※ Multiclass Problems: K classes (K>2 classes)

Let's try to build a K-class discriminant by combining a number of 2-class discriminant functions. But this faces some difficulties.

Try K Discriminant Functions, $g_k(x)$

Consider a single K-class discriminant function (i.e., K linear functions).
Decision rule:

$$\underline{x} \in S_k \quad \text{iff} \quad g_k(\underline{x}) > g_j(\underline{x}), \quad \forall j \neq k$$

Decision hyperplanes:

$$g_k(\underline{x}) = g_j(\underline{x})$$
$$\underline{w}_k^T \underline{x}^{(a)} = \underline{w}_j^T \underline{x}^{(a)}$$
$$(\underline{w}_k^T - \underline{w}_j^T)\underline{x}^{(a)} = 0$$

Definition

If $g_k(\underline{y}_m^{(k)}) > g_j(\underline{y}_m^{(k)}) \forall m = 1,2,...,M_k; j \neq k$

Then the classes are linearly separable
This classifier is known as a **linear machine.**

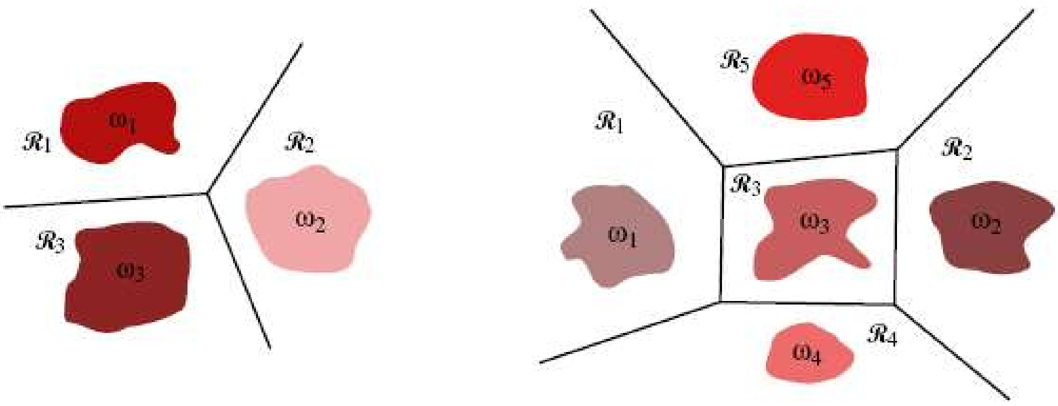


Figure 5.4: Decision boundaries produced by a linear machine for a three-class problem and a five-class problem.

Example 1: Minimum Distance to Class Means Classifier (Linear)

For S_k: $\langle \underline{y}_k \rangle \overset{\Delta}{=} \frac{1}{M_k} \sum_{m=1}^{M_k} \underline{y}_m^{(k)}$

Rule: Assign unknown $\underline{x}$ to same class at closest $<\underline{y}_k>$ using Euclidean metric.

Distance d:

$$d^2[\underline{x}, <\underline{y}_k>] = \sum_{n=1}^{N} [x_n - <y_{nk}>]^2$$

$$= [\underline{x} - <\underline{y}_k>]^T [\underline{x} - <\underline{y}_k>]$$

$$= \underline{x}^T \underline{x} - 2\underline{x}^T \langle \underline{y}_k \rangle + \langle \underline{y}_k \rangle^T \langle \underline{y}_k \rangle$$

Let $g_k(\underline{x}) = -\frac{1}{2} d^2[\underline{x}, <\underline{y}_k>] + \frac{1}{2} \underline{x}^T \underline{x}$

$g_k(\underline{x}) = \underline{x}^T <\underline{y}_k> - \frac{1}{2} <\underline{y}_k>^T <\underline{y}_k> = \underline{w}^T \underline{x} + w_{N+1}$

Example 2: Minimum Distance to Class Member Classifier

$D(\underline{x}, S_k) = \min_{m=1,\dots,M_k} \{d(\underline{x}, \underline{y}_m^{(k)})\}$

Decision Rule:

$\underline{x} \in S_j$ if $D(\underline{x}, S_j) = \min_k D(\underline{x}, S_k)$

Assign $\underline{x}$ to the same class as the nearest prototype.

.

Generalized Linear Discriminant Functions to Nonlinear Classifiers

Example 1: Quadratic Discriminant Function

$$g_k(\underline{x}) = \sum_{n=1}^{N}[w_{nn}^{(k)}x_n^2 + w_n^{(k)}x_n] + \sum_{n=1}^{N-1}\ \sum_{j=n+1}^{N} w_{nj}^{(k)}x_n x_j + w_{N+1}^{(k)}$$

$$g_k(\underline{x}) = \underline{x}^T \underline{A}_k \underline{x} + \underline{x}^T \underline{b}_k + w_{N+1}^{(k)}$$

Weights in symmetric matrix $\underline{A}_k$

Vector $\underline{b}_k$

No. of terms – N square terms
N linear
N(N-1)/2 cross product terms
1 constant term

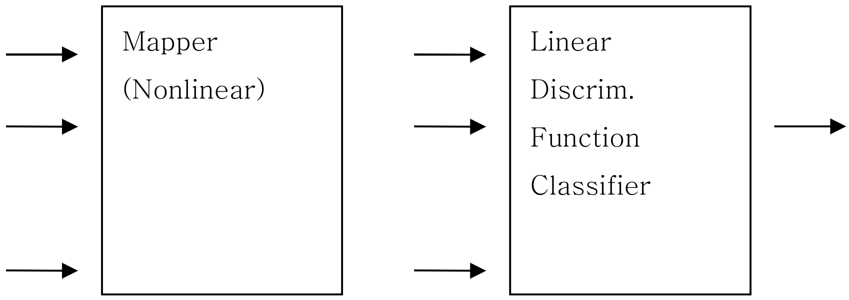
Let's generalize

Nonlinear, but can be cast in the form of a linear classifier

Let $\underline{f}=[x_1^2,...,x_N^2, x_1,...,x_N, x_1x_2, x_1x_3,...,x_{N-1}x_N]^T = [f_1,....,f_N, f_{N+1},..., f_{2N}, f_{2N+1},..., f_T]^T$

where T=N(N+3)/2

$$g_k(\underline{x}) = w_1^{(k)} f_1 + w_2^{(k)} f_2 + ... + w_T^{(k)} f_T + w_{T+1}^{(k)} = w^{(k)} \bullet f + w_{T+1}^{(k)}$$



It is also called as the Φ **Machine**

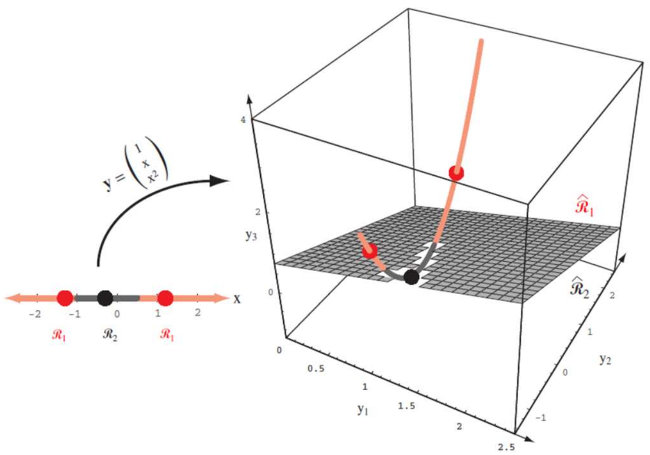


Figure 5.5: The mapping $\mathbf{y} = (1, x, x^2)^t$ takes a line and transforms it to a parabola in three dimensions. A plane splits the resulting $\mathbf{y}$ space into regions corresponding to two categories, and this in turn gives a non-simply connected decision region in the one-dimensional x space.

Example 2: Higher Order Polynomial Discriminant Functions

Can be extended to any r-th order polynomial discriminant function.

Let $\Phi(x) = w_1 f_1(\underline{x}) + w_2 f_2(\underline{x}) + \ldots + w_M f_M(\underline{x}) + w_{M+1}$

where $f_i(\underline{x})$ are linearly independent, real, single-valued functions independent of the weights.

Example:

i) $f_i(\underline{x}) = x_i$ -> linear case

ii) $f_i(\underline{x}) = x_j^n x_l^m$ $\quad j,l=1,\ldots,N \quad n,m \in [0,1]$ -> quadratic

iii) $f_i(\underline{x}) = x_{l1}^{n1}, x_{l2}^{n2}, \ldots, x_{lr}^{nr}$ -> r-th order polynomial

$l_i = 1,\ldots,N$

$n_i \in [0,1]$

We can use the Φ machine to map the r-th order polynomial nonlinear discriminant function classifier into a linear classifier that operates in a higher dimensional space.

However, we do not know how to solve for the weights yet.

It's coming soon! The techniques are known as linear training algorithms ☺