Notation Preliminaries

Unknown data vectors: $\underline{x}$

Prototypes:

$\underline{y}_m^{(k)}$ is the m-th prototype that belongs to class S_k

k = class index

m_k = no. of prototypes in S_k

$\underline{y}_m^{(k)}$, m = 1,2,…, m_k defines all prototypes of class S_k

Pattern Space (Dimensionality R)

Unknown $\underline{x}=(x_1, x_2, x_3, \ldots, x_R)^T$

Prototype $\underline{y}_m^{(k)} = (y_{1m}^{(k)}, y_{2m}^{(k)}, y_{3m}^{(k)}, \ldots, y_{Rm}^{(k)})$

Feature Space (Dimensionality N)

Unknown $\underline{x}=(x_1, x_2, x_3, \ldots, x_N)^T$

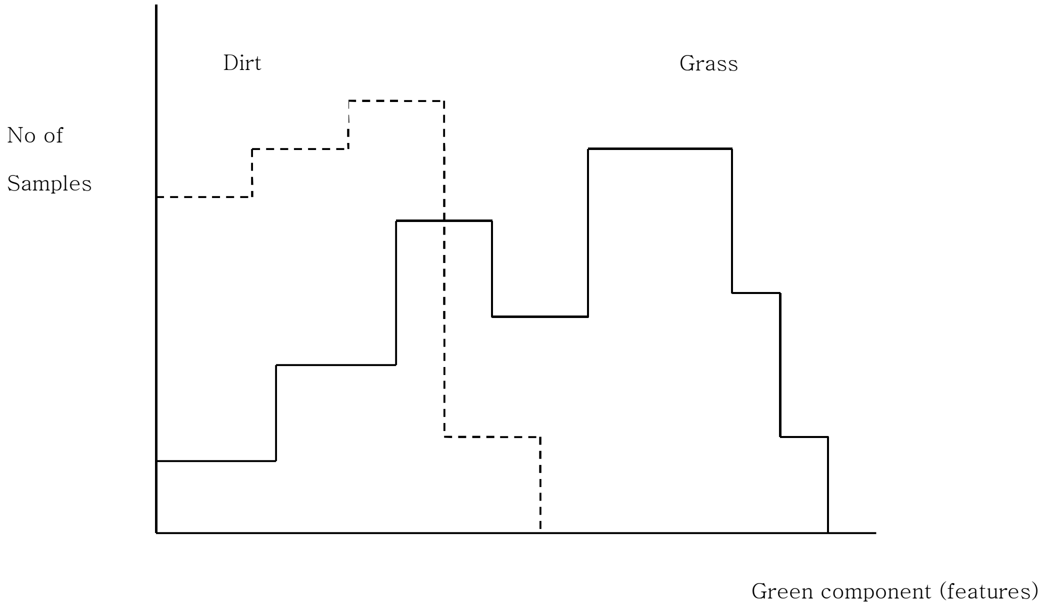
Prototype $\underline{y}_m^{(k)} = (y_{1m}^{(k)}, y_{2m}^{(k)}, y_{3m}^{(k)}, \ldots, y_{Nm}^{(k)})$

Classification Space (Dimensionality K)

S_k, k=1,2,3,…K defines all classes.

Introductory Example

- Classify dirt and grass regions from aerial pictures or images.



Decision Rule if $x<5.5$, $x \in S_1 = dirt$
If $x \geq 5.5$, $x \in S_2 = grass$

- Better than randomly assigning class
- Will have classification errors? YES

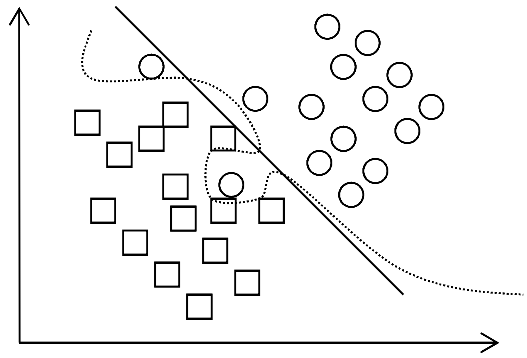
How to improve?

1. Incorporate statistics of samples
2. Add more features
   A. Possible features
      i. Blue
      ii. Red
      iii. Frequency: to get a feature, define some average frequency magnitude over the image, $\bar{g}$.
      iv. Texture regularity

For 2 features

$N = 2$

Sample: $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \text{green component} \\ \text{texture regularity} \end{bmatrix}$



2D feature space

Feature Space	Decision Surface	Linear Case
1-D	Point (0-D)	
2-D	1-D (curve)	Line
3-D	2-D (surface)	Plane
N-D	(N-1)-D surface	Hyperplane (N-1)-D

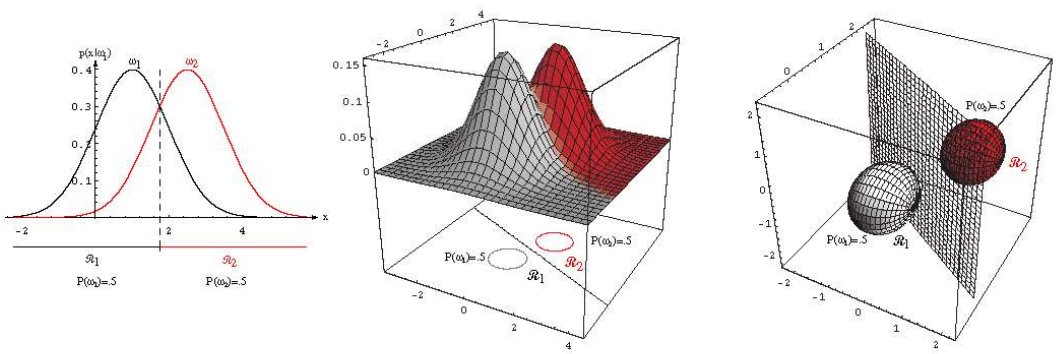


Figure 2.10: If the covariances of two distributions are equal and proportional to the identity matrix, then the distributions are spherical in d dimensions, and the boundary is a generalized hyperplane of $d-1$ dimensions, perpendicular to the line separating the means. In these 1-, 2-, and 3-dimensional examples, we indicate $p(\mathbf{x}|\omega_i)$ and the boundaries for the case $P(\omega_1) = P(\omega_2)$. In the 3-dimensional case, the grid plane separates $\mathcal{R}_1$ from $\mathcal{R}_2$.

Inverse Crime

- Test performance on new data, not on the training set (no inverse crime)

Note

1. Important to select good features

2. Important to have a good classifier

3. Performance on new patterns (unknowns)
   A. How well does the system generalize?
   B. How representative is the training set of the actual unknown data?
   C. What is the inherent dimensionality of the problem?

4. Can selection of a decision boundary be automated? Training or Learning

5. To what degree are underlying statistics of the samples (features) known? How can they be used to improve the classifier's performance? -> Statistical classification

6. Can these algorithms be implemented in parallel hardware for fast execution? Artificial neural networks

7. What if you don't know what classes the prototypes belong to? Unsupervised classification and Unsupervised learning.

Distance Functions

A distance function must satisfy:

1. $d(\underline{x}, \underline{y}) = d(\underline{y}, \underline{x})$

2. $d(\underline{x}, \underline{y}) \leq d(\underline{x}, \underline{z}) + d(\underline{z}, \underline{y})$

3. $d(\underline{x}, \underline{y}) = 0, iff\ \underline{x} = \underline{y}$

4. $d(\underline{x}, \underline{y}) \geq 0\ \forall \underline{x}, \underline{y}$

Use of Distance Functions (Ex.)

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

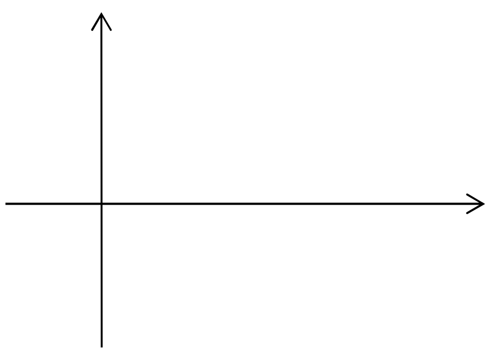
$$D\left[\underline{x}, \left\{\underline{y}_m^{(k)}\right\}\right] = \frac{1}{M_k} \sum_{1}^{M_k} d^2(\underline{x}, \underline{y}_m^{(k)})$$

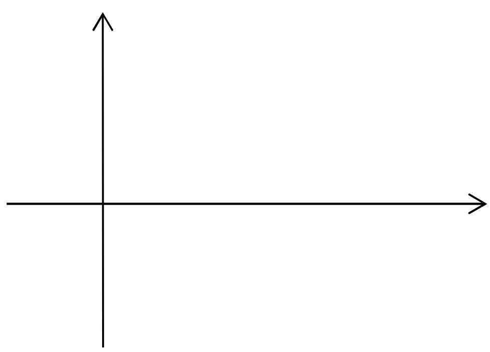
This is Mean Square Distance

d might be Euclidean distance

Data Normalization

Normalization is often required for distance functions to have the desired meaning.





Example 1

$$x_r' = \frac{x_r}{a_r}$$

where $a_r = \max_{k,m}\{y_{r_m}^{(k)}\} - \min_{k,m}\{y_{r_m}^{(k)}\}$

This is extremal value normalization (sensitive to noise)

Example 2

Less sensitive to noise
Let $a_r = \sigma_r$

$$\sigma_r^2 = \frac{\sum_{k=1}^{K}\sum_{m=1}^{M_k}[y_{r_m}^{(k)} - \overline{y}_r]^2}{\sum_{k=1}^{K} M_k}$$

where $\overline{y_r} = [\sum\sum y_{r_m}^{(k)}] / \sum_{k=1}^{K} M_k$

This is standard deviation normalization.

♠ **Distribution Free Classification**
(DHS Sec. 2.4 and Ch. 5)

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) => $g_k(\underline{x})$

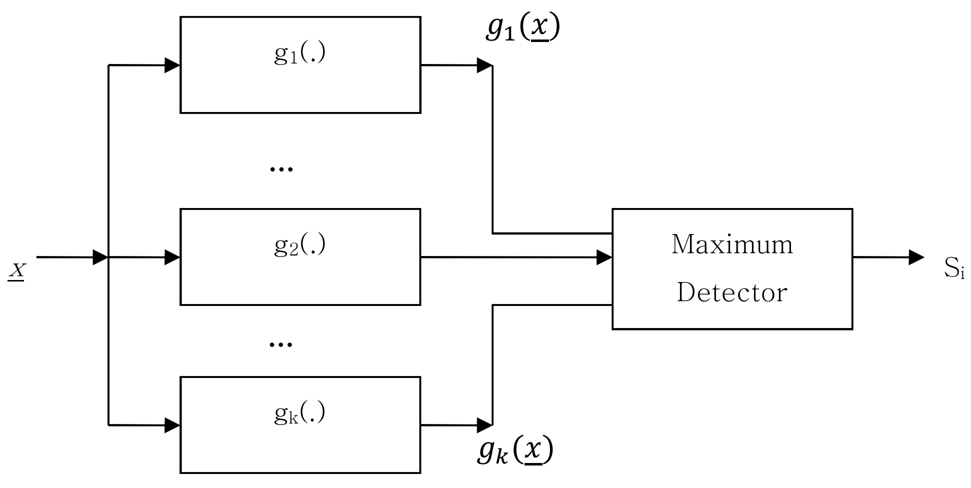
Fundamentals

$g_k(\underline{x})$ for k=1,2,…,K

A point 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

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

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

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

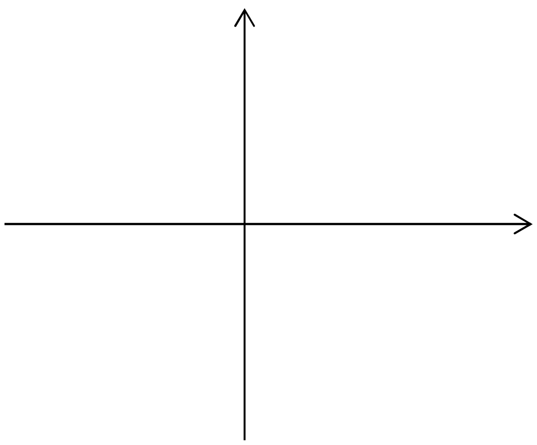
$g_1(\underline{x})> g_2(\underline{x}) \Rightarrow \underline{x} \in S_1$

<u>Decision rule for a 2-class problem</u>

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

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

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



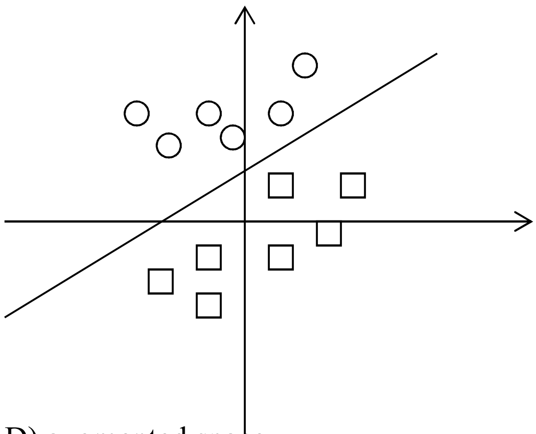
<u>Linear Discriminant Functions (Read DHS 5.2.1)</u>

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

$g_k(\underline{x}) = \underline{w}^T \underline{x} + w_{N+1}^{(k)} = \underline{w}^{(a)} \underline{x}^{(a)}$

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

- Superscript (a) is often suppressed



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

여기에 수식을 입력하십시오.

For a 2-class problem:

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

Linear decision surface:

2-D, Line

3-D, plane

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

A Single Neuron Model

Calculate g for 2-class problem: