♠ Distribution-Free (No Statistics) Classification

Objective: Determine a function in feature space that specifies the membership in each class, partitioning the feature space into mutually exclusive regions by non-statistical criteria.

➔ Discriminant Functions (or Characteristic Functions) => $g_k(\underline{x})$

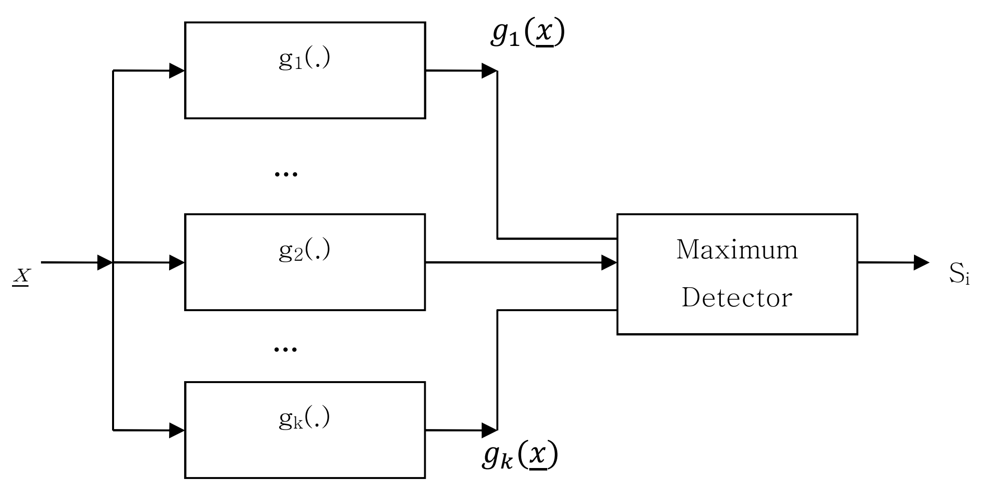
Fundamentals

$g_k(\underline{x})$ for k=1,2,…,K

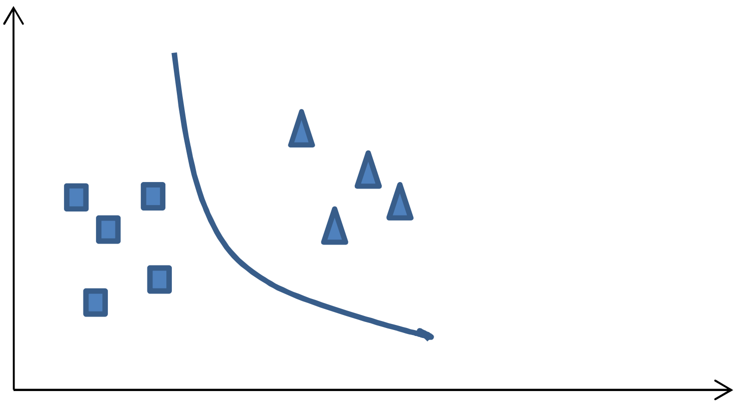
A point $\underline{x}$ is a member of class S_i

iff $g_i(\underline{x}) > g_j(\underline{x}) \; \forall j \neq i$ ← Classification rule

Classification System



Feature Space N=2



Separating surfaces or decision boundaries defined by

$g_k(\underline{x}) - g_j(\underline{x}) = 0$ separates S_k from S_j.

There are K classes => $\frac{K(K-1)}{2}$ possible decision boundaries, but some may be redundant (e.g. S_3-S_4 above)

Note: g's are not unique

Can add constant to all g_k's:

$g_k(\underline{x})$ <-> $g_j(\underline{x})$

$g_k(\underline{x}) + C$ <-> $g_j(\underline{x}) + C$

Can define new g':

$g'_k(\underline{x})=f(g_k(\underline{x}))$

$g'_j(\underline{x})=f(g_j(\underline{x}))$

where f is a monotonic and increasing function

Let's try Distance Functions

Measure of the distance from an unknown $\underline{x}$ to a class S_k:

$$D\left[\underline{x}, \left\{\underline{y}_m^{(k)}\right\}\right] = \frac{1}{M_k}\sum_{1}^{M_k} d^2(\underline{x}, \underline{y}_m^{(k)})$$

This is Mean Square Distance

d might be Euclidean distance

Properties of Distance Functions

A distance function must satisfy:

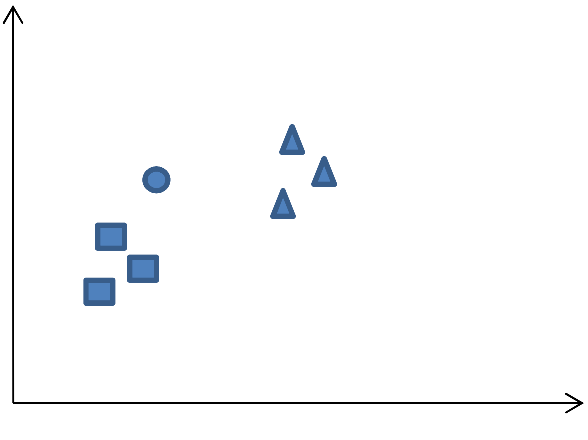
1. $d(\underline{x}, \underline{y}) = d(\underline{y}, \underline{x})$
2. $d(\underline{x}, \underline{y}) \le d(\underline{x}, \underline{z}) + d(\underline{z}, \underline{y})$
3. $d(\underline{x}, \underline{y}) = 0, iff\ \underline{x} = \underline{y}$
4. $d(\underline{x}, \underline{y}) \ge 0\ \forall \underline{x}, \underline{y}$

Example: N=2 classes, Minimum-distance-to-class classifier

$d(\underline{x},S_1)<d(\underline{x},S_2)$ => $\underline{x} \in S_1$

Let $g_k(\underline{x}) = -d(\underline{x},S_k)$

$g_1(\underline{x})> g_2(\underline{x})$ => $\underline{x} \in S_1$

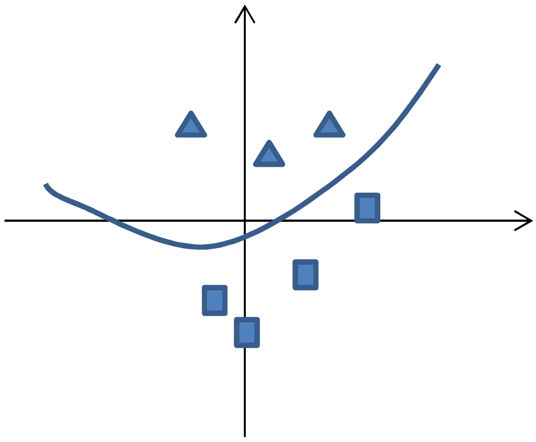


Decision rule for a 2-class problem

Let g(k)=$g_1(\underline{x})$-$g_2(\underline{x})$

If g($\underline{x}$)>0, $\underline{x} \in S_1$

If g($\underline{x}$)<0, $\underline{x} \in S_2$

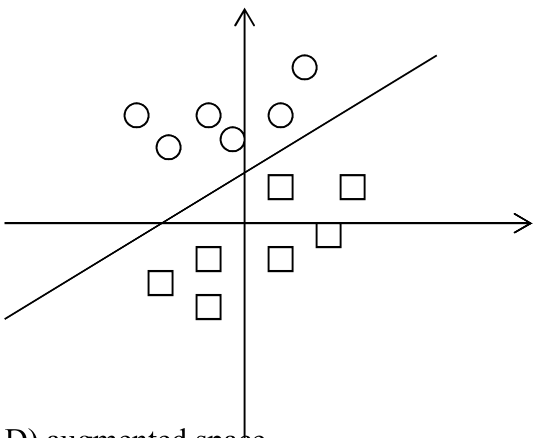


Linear Discriminant Functions

$$g_k(\underline{x}) = w_1^{(k)} x_1 + w_2^{(k)} x_2 + \ldots + w_N^{(k)} x_N + w_{N+1}^{(k)}$$
$$g_k(\underline{x}) = \underline{w}^T \underline{x} + w_{N+1}^{(k)} = \underline{w}^{(a)} \underline{x}^{(a)}$$

$$\underline{x}^{(a)} = \begin{bmatrix} x_1 \\ \cdots \\ x_N \\ 1 \end{bmatrix}$$ =input vector in augmented, (a), space

- Superscript (a) is often suppressed



x_3=1 plane in (3-D) augmented space.

Linear decision surface:

2-D, Line

3-D, plane

N-D, hyperplane (N-1)-D

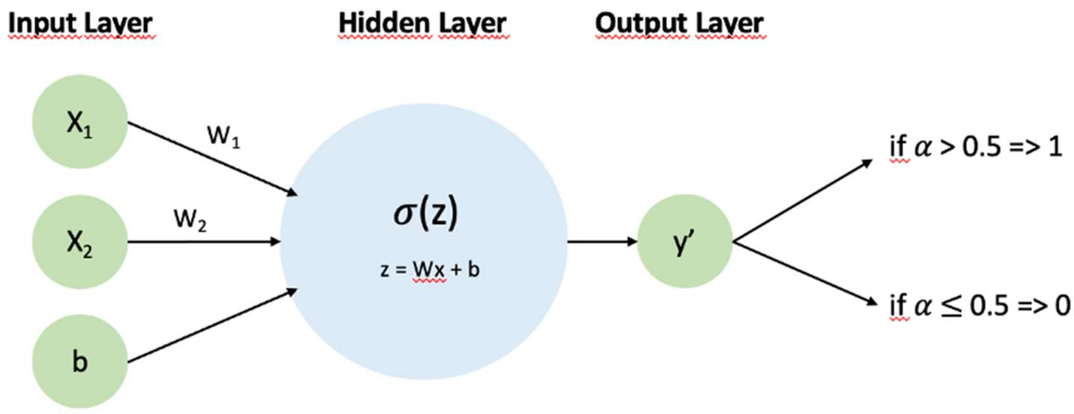
Example: Two-Class Problem

For a 2-class problem:

$$g(\underline{x}) = g_1(\underline{x}) - g_2(\underline{x})$$
$$= (\underline{w}_1^T\underline{x} + w_{N+1}^{(1)}) - (\underline{w}_2^T\underline{x} + w_{N+1}^{(2)})$$
$$= (\underline{w}_1^T - \underline{w}_2^T)\underline{x} + (w_{N+1}^{(1)} - w_{N+1}^{(2)})$$
$$g(\underline{x}) = \underline{w}^T\underline{x} + w_{N+1}$$

A Single Neuron Model

- Calculate g for 2-class problem:



- Compare to the 2-class linear classifier

Ex.) Find a linear classifier for the following data

Table 2.1 Classification Data

x_1	x_2	t
0.2	0.3	0
0.2	0.8	0
0.8	0.2	0
1.0	0.8	1

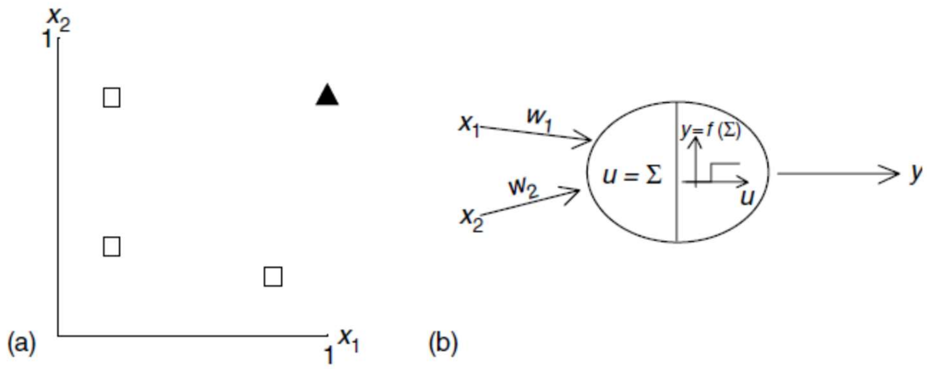


Figure 2.6 Classification by a threshold neuron: (a) classification data and (b) configuration of a linear threshold neuron for this task.

Inputs and weights as vectors. For simplicity, vector notation will be used. In this notation, an input vector is represented by upper case **x** as

$$\mathbf{x} = \{x_1, x_2\}.$$

Thus, the four input vectors can be represented as

$$\mathbf{x}_1 = \{0.2, 0.3\},\ \mathbf{x}_2 = \{0.2, 0.8\},\ \mathbf{x}_3 = \{0.8, 0.2\},\ \mathbf{x}_4 = \{1.0, 0.8\}.$$

Similarly, the weight vector can be denoted in vector form as

$$\mathbf{w} = \{w_1, w_2\}.$$

The four input vectors and the weight vector are graphically presented in Figure 2.8.

The weighted sum of the inputs, $u = x_1.w_1 + x_2.w_2$, can be represented as multiplication of the input and weight vectors or dot product as

$$u = \mathbf{w} \cdot \mathbf{x} = \{w_1, w_2\} \cdot \{x_1, x_2\}$$
$$= w_1 x_1 + w_2 x_2$$

$$\text{For } \mathbf{w} = \{1, 1\}, \quad u = x_1 + x_2.$$

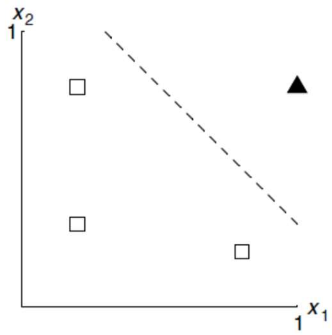


Figure 2.7 Classification boundary of the threshold neuron superimposed on the data.

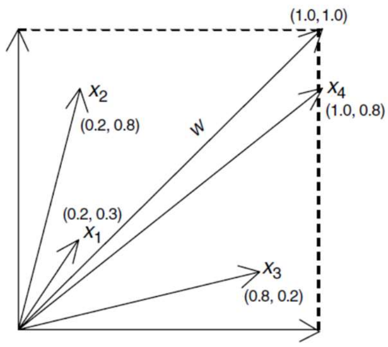


Figure 2.8 Representation of the input data and the weights as vectors.