowledge affect our decision?

Bayesian Decision Theory

- Design classifiers to make decisions subject to minimizing an expected ”risk”.
  - The simplest risk is the classification error (i.e., assuming that misclassification costs are equal).
  - When misclassification costs are not equal, the risk can include the cost associated with different misclassifications.

Terminology

- Category name ω *(class label):*
  - e.g., ω_1 for sea bass, ω_2 for salmon

- Probabilities $P(\omega_1)$ and $P(\omega_2)$ *(priors):*
  - e.g., prior knowledge of how likely is to get a sea bass or a salmon

- Probability density function $p(x)$ *(evidence):*
  - e.g., how frequently we will measure a pattern with feature value x (e.g., x corresponds to lightness)

Terminology (cont'd)

- Conditional probability density $p(x/\omega_j)$ (*likelihood*) :

–e.g., how frequently we will measure a pattern with feature value *x* given that the pattern belongs to class ω_j

e.g., lightness distributions between salmon/sea-bass populations

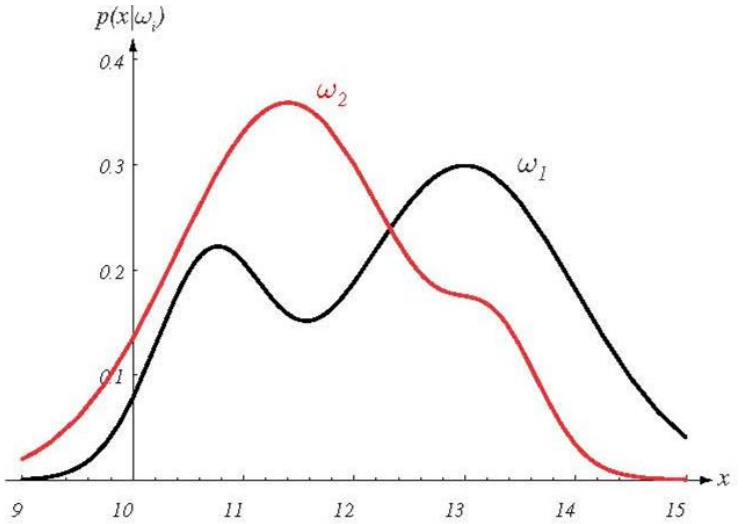


FIGURE 2.1. Hypothetical class-conditional probability density functions show the probability density of measuring a particular feature value *x* given the pattern is in category ω_i. If *x* represents the lightness of a fish, the two curves might describe the difference in lightness of populations of two types of fish. Density functions are normalized, and thus the area under each curve is 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons,

Terminology (cont'd)

- Conditional probability $P(\omega_j/x)$ *(posterior) :*

  – e.g., the probability that the fish belongs to class ω_j given feature x.

- Ultimately, we are interested in computing $P(\omega_j/x)$ for each class ω_j.

Decision Rule Using Prior Probabilities Only

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

$$P(error) = \begin{cases} P(\omega_1) \ \ if \ \ we\, decide\, \omega_2 \\ P(\omega_2) \ if \ \ we\, decide\, \omega_1 \end{cases}$$

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

- Favours the most likely class.
- This rule will be making the same decision all times.

–i.e., optimum if no other information is available

Decision Rule Using Conditional Probabilities

•Using Bayes' rule:

$$P(\omega_j / x) = \frac{p(x/\omega_j)P(\omega_j)}{p(x)} = \frac{likelihood \ \times \ prior}{evidence}$$

where $p(x) = \sum_{j=1}^{2} p(x/\omega_j)P(\omega_j)$ (i.e., scale factor – sum of probs = 1)

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

or

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

or

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

likelihood ratio threshold

Probability of Error

- The probability of error is defined as:

$$P(error/x) = \begin{cases} P(\omega_1/x) & if\ we\ decide\ \omega_2 \\ P(\omega_2/x) & if\ we\ decide\ \omega_1 \end{cases}$$

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

- What is the average probability error?

$$P(error) = \int_{-\infty}^{\infty} P(error, x)dx = \int_{-\infty}^{\infty} P(error/x)\,p(x)dx$$

- The Bayes rule is optimum, that is, it minimizes the average probability error!

Where do Probabilities come from?

- There are two competitive answers:

(1) **Relative frequency** (objective) approach.

– Probabilities can only come from experiments.

(2) **Bayesian** (subjective) approach.

– Probabilities may reflect degree of belief and can be based on opinion.

Example (objective approach)

- Classify cars whether they are more or less than $50K:
  - Classes: C_1 if price > $50K, C_2 if price <= $50K
  - Features: x, the height of a car

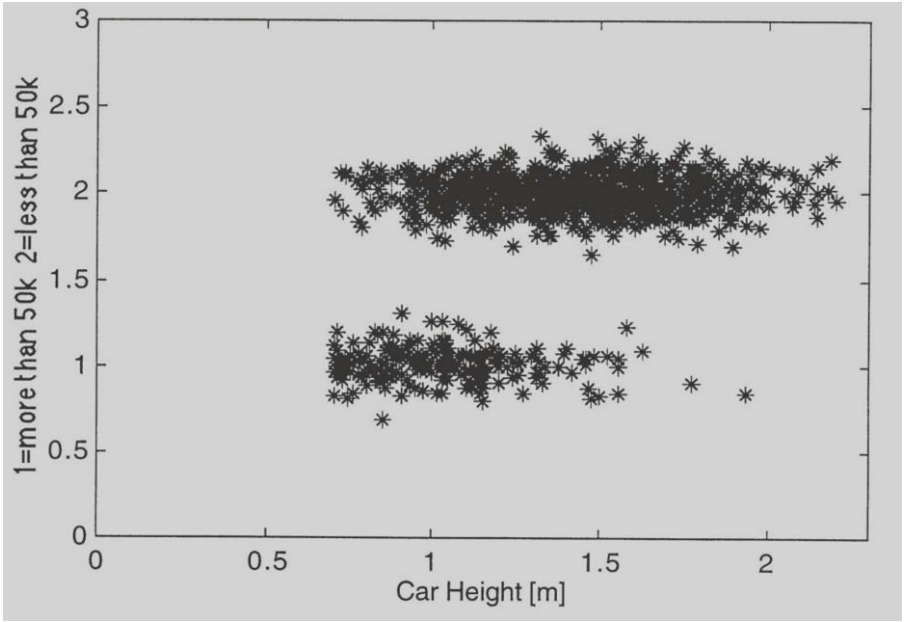
- Use the Bayes' rule to compute the posterior probabilities:

$$P(C_i / x) = \frac{p(x / C_i)P(C_i)}{p(x)}$$

- We need to estimate $p(x/C_1)$, $p(x/C_2)$, $P(C_1)$, $P(C_2)$

Example (cont'd)

- Collect data
  - Ask drivers how much their car was and measure height.
- Determine prior probabilities $P(C_1)$, $P(C_2)$
  - e.g., 1209 samples: #C_1=221 #C_2=988



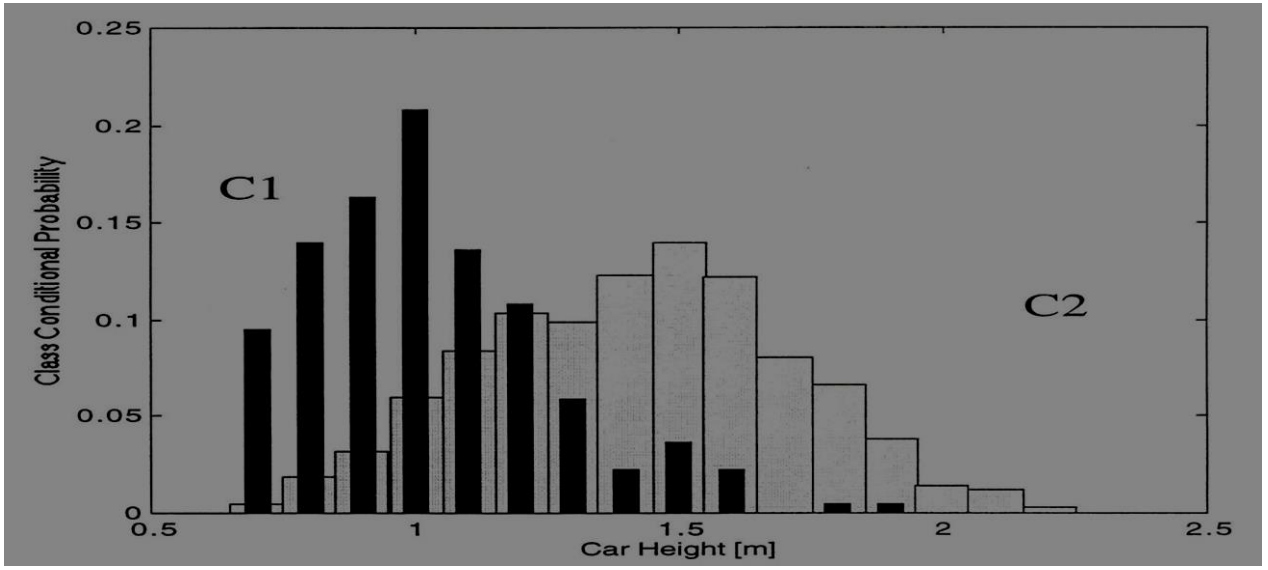
$$P(C_1) = \frac{221}{1209} = 0.183$$

$$P(C_2) = \frac{988}{1209} = 0.817$$

Example (cont'd)

- Determine class conditional probabilities (*likelihood*)
  - Discretize car height into bins and use normalized histogram

$p(x / C_i)$



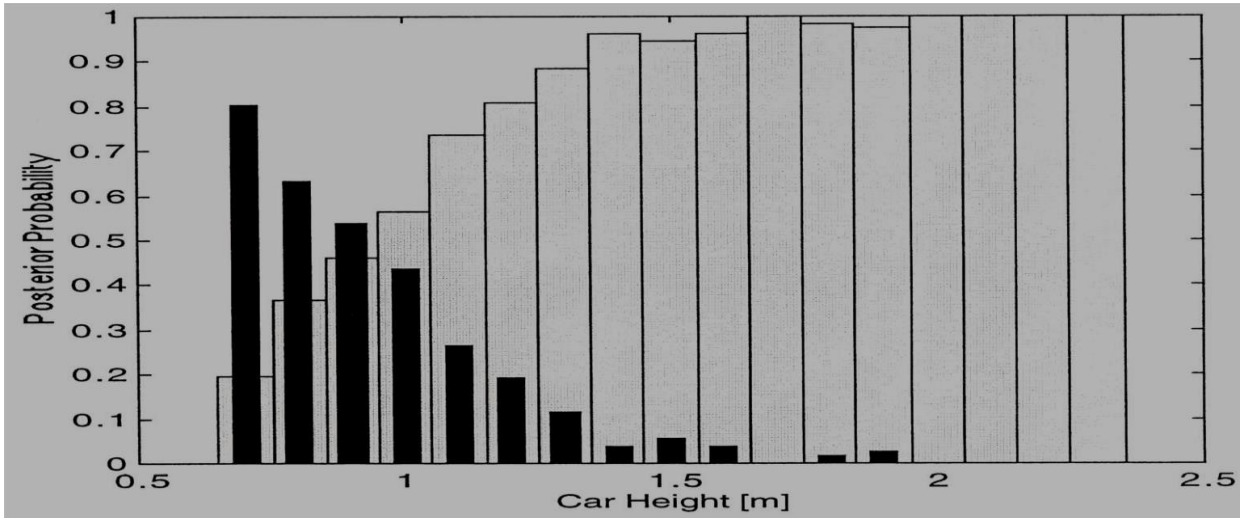
Example (cont'd)

- Calculate the posterior probability for each bin:

$$P(C_1 / x = 1.0) = \frac{p(x = 1.0 / C_1)\, P(C_1)}{p(x = 1.0 / C_1)\, P(C_1) + p(x = 1.0 / C_2)\, P(C_2)} =$$

$$= \frac{0.2081 * 0.183}{0.2081 * 0.183 + 0.0597 * 0.817} = 0.438$$

$P(C_i / x)$



A More General Theory

- Use more than one features.
- Allow more than two categories.
- Allow actions other than classifying the input to one of the possible categories (e.g., rejection).
- Employ a more general error function (i.e., expected "risk") by associating a "cost" (based on a "loss" function) with different errors.

Terminology

- Features form a vector $\mathbf{x} \in R^d$
- A set of *c* categories $\omega_1, \omega_2, \ldots, \omega_c$
- A finite set of ***l*** actions $\alpha_1, \alpha_2, \ldots, \alpha_l$
- A *loss* function $\lambda(\alpha_i / \omega_j)$

–the cost associated with taking action α_i when the correct classification category is ω_j

- Bayes rule (using vector notation):

$$P(\omega_j / \mathbf{x}) = \frac{p(\mathbf{x} / \omega_j) P(\omega_j)}{p(\mathbf{x})}$$

$$where \quad p(\mathbf{x}) = \sum_{j=1}^{c} p(\mathbf{x} / \omega_j) P(\omega_j)$$

Conditional Risk (or Expected Loss)

- Suppose we observe **x** and take action α_i

- The **conditional risk** (or **expected loss**) with taking action α_i is defined as:

$$R(a_i / \mathbf{x}) = \sum_{j=1}^{c} \lambda(a_i / \omega_j) P(\omega_j / \mathbf{x})$$

Overall Risk

- Suppose $\alpha(\mathbf{x})$ is a general decision rule that determines which action $\alpha_1, \alpha_2, \ldots, \alpha_l$ to take for every **x**.

- The overall risk is defined as:

$$R = \int R(a(\mathbf{x}) / \mathbf{x})\, p(\mathbf{x}) d\mathbf{x}$$

- The optimum decision rule is the *Bayes rule*

Overall Risk (cont'd)

- The *Bayes rule* minimizes *R* by:
  - (i) Computing $R(\alpha_i/\mathbf{x})$ for every α_i given an **x**
  - (ii) Choosing the action α_i with the minimum $R(\alpha_i/\mathbf{x})$

- The resulting minimum R^* is called *Bayes risk* and is the best (i.e., optimum) performance that can be achieved:

$$R^* = \min R$$

Example: Two-category classification

- Define
  - α_1: decide ω_1
  - α_2: decide ω_2
  - $\lambda_{ij} = \lambda(\alpha_i / \omega_j)$

- The conditional risks are:

$$R(a_i / \mathbf{x}) = \sum_{j=1}^{c} \lambda(a_i / \omega_j) P(\omega_j / \mathbf{x})$$

$$R(a_1/\mathbf{x}) = \lambda_{11} P(\omega_1/\mathbf{x}) + \lambda_{12} P(\omega_2/\mathbf{x})$$
$$R(a_2/\mathbf{x}) = \lambda_{21} P(\omega_1/\mathbf{x}) + \lambda_{22} P(\omega_2/\mathbf{x})$$

Example: Two-category classification (cont'd)

- Minimum risk decision rule:

$$\textbf{Decide } \omega_1 \text{ if } R(a_1/\mathbf{x}) < R(a_2/\mathbf{x}); \text{ otherwise decide } \omega_2$$

or

$$\textbf{Decide } \omega_1 \text{ if } (\lambda_{21} - \lambda_{11})P(\omega_1/\mathbf{x}) > (\lambda_{12} - \lambda_{22})P(\omega_2/\mathbf{x}); \text{ otherwise decide } \omega_2$$

or (i.e., using likelihood ratio)

$$\textbf{Decide } \omega_1 \text{ if } \frac{p(\mathbf{x}/\omega_1)}{p(\mathbf{x}/\omega_2)} > \frac{(\lambda_{12} - \lambda_{22})}{(\lambda_{21} - \lambda_{11})} \frac{P(\omega_2)}{P(\omega_1)}; \text{ otherwise decide } \omega_2$$

likelihood ratio threshold

Special Case:
Zero-One Loss Function

•Assign the same loss to all errors:

$$\lambda(a_i/\omega_j) = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases}$$

•The conditional risk corresponding to this loss function:

$$R(a_i/\mathbf{x}) = \sum_{j=1}^{c} \lambda(a_i/\omega_j) P(\omega_j/\mathbf{x}) = \sum_{i \neq j} P(\omega_j/\mathbf{x}) = 1 - P(\omega_i/\mathbf{x})$$

Special Case:
Zero-One Loss Function (cont'd)

- The decision rule becomes:

$$\textbf{Decide } \omega_1 \text{ if } R(a_1/\mathbf{x}) < R(a_2/\mathbf{x}); \text{ otherwise decide } \omega_2$$

or $$\textbf{Decide } \omega_1 \text{ if } 1 - P(\omega_1/\mathbf{x}) < 1 - P(\omega_2/\mathbf{x}); \text{ otherwise decide } \omega_2$$

or $$\textbf{Decide } \omega_1 \text{ if } P(\omega_1/\mathbf{x}) > P(\omega_2/\mathbf{x}); \text{ otherwise decide } \omega_2$$

- The overall risk turns out to be the average probability error!

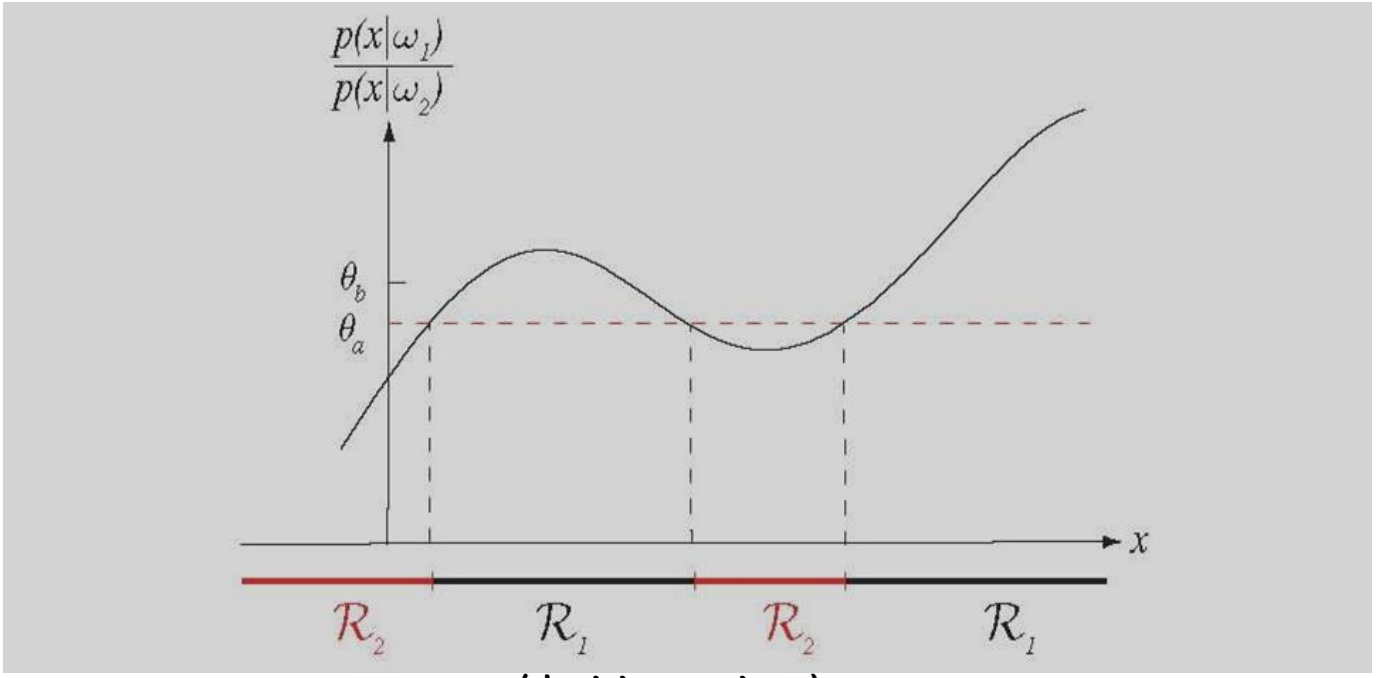
Example

Assuming general loss:

$$\textbf{Decide } \omega_1 \text{ if } \frac{p(\mathbf{x}/\omega_1)}{p(\mathbf{x}/\omega_2)} \gtrless \frac{(\lambda_{12}-\lambda_{22})}{(\lambda_{21}-\lambda_{11})}\frac{P(\omega_2)}{P(\omega_1)}\text{; otherwise decide } \omega_2$$

Assuming zero-one loss:

Decide ω_1 if $p(x/\omega_1)/p(x/\omega_2) > P(\omega_2)/P(\omega_1)$ otherwise **decide** ω_2



(decision regions)

$$\theta_a = P(\omega_2)/P(\omega_1)$$

$$\theta_b = \frac{P(\omega_2)(\lambda_{12}-\lambda_{22})}{P(\omega_1)(\lambda_{21}-\lambda_{11})}$$

assume: $\lambda_{12} > \lambda_{21}$