Single-Layer Perceptron

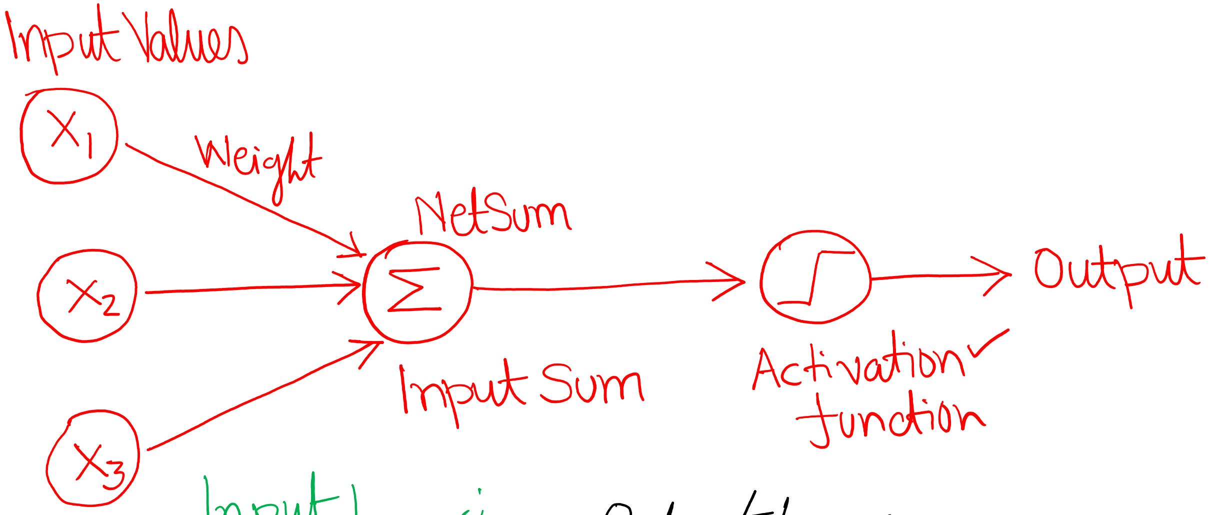
a simple Neural Network

Input Layer

Output Layer

feature vector

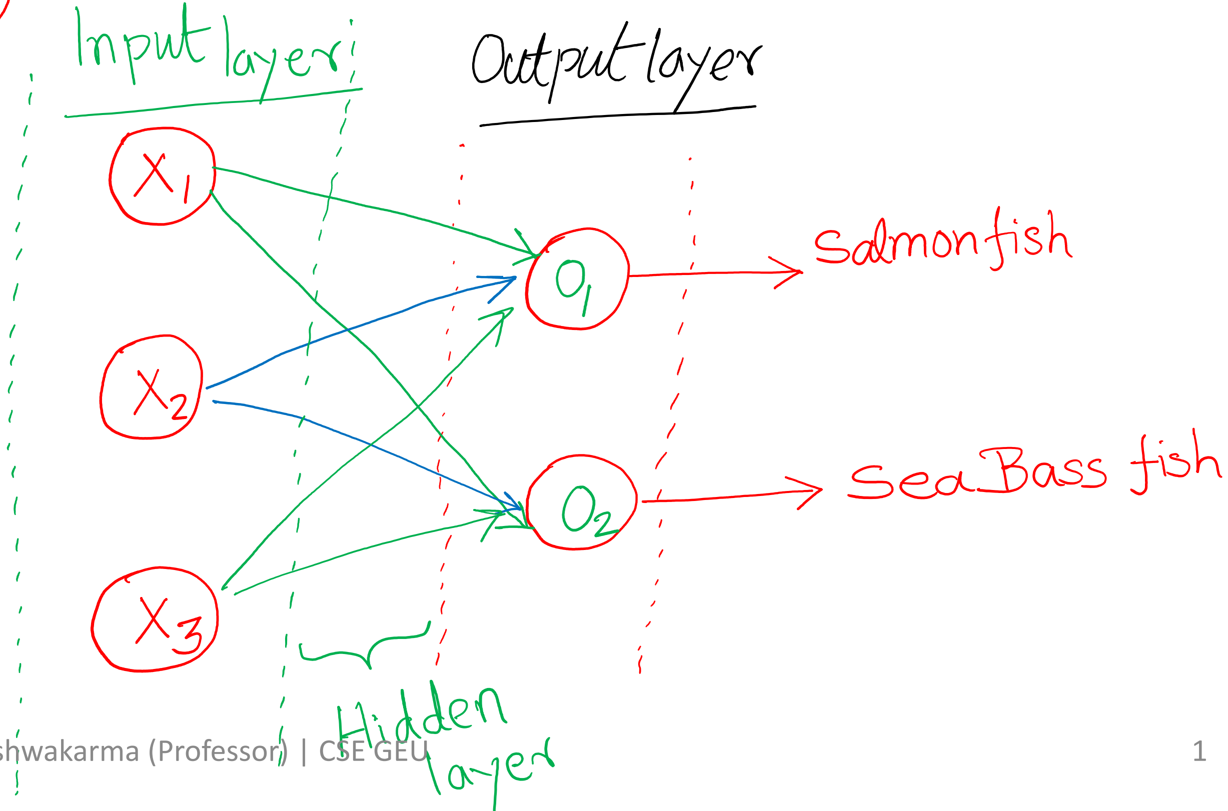
$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \Bigg\} \text{ features } \underline{\underline{3}}$$



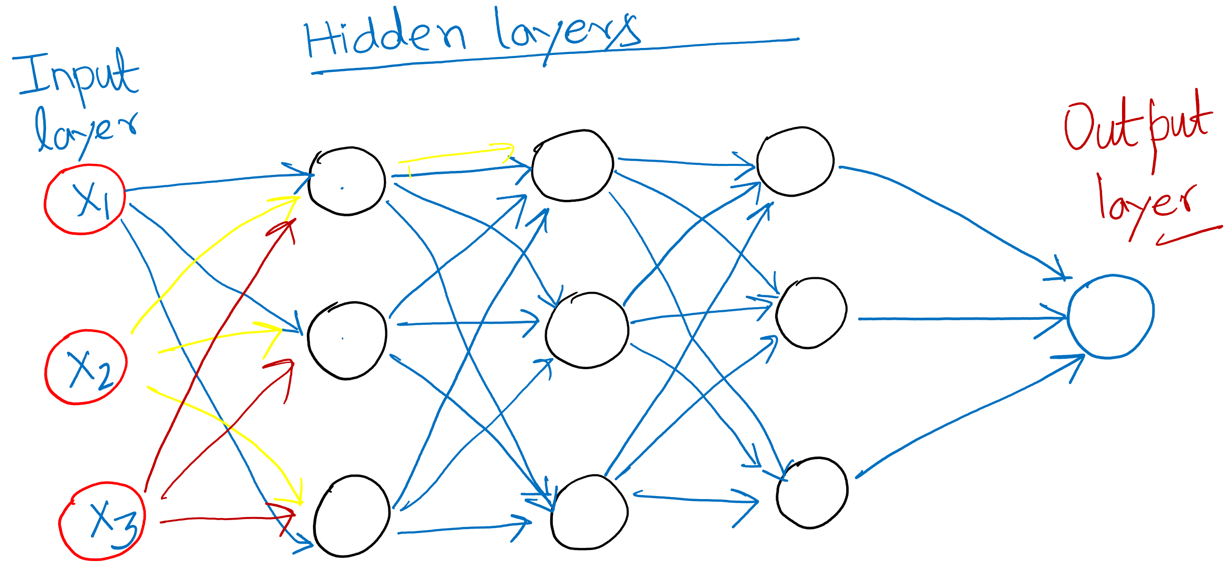
fishing plant

fish → Salmon, Seabass

$$X = \begin{bmatrix} x_1 - \text{length} \\ x_2 - \text{Width} \\ x_3 - \text{lightness} \end{bmatrix}$$



Multi Layer Perceptron



Standardization and Normalization

Data Set $\boxed{20, 10, 15, 12}$

$x_{max} = 20$

$x_{min} = 10$

Standardization = ?

$$y_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

$x_1 = 20$

$x_2 = 10$

$x_3 = 15$

$x_4 = 12$

$$\bar{x} = \frac{x_1 + x_2 + x_3 + x_4}{4}$$

$$= \frac{20 + 10 + 15 + 12}{4}$$

$$= \frac{57}{4}$$

$$\bar{x} = 14.25$$

$$y_1 = \frac{20 - 14.25}{3.75} = 1.53$$

$$y_2 = \frac{10 - 14.25}{3.75} = -1.13$$

$$y_3 = \frac{15 - 14.25}{3.75} = 0.2$$

$$y_4 = \frac{12 - 14.25}{3.75} = -0.6$$

20	$x_1 \longrightarrow y_1$	1.53	
10	$x_2 \longrightarrow y_2$	-1.13	
15	$x_3 \longrightarrow y_3$	0.2	
12	$x_4 \longrightarrow y_4$	-0.6	

$\bar{y} = 0$ $\quad Var_y = 1$

$$\frac{x_i - \bar{x}}{std._x} = y_i$$

$$\bar{Y} = \frac{1.53 + (-1.13) + 0.2 + (-0.6)}{4} = 0$$

$std_y = ?\ 1.001 \approx 1$

Variance $= (std_y)^2$

$= 1$

x_i	$x_i - \bar{x}$	$(x_i - \bar{x})^2$
20	$20 - 14.25 = 5.75$	33.06
10	$10 - 14.25 = -4.25$	18.06
15	$15 - 14.25 = 0.75$	0.56
12	$12 - 14.25 = -2.25$	5.0625

$$std_x = \sqrt{\frac{33.06 + 18.06 + 0.56 + 5.0625}{4}} = \sqrt{\frac{56.74}{4}}$$

$$= \sqrt{14.185} = 3.76$$

Standardization and Normalization ✓

Data Set $\boxed{20, 10, 15, 12}$

$x_{max} = 20$

$x_{min} = 10$

Standardization = ?

$$y_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

$x_1 = 20$
$x_2 = 10$
$x_3 = 15$
$x_4 = 12$

$$\bar{x} = \frac{x_1 + x_2 + x_3 + x_4}{4}$$

$$= \frac{20 + 10 + 15 + 12}{4}$$

$$= \frac{57}{4}$$

$$\bar{x} = 14.25$$

$$y_1 = \frac{20 - 14.25}{3.75} = 1.53$$

$$y_2 = \frac{10 - 14.25}{3.75} = -1.13$$

$$y_3 = \frac{15 - 14.25}{3.75} = 0.2$$

$$y_4 = \frac{12 - 14.25}{3.75} = -0.6$$

20	$x_1 \longrightarrow y_1$	1.53	
10	$x_2 \longrightarrow y_2$	-1.13	
15	$x_3 \longrightarrow y_3$	0.2	
12	$x_4 \longrightarrow y_4$	-0.6	

$\bar{y} = 0 \quad Var_y = 1$

$$\frac{x_i - \bar{x}}{std_x} = y_i$$

$$\bar{Y} = \frac{1.53 + (-1.13) + 0.2 + (-0.6)}{4} = 0$$

$std_y = ?\ 1.001 \approx 1$

$$Variance = (std_y)^2 = 1$$

x_i	$x_i - \bar{x}$	$(x_i - \bar{x})^2$
20	$20 - 14.25 = 5.75$	33.06
10	$10 - 14.25 = -4.25$	18.06
15	$15 - 14.25 = 0.75$	0.56
12	$12 - 14.25 = -2.25$	5.0625

$$std_x = \sqrt{\frac{33.06 + 18.06 + 0.56 + 5.0625}{4}} = \sqrt{\frac{56.74}{4}}$$

$$= \sqrt{14.185} = 3.76$$

Normalization

x_i	x_{max}	x_{min}	$x_{max}-x_{min}$	$x_i - x_{min}$	$y_i = \frac{x_i - x_{min}}{x_{max}-x_{min}}$	$2y_i - 1$
20	20	10	10	$20-10=10$	$10/10=1$	1
10				$10-10=0$	$0\div10=0$	-1
15				$15-10=5$	$5\div10=0.5$	0
12				$12-10=2$	$2\div10=0.2$	-0.6

x_i $\boxed{0 \le y_i \le 1}$

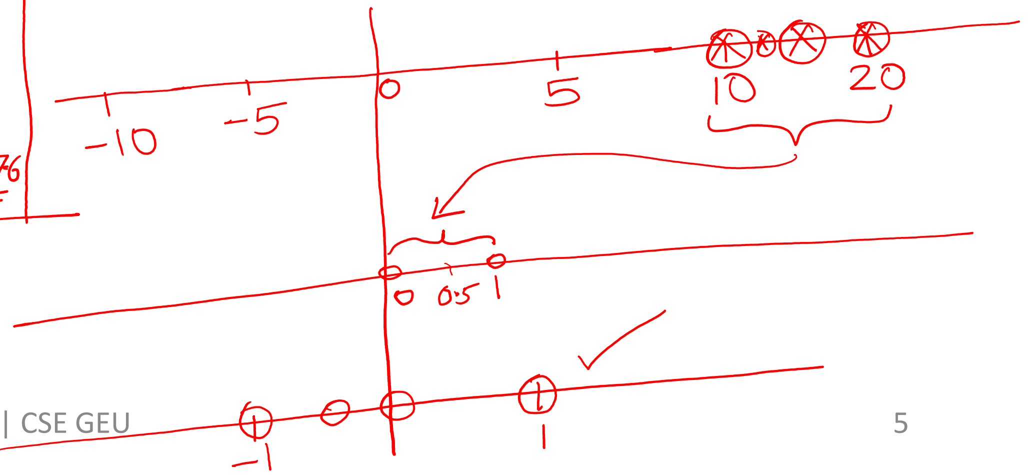
$\boxed{-1 \le y_i \le 1}$ ✓

$$Y_{mean} = \bar{Y} = \frac{1+0+0.5+0.2}{4} = \frac{1.7}{4} = 0.425$$

Y_i	$\bar{Y}$	$Y_i - \bar{Y}$	$Y_i - \bar{Y}$	$(Y_i-\bar{Y})^2$	$Std = \sqrt{\Sigma(y_i-\bar{y})^2 \div 4}$
1	0.425	$1-0.425$	0.575	0.3306	
0		$0-0.425$	-0.425	0.1806	$Std = \sqrt{\frac{0.5675}{4}}$
0.5		$.5-0.425$	0.075	0.0056	$Std_y = \sqrt{0.1418} = 0.376$
0.2		$.2-0.425$	-0.225	0.0506	

$$Var_y = (Std_y)^2$$

$$\boxed{V_{ary} = 0.1418}$$



Linear Discriminant Analysis (LDA) Supervised

→ Supervised tech.

favour of Wine ← 1-type, 2-type, 3-type } Customers

→ ingradients.

Attributes / features → Model → Type-1, Type-2, Type-3

* Dimensionality Reduction
total features = 13

* pre-processing step for pattern classification

* Lower the dimensionality space –

(n) 13-dimensional samples
total records = 178

→ few dimensions (K) ≤ (n-1)

$\boxed{K \leq n-1}$

n = 13
K = 12

K = may less than 12

Input variables | O/p

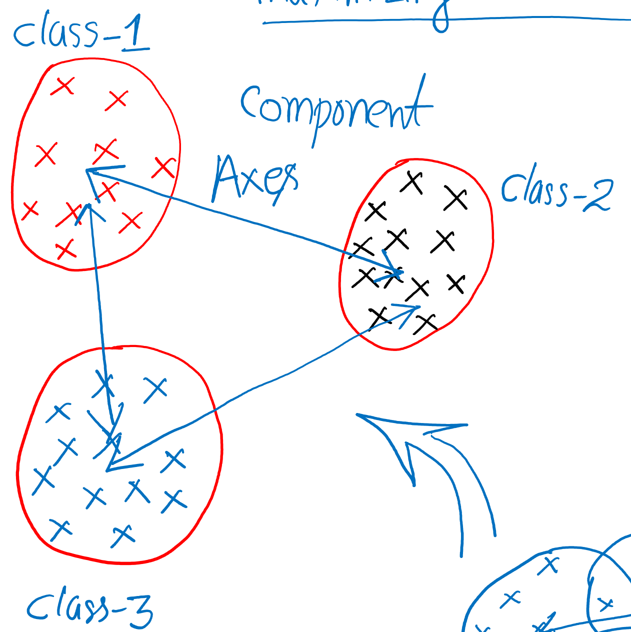
x_1	x_2	x_3	x_4	Y

dependent Variable

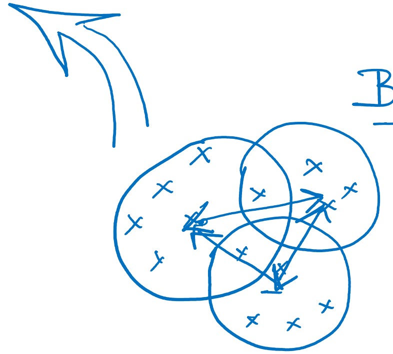
LDA Vs PCA
Supervised   Unsupervised

LDA
maximizing the component axes for class separation

Good Projection



Bad Projection



(1) compute d-dimensional mean vectors for the different classes from the dataset

(2) Computation of scatter matrices

(3) computation eigen-vectors $(e_1, e_2, e_3, \dots e_d)$

eigen values $(\lambda_1, \lambda_2, \lambda_3 \dots \lambda_d)$

(4) sort the eigen vectors

$$\left.\begin{matrix} e_3 \\ e_{10} \\ e_1 \\ e_9 \end{matrix}\right\} \text{K eigen vectors}$$

e_{d-3}

e_{d-1}

$\vdots$

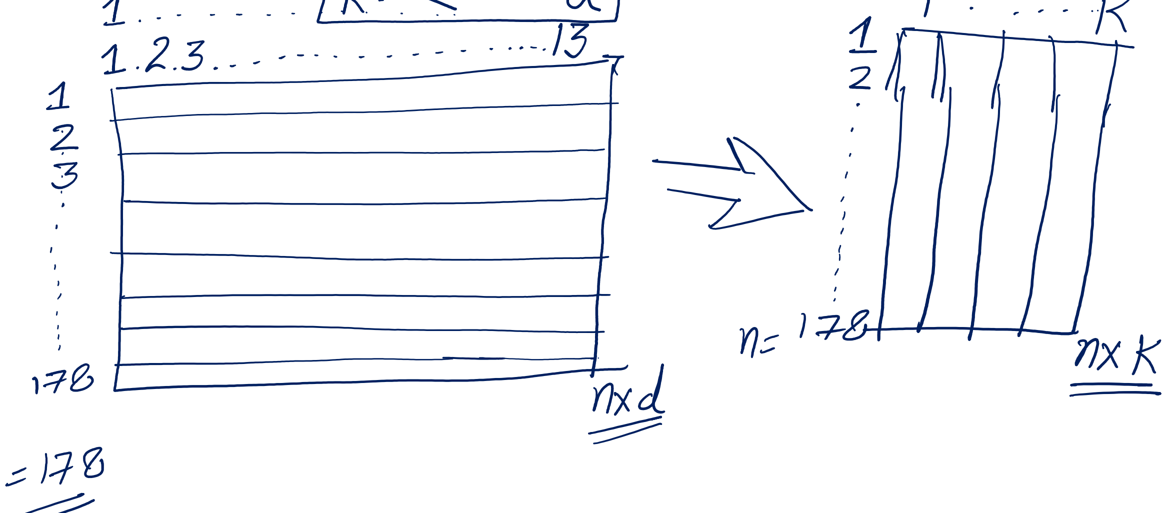
$\boxed{d \times K \text{ dimensional matrix } W}$

$\boxed{K < d}$

New Sample space

(5) $Y_{n \times K} = X_{n \times d} * W_{d \times K}$

$X \quad n \times d$



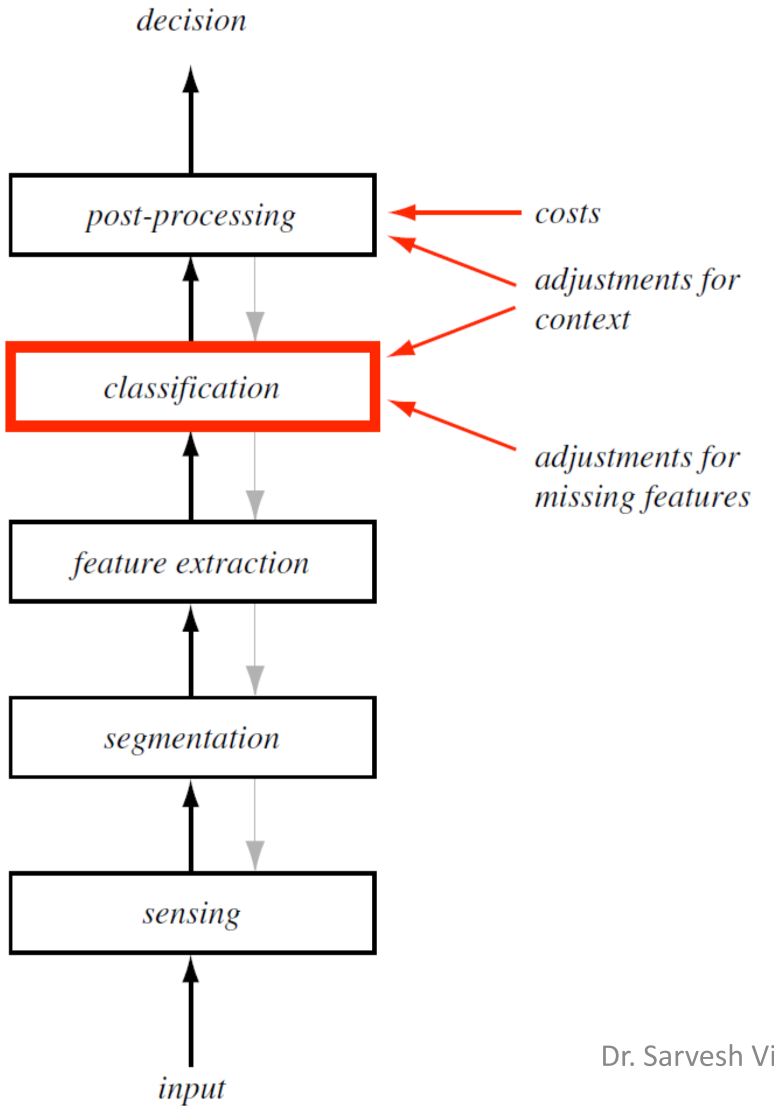
total features = n

selected features = $\boxed{m < n}$

discarded features = $\boxed{n - m = \bar{m}}$

$Z_1 = \{n-1\}$ $Z_2 = \{n-2\}$ $Z_3 = \{n-3\}$ $Z_4 = \{n-4\}$

Classifying an Object



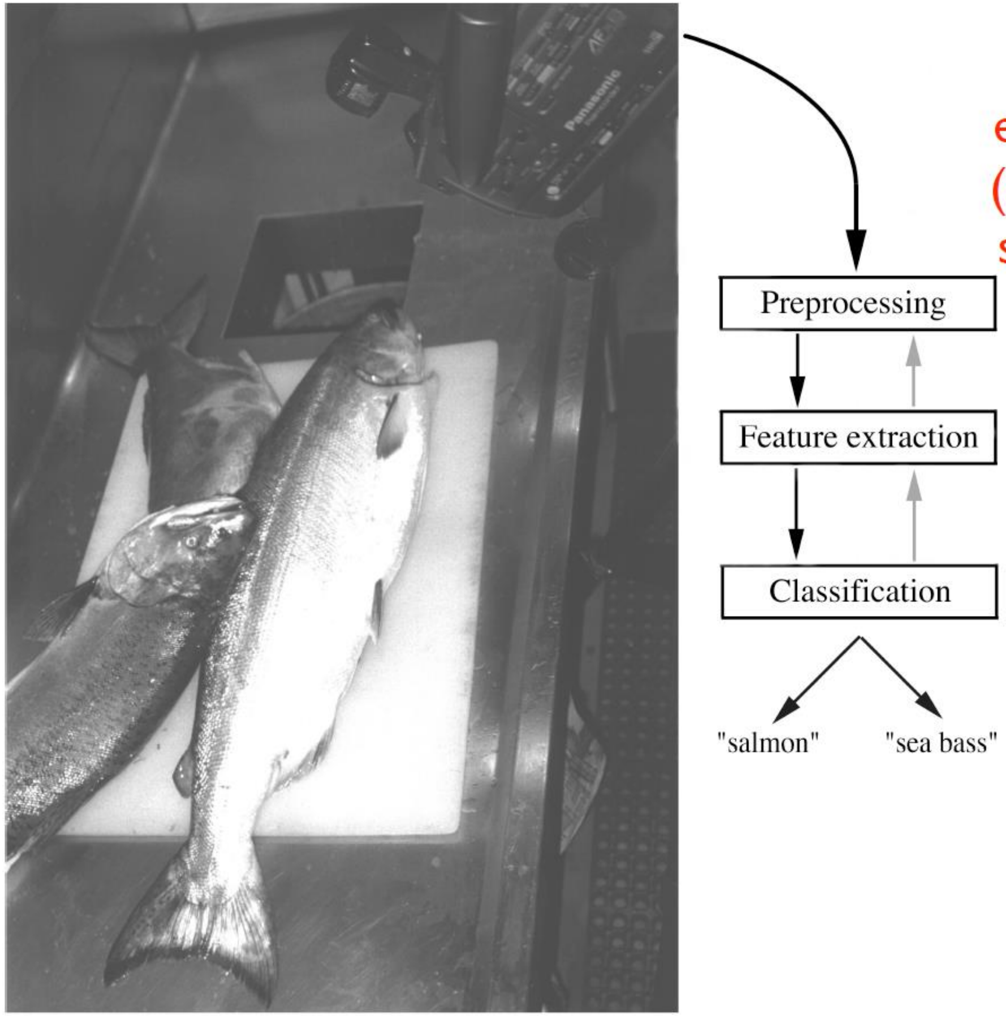
Obtaining Model Inputs

Physical signals converted to digital signal (transducer(s)); a region of interest is identified, features computed for this region

Making a Decision

Classifier returns a class; may be revised in post-processing (e.g. modify recognized character based on surrounding characters)

Example (DHS): Classifying Salmon and Sea Bass



e.g. image processing
(adjusting brightness)
segment fish regions

FIGURE 1.1. The objects to be classified are first sensed by a transducer (camera), whose signals are preprocessed. Next the features are extracted and finally the clas-

Designing a classifier or clustering algorithm

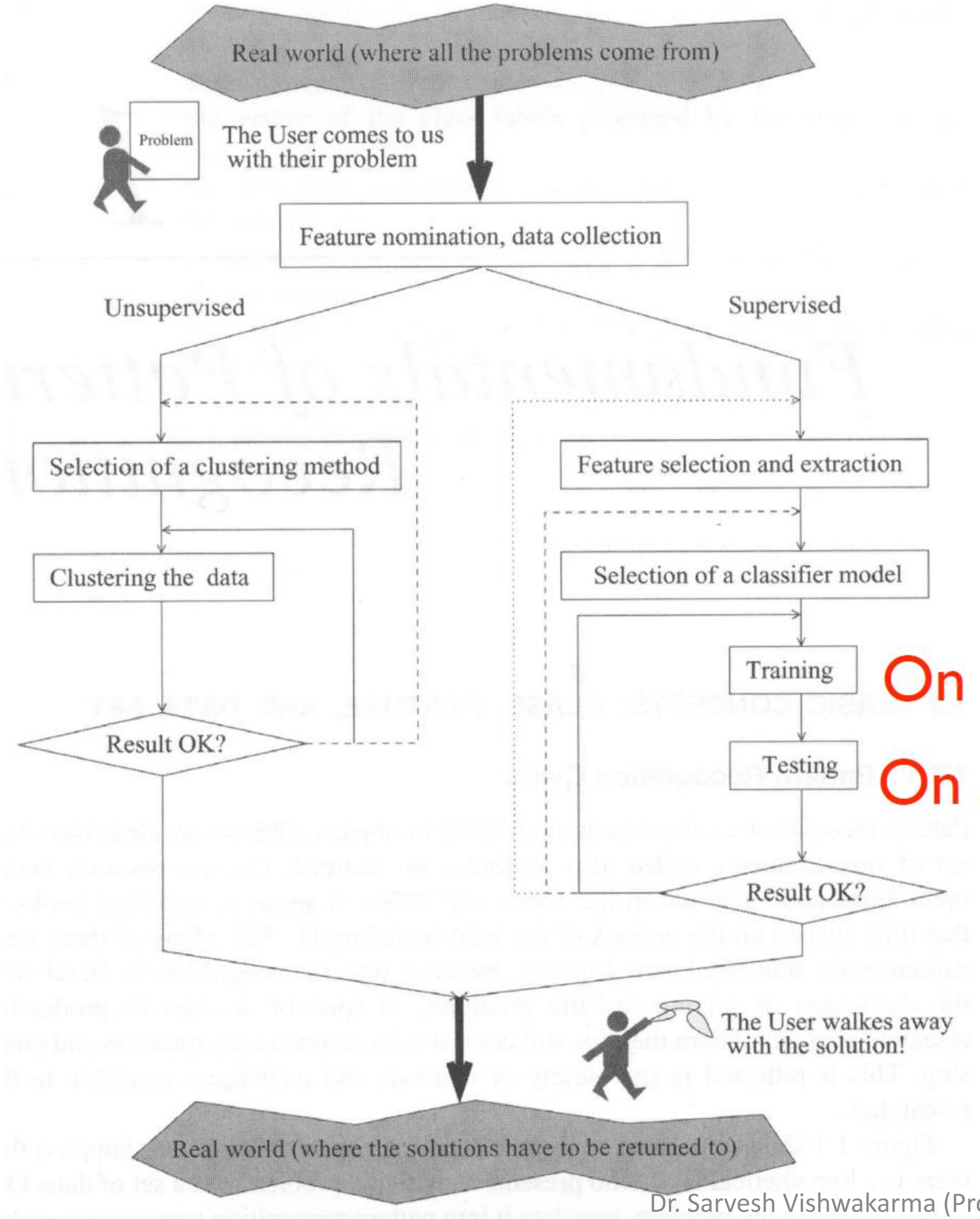


Fig. 1.1 *The pattern recognition cycle.*

On a training set (learn parameters)

On a *separate* testing set

Feature Selection and Extraction

Feature Selection

Choosing from available features those to be used in our classification model. Ideally, these:

- Discriminate well between classes
- Are simple and efficient to compute

Feature Extraction

Computing features for inputs at run-time

Preprocessing

User to reduce data complexity and/or variation, and applied before feature extraction to permit/simplify feature computations; sometimes involves other PR algorithms (e.g. segmentation)

Types of Features

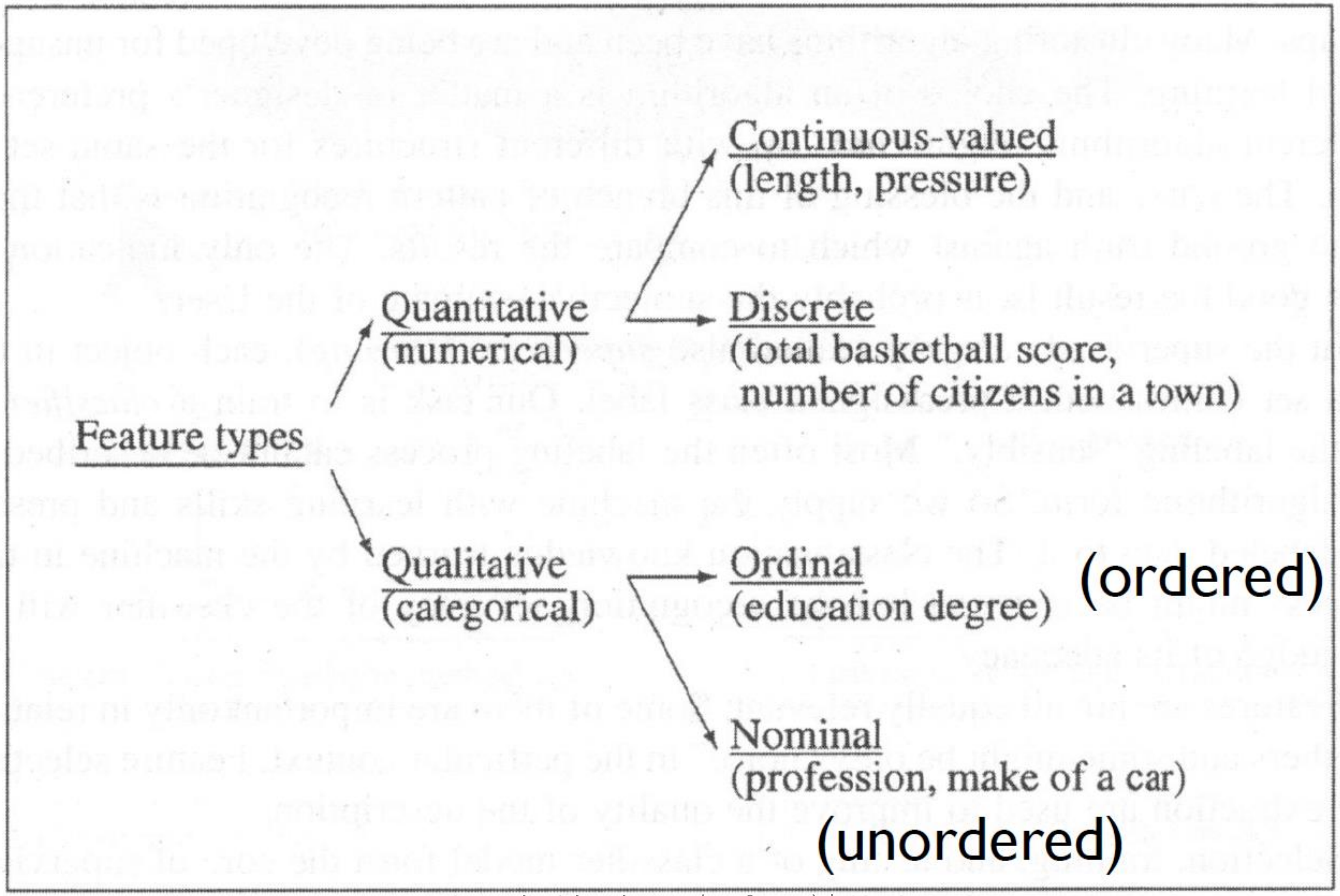
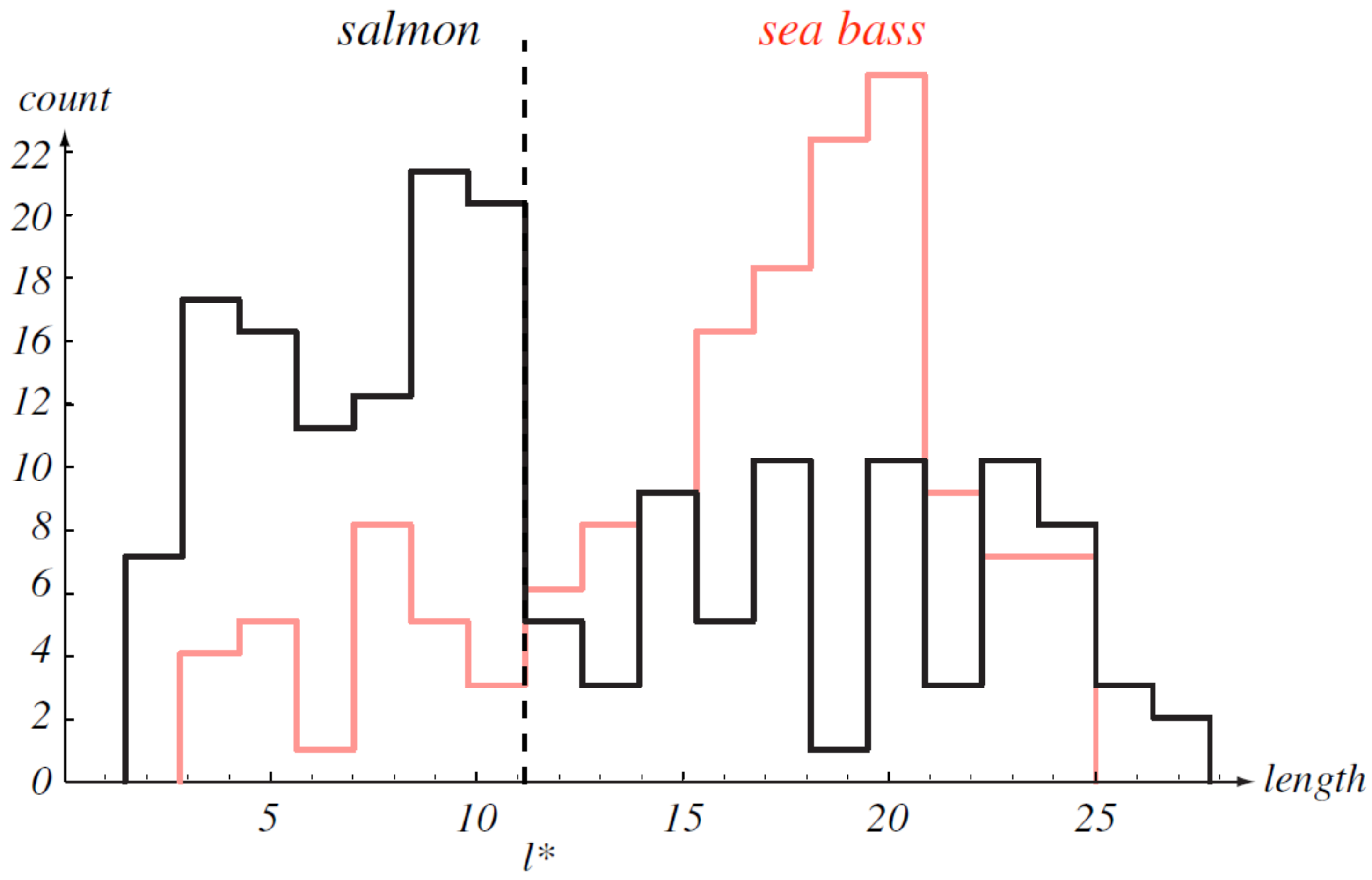


Fig. 1.2 Types of features.

Example Single Feature (DHS): Fish Length



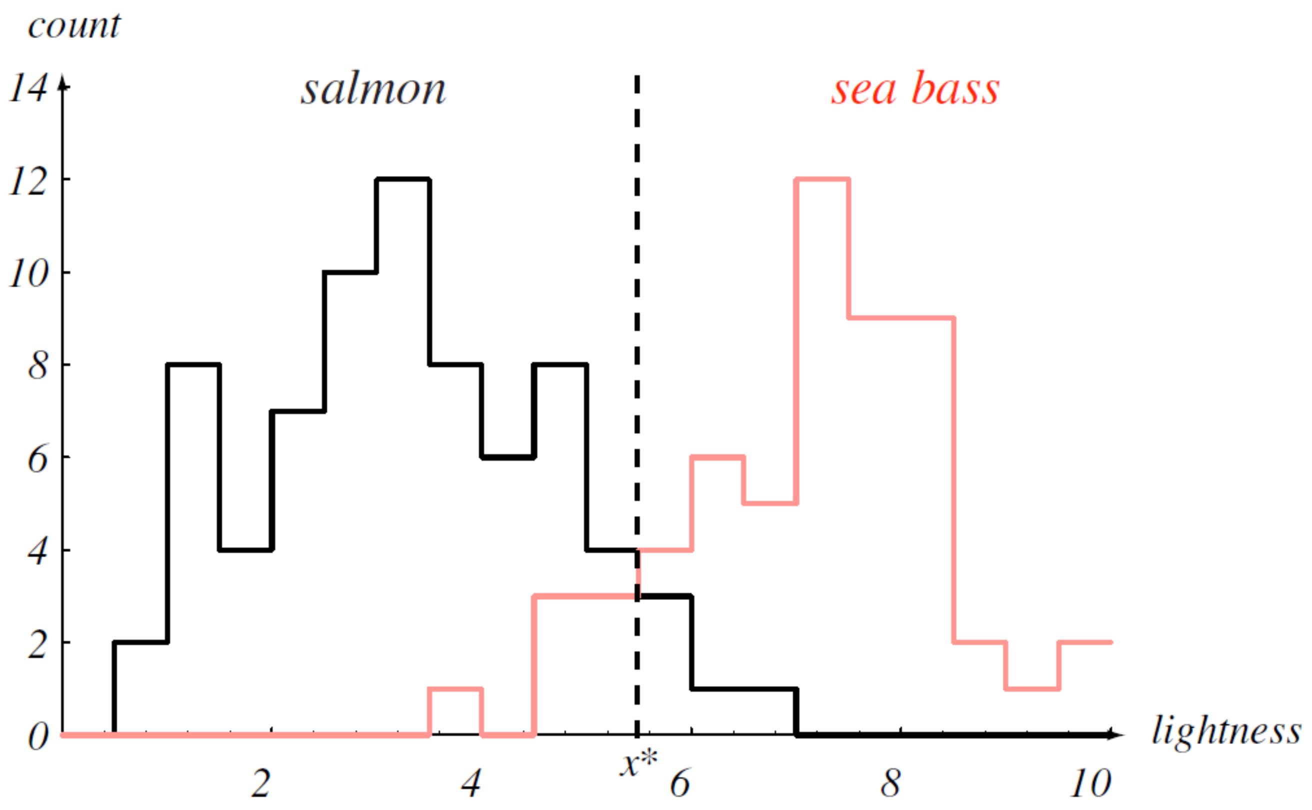
A Poor Feature for Classification

Computed on a training set

No threshold will prevent errors

Threshold *l** shown will produce fewest errors on average

A Better Feature: Average Lightness of Fish Scales



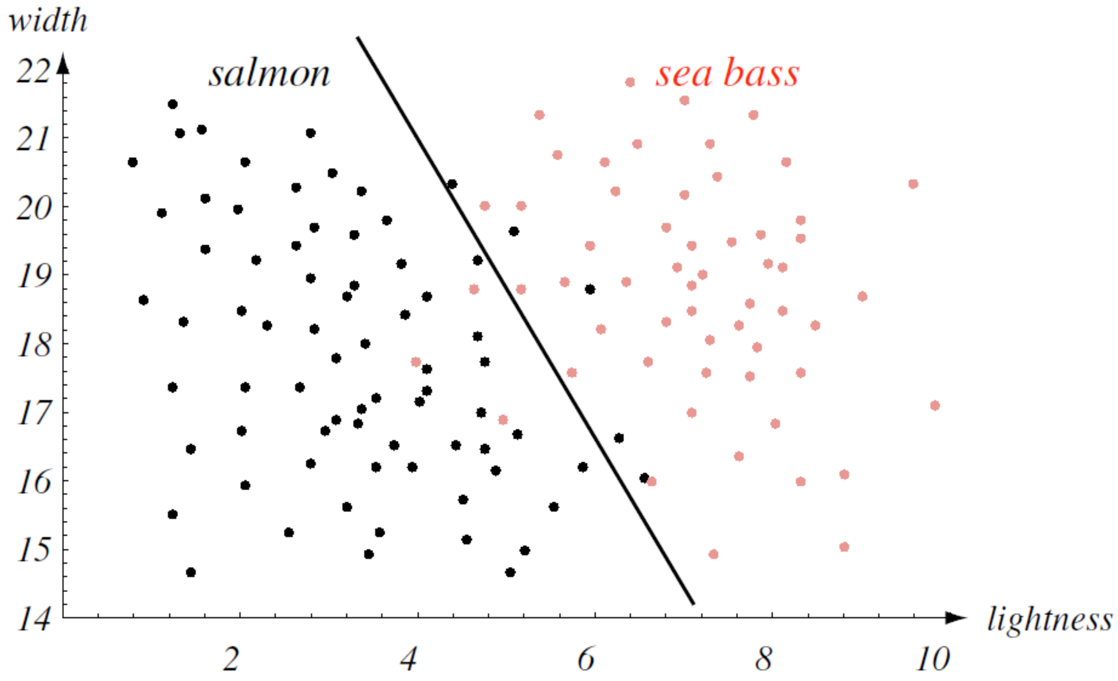
Still some errors

even for the best threshold, x^* (again, min. average # errors)

Unequal Error Costs

If worse to confuse bass for salmon than vice versa, we can move x^* to the left

A Combination of Features: Lightness and Width



Feature Space

Is now two-dimensional; fish described in model input by a *feature vector* (x1, x2) representing a point in this space

Decision Boundary

A linear discriminant (line used to separate classes) is shown; still some errors

In general, determining appropriate features is a difficult problem, and determining optimal features is often impractical or impossible.

Classifier: A Formal Definition

Classifier (continuous, real-valued features)

Defined by a function from a n-dimensional space of real numbers to a set of *c* classes, i.e.

$$D : R^n \rightarrow \Omega, \text{ where } \Omega = \{\omega_1, \omega_2, \ldots \omega_c\}$$

Canonical Model

Classifier defined by *c* discriminant functions, one per class. Each returns a real-valued "score." Classifier returns the class with the highest score.

$$g_i : R^n \rightarrow R, \quad i = 1, \ldots c$$

$$D(x) = \omega_{i*} \in \Omega \longleftrightarrow g_{i*} = \max_{i=1,\ldots,c} g_i(x)$$

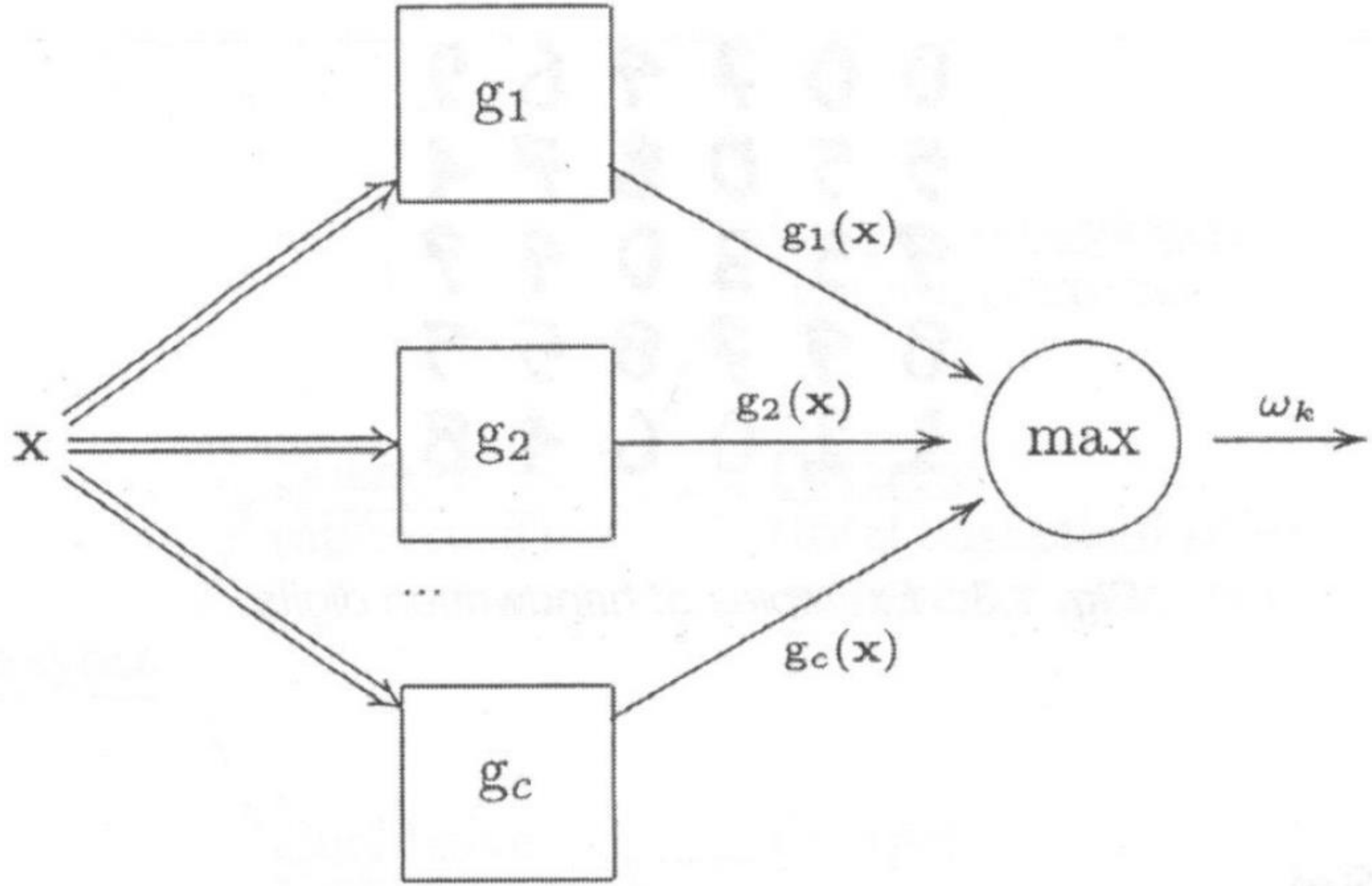


Fig. 1.4 *Canonical model of a classifier. The double arrows denote the n-dimensional input vector **x**, the output of the boxes are the discriminant function values, $g_i(x)$ (scalars), and the output of the maximum selector is the class label $\omega_k \in \Omega$ assigned according to the maximum membership rule.*

Regions and Boundaries

Classification (or Decision) Regions

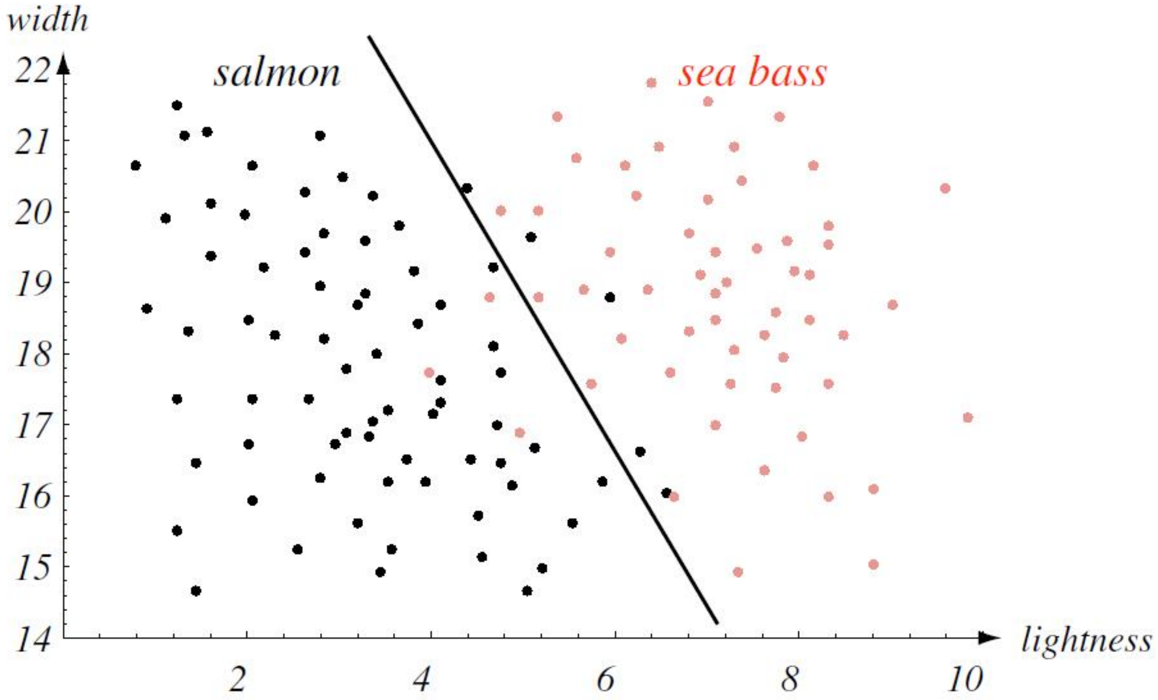
Regions in feature space where one class has the highest discriminant function "score"

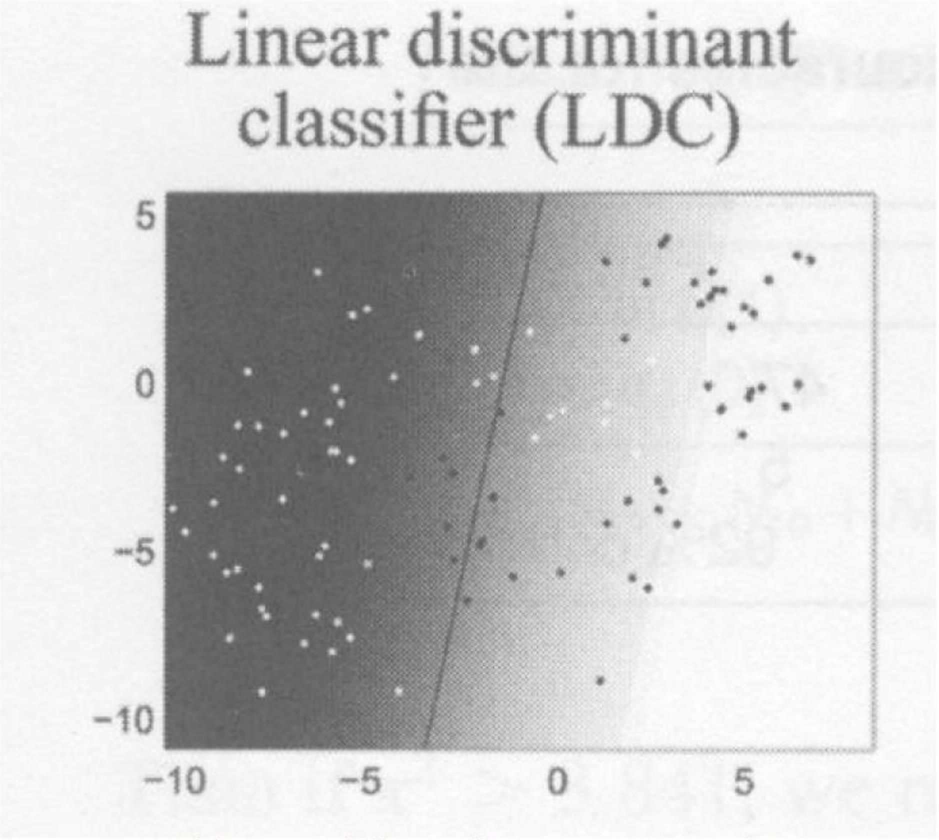
$$R_i = \left\{ x \middle| x \in R^n, \quad g_i(x) = \max_{k=1,\ldots,c} g_k(x) \right\}, \quad i = 1, \ldots, c$$

Classification (or Decision) Boundaries

Exist where there is a tie for the highest discriminant function value

Example: Linear Discriminant Separating Two Classes





(from Kuncheva: visualizes changes (gradient) for class score)