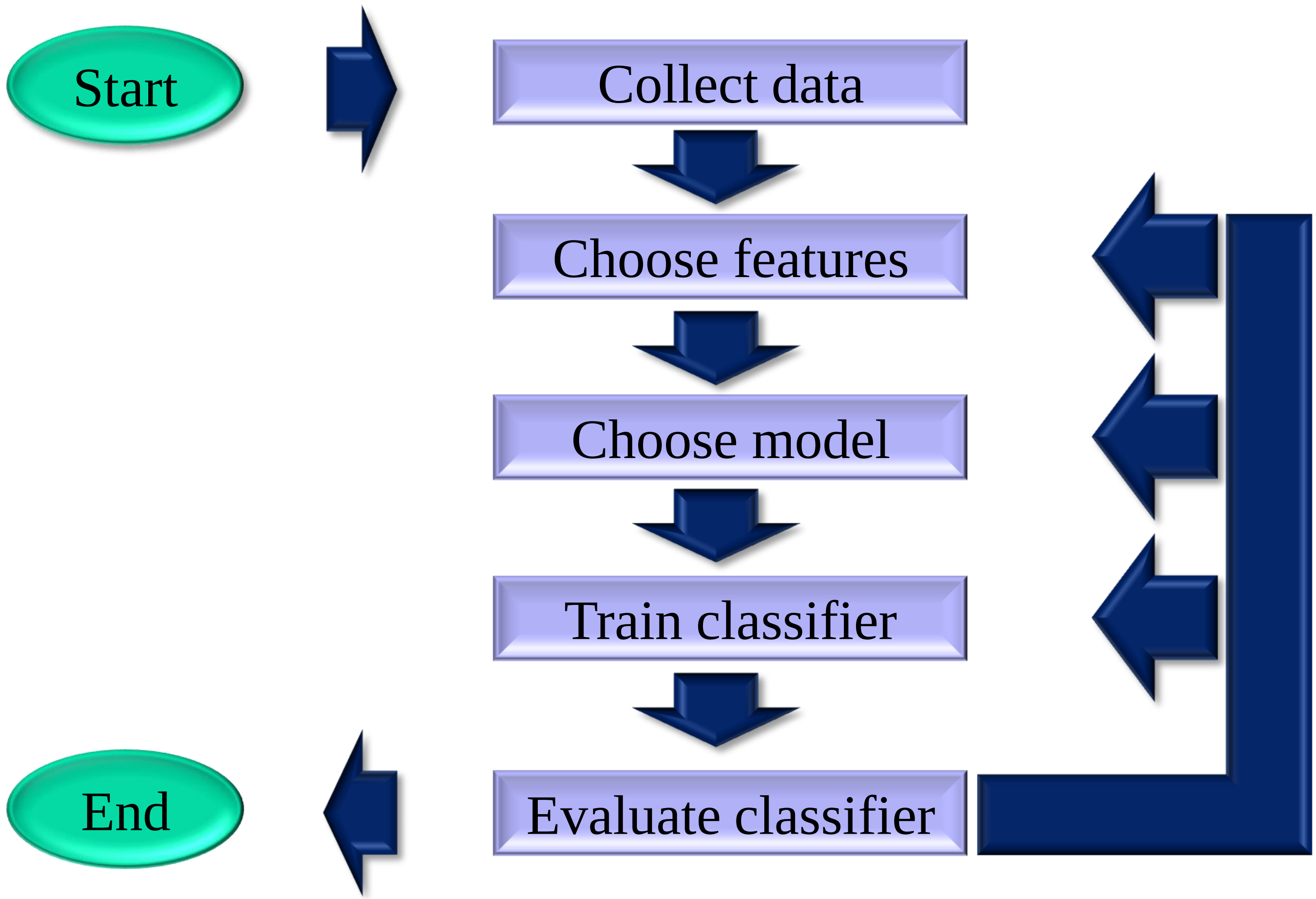
Prof. Marios Savvides

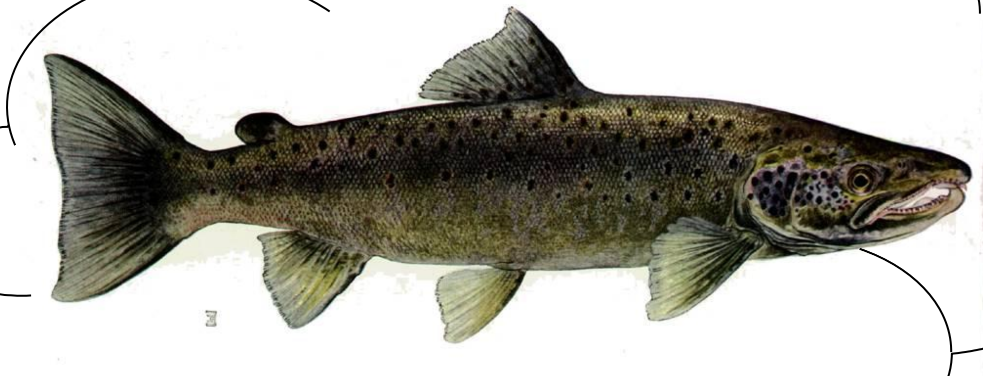
Pattern Recognition Theory

Lecture 1: Decision Theory I

Classifier Design Cycle



Our First Classification Problem …



Salmon

Bass

Steps to Good Classification

Image Processing

- De-noising, image enhancement
- Edge detection
- Segmentation

Feature Extraction

- Width?
- Height?
- Texture?
- Colour?

Classification

- 2-class problem?
- Multi-class problem?

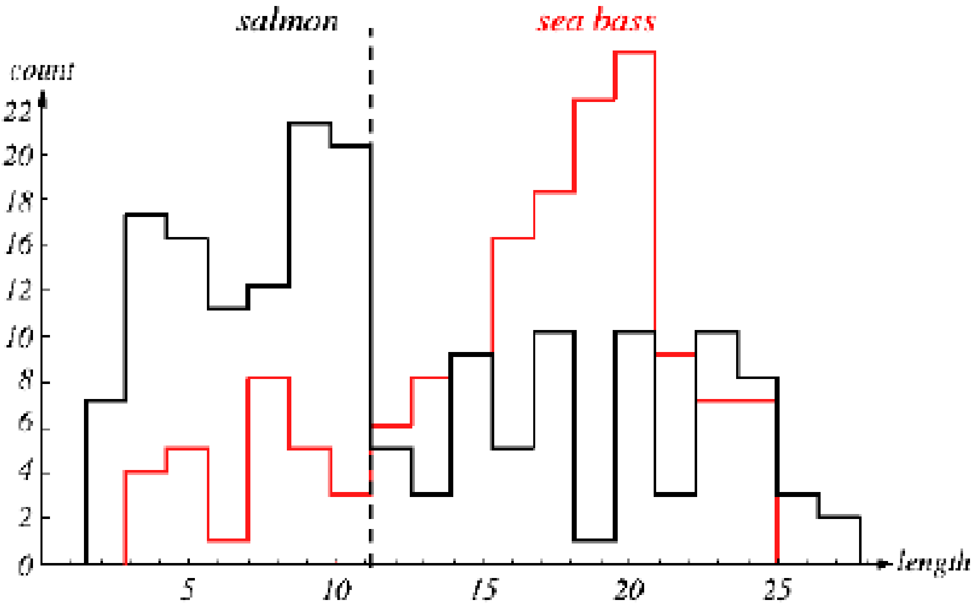
What's a Feature Space?...

- What's a feature?
  - A feature is a distinctive characteristic or quality of the object
- Combine more than one feature to get → D-dim **feature vector**
- The D-dimensional space thus defined → **feature space**

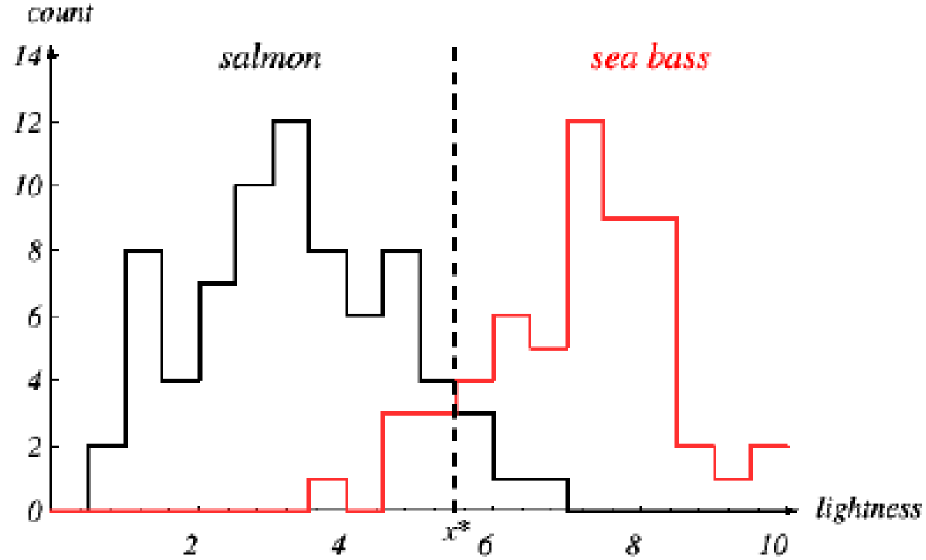
$$\mathbf{x} = \begin{bmatrix} \boxed{\text{Length}} \\ \boxed{\text{Lightness}} \\ \boxed{\text{Texture}} \end{bmatrix}$$

Here D = 3

How Features Help



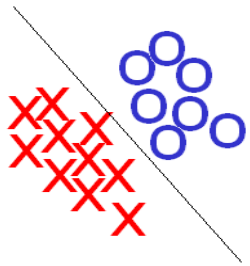
$$x_1 = length$$



$$x_2 = lightness$$

Properties of Features

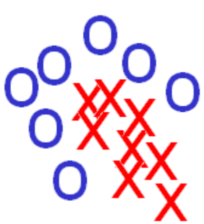
- The quality of the feature vector is related to its ability to discriminate samples from different classes



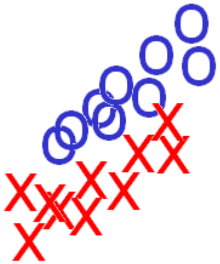
Good Features – Linearly Separable

Bad Features – Not Discriminating

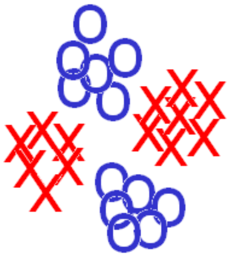
- Other Properties



Non-linearly Separable



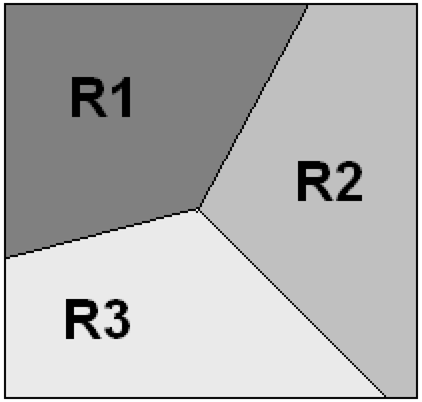
Positively Correlated

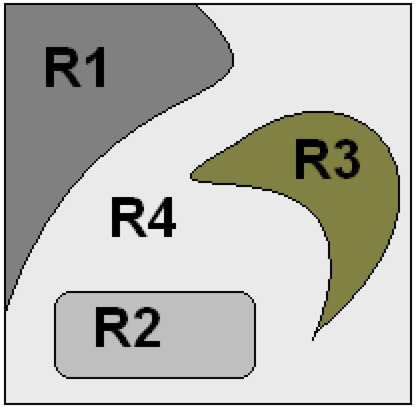


Multi-modal

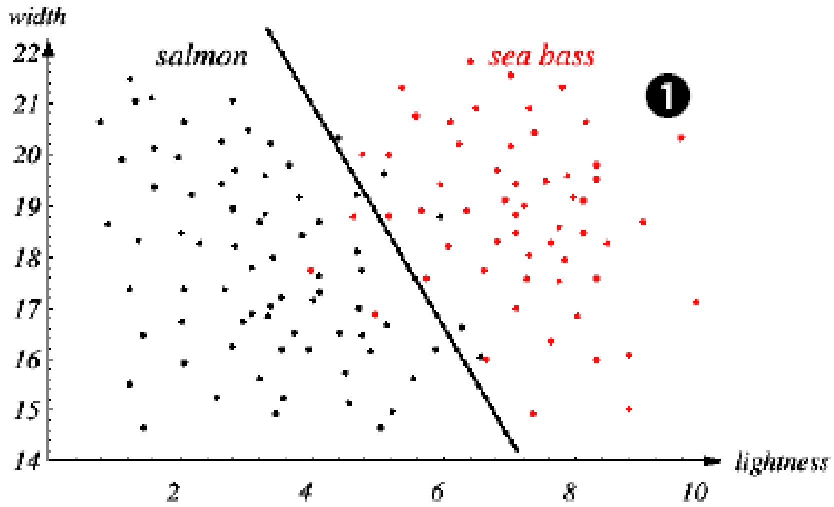
Classifiers

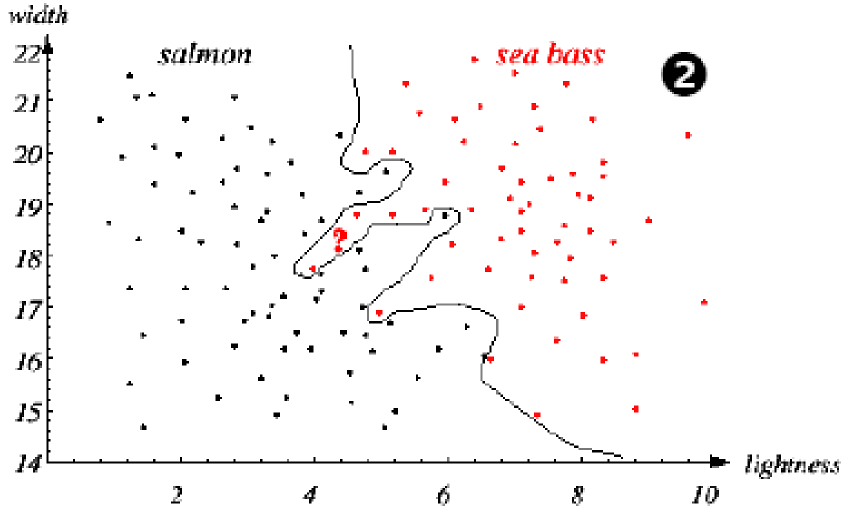
- A classifier is designed to partition the feature space into class-corresponding **decision regions**
  - What criterion do I have to minimize?
  - What cost function to use?
  - Are all errors equals?
- Borders between the decision regions are called **decision boundaries**





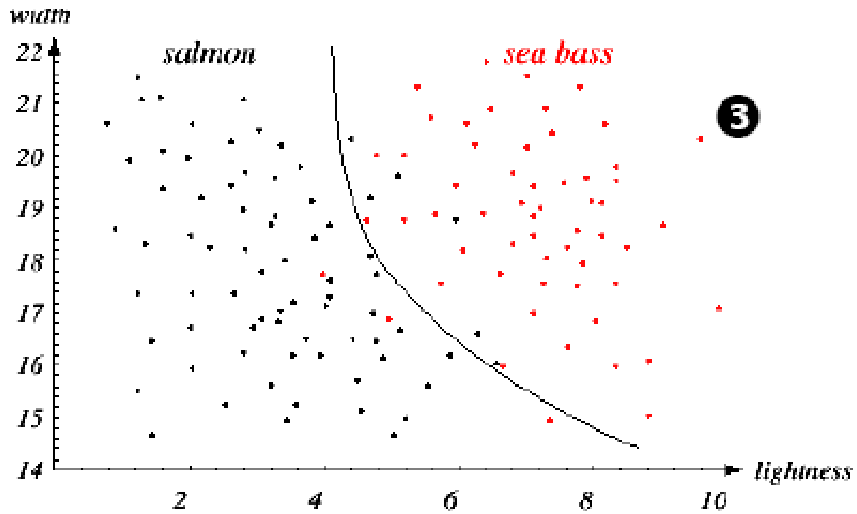
Decision Boundaries





Which decision boundary?

- **1**: Linear boundary with significant training error
- **2**: Nonlinear complex boundary with zero training error
- **3**: Simple non-linear boundary with small training error

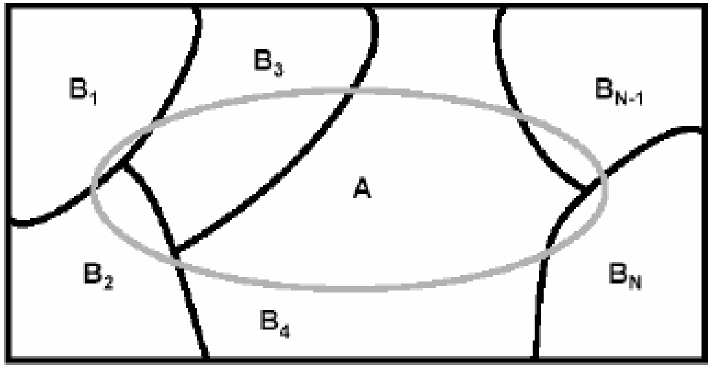


Our First Classifier

- Minimum error classifier
  - Goal is to design a classifier to partition the decision space to minimize the number of errors (misclassifications)
  - During classifier design, we assume that the underlying probabilistic structure (class dependence) is known
    - i.e. the probability
      f(feature vector **x** given class ω) is known →f(**x**| ω)

A Short Detour: Bayes Rule

Given that event 'A' has occurred, what is the probability that one of the event 'B's occur?



Rev. Thomas Bayes

(1702-1761)

$$P(B_i \mid A) = \frac{P\left(A \cap B_i\right)}{P(A)} = \frac{P\left(A \mid B_i\right) P(B_i)}{\sum_{j}^{N} P\left(A \mid B_j\right) P(B_j)}$$

A Few Definitions…

Likelihood: The conditional probability of observing a feature value of *x* given that the correct class is ω_i

Posterior Probability: The conditional probability of correct class being ω_i given that feature value *x* has been observed

Prior Probability: The probability of class ω_i

$$P(\omega_1)+P(\omega_2)+\ldots+P(\omega_c)=1$$

$$P(\omega_i \mid x) = \frac{P(x \cap \omega_i)}{P(x)} = \frac{P(x \mid \omega_i)P(\omega_i)}{\sum_{k=1}^{C} P(x \mid \omega_k)P(\omega_k)}$$

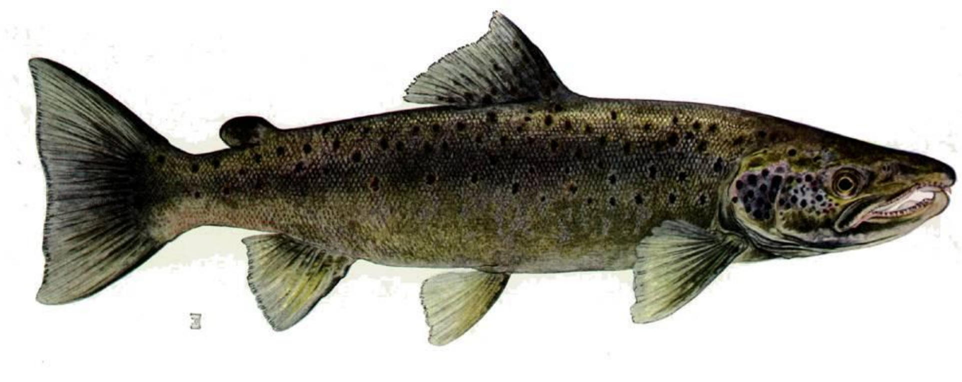
Evidence : The total probability of observing the feature value of *x*

Posterior Probability For Classification

$$P(\omega_i \mid x) = \frac{P\left(x \cap \omega_i\right)}{P(x)} = \frac{P\left(x \mid \omega_i\right) P(\omega_i)}{\sum_{k=1}^{C} P\left(x \mid \omega_k\right) P(\omega_k)}$$

- Bayes Classifiers decide on the class that has the **largest posterior probability**
- It can be shown that Bayes classifiers are statistically the best classifiers one can construct i.e. they are minimum error classifiers (optimal)

And Now, Back to our Fish…



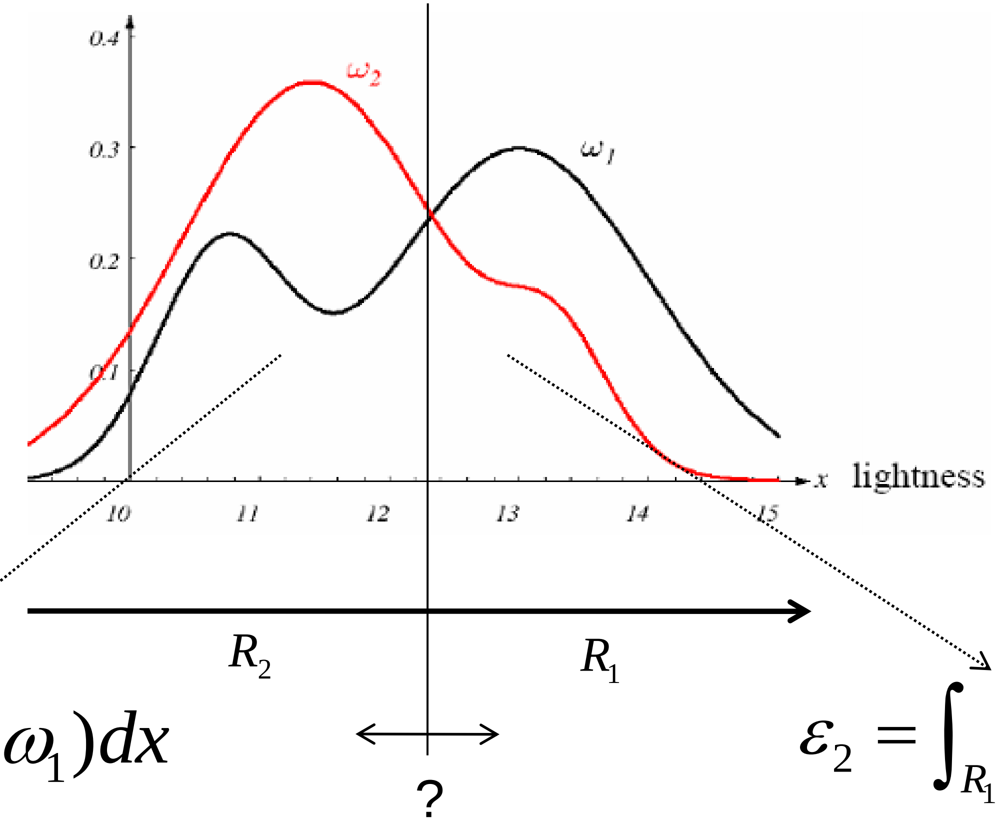
- We will build a **Bayes classifier** i.e. a minimum error classifier, to distinguish salmon samples from bass samples

- For this classifier, consider only one of the **features** i.e. the **lightness** of the fish

Minimum Probability of Error

$\varepsilon = P(error \mid class)$ probability of assigning x to the wrong class ω

$P_e = P(\omega_1)\varepsilon_1 + P(\omega_2)\varepsilon_2$ total probability of error



$$\varepsilon_1 = \int_{R_2} f(x \mid \omega_1)dx \qquad \varepsilon_2 = \int_{R_1} f(x \mid \omega_2)dx$$

Where to put the threshold?

Minimum Error Classifier

We want to minimize P_e

$$
\begin{aligned}
P_e &= P(\omega_1)\varepsilon_1 + P(\omega_2)\varepsilon_2 \\
&= P(\omega_1)\int_{R_2} f(x|\omega_1)dx + P(\omega_2)\int_{R_1} f(x|\omega_2)dx \\
&= P(\omega_1)\{1 - \int_{R_1} f(x|\omega_1)dx)\} + P(\omega_2)\int_{R_1} f(x|\omega_2)dx \\
&= P(\omega_1) + \{P(\omega_2)\int_{R_1} f(x|\omega_2)dx - P(\omega_1)\int_{R_1} f(x|\omega_1)dx\} \\
&= P(\omega_1) + \{\int_{R_1} P(\omega_2)f(x|\omega_2) - P(\omega_1)f(x|\omega_1)\}dx
\end{aligned}
$$

as small as possible – this difference is negative within R_1

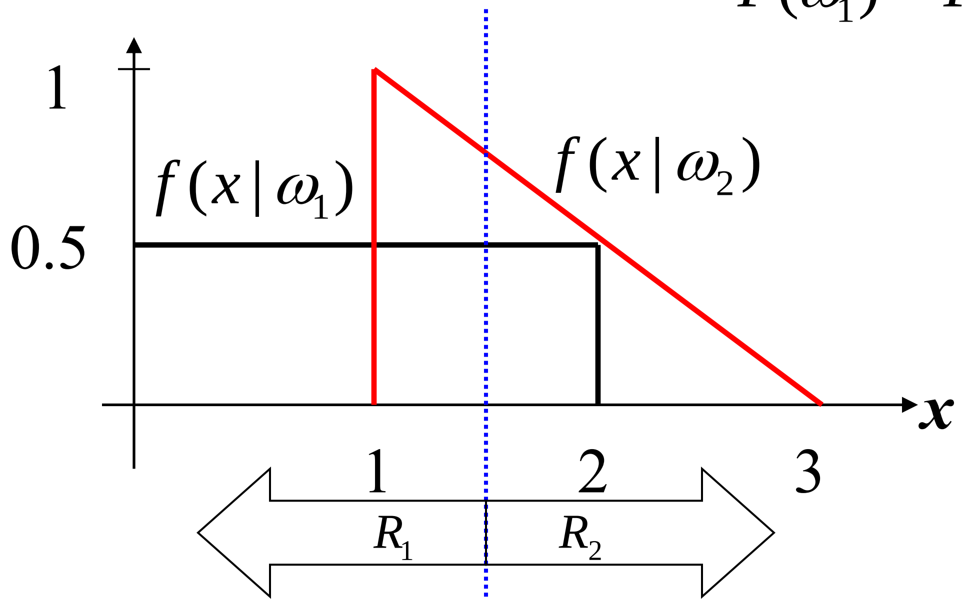
i.e., if $P(\omega_2)f(x|\omega_2) - P(\omega_1)f(x|\omega_1) < 0 \longrightarrow \omega_1$

Minimum Error Classifier

$$\left\{P(\omega_1)f(x|\omega_1)-P(\omega_2)f(x|\omega_2)\right\} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 0$$

Example : 2-Class Classifier Error

$$P(\omega_1) = P(\omega_2) = 0.5$$

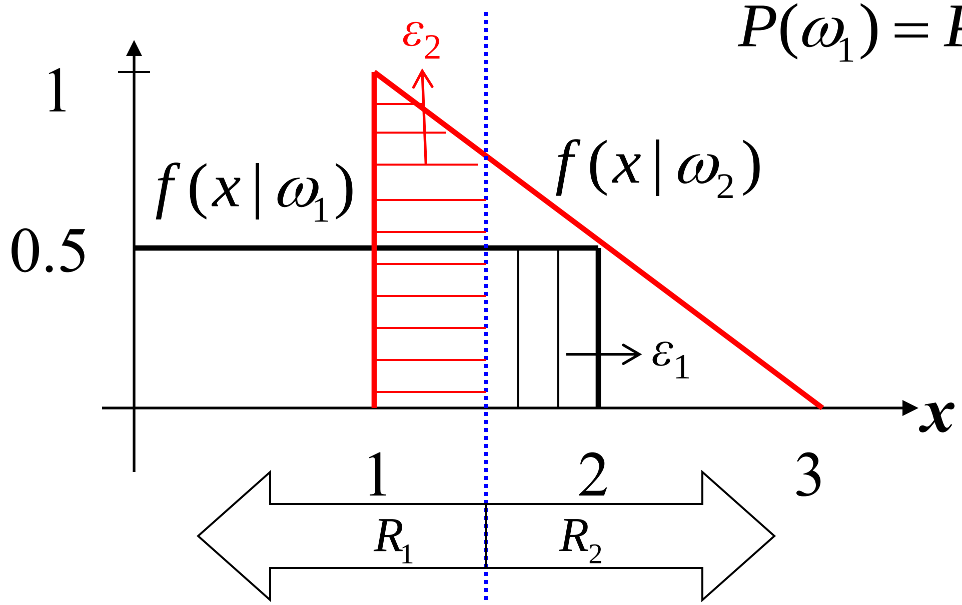


The Classifier assigns:

x $\leq$ 1.5 to class 1

x > 1.5 to class 2

Example : 2-Class Classifier Error



$$P(\omega_1) = P(\omega_2) = 0.5$$

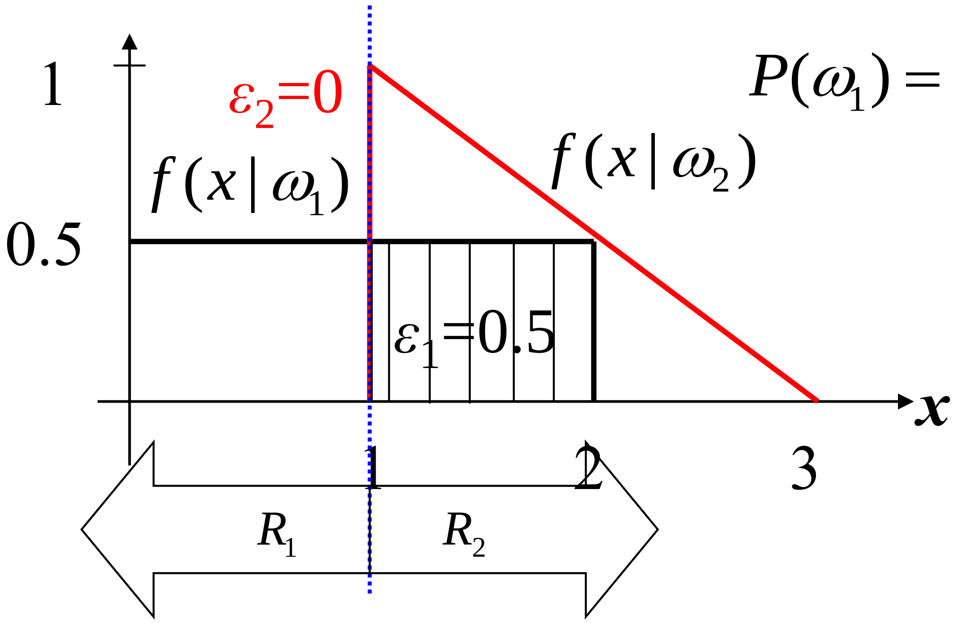
$$\varepsilon_1 = \int_{R_2} f(x|\omega_1)dx = \int_{1.5}^{3} f(x|\omega_1)dx = \int_{1.5}^{2} (1/2)dx = 1/4$$

$$\varepsilon_2 = \int_{R_1} f(x|\omega_2)dx = \int_{0}^{1.5} f(x|\omega_2)dx = \int_{1}^{1.5} \left(1 - \frac{1}{2}(x-1)\right)dx = \frac{3}{4} - \frac{5}{16} = \frac{7}{16}$$

$$P_e = P(\omega_1)\varepsilon_1 + P(\omega_2)\varepsilon_2 = (0.5)\frac{1}{4} + (0.5)\frac{7}{16} = 0.343$$

Is this optimal ?

Example : 2-Class Classifier Error



$$P(\omega_1)=P(\omega_2)=0.5$$

Minimum Error Classifier will give a smaller error!!

$$P_e=P(\omega_1)\varepsilon_1+P(\omega_2)\varepsilon_2=(0.5)(0.5)+0=0.25$$

Likelihood Ratio

$$\boxed{\left\{P(\omega_1) f(x|\omega_1) - P(\omega_2) f(x|\omega_2)\right\} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \mathbf{0}}$$

$$l(x) = \frac{f(x|\omega_1)}{f(x|\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{P(\omega_2)}{P(\omega_1)} = T$$

Likelihood ratio

Ratio of a priori probabilities

Log

monotonically increasing both sides
without affecting decision rule

Log Likelihood ratio

$$\ln\left(l(x)\right) = \ln\left(\frac{f(x|\omega_1)}{f(x|\omega_2)}\right) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \ln\left(\frac{P(\omega_2)}{P(\omega_1)}\right) = \ln\left(T\right)$$