Linear discriminants

The input for the learning task (the training data) is the pairs: (x_i, y_i), where $x_i = (x_i(1), \dots, x_i(n))$ is a feature vector, and y_i is the desired prediction. Consider a simple linear function that attempts to predict y from x:

$$y \approx a_0 + a(1)x(1) + \ldots + a(n)x(n)$$

In vector notation:

$$y \approx a^T x + a_0$$

Here both a and x are n-dimensional vectors. Another way of writing it is to consider an extension of the vector x that includes a bias. Then we can write

$$y \approx a^T x$$

where both a and x are $n+1$-dimensional vectors. The goal of learning is to determine the coefficients vector a from the training data. Given m training examples $(x_1, y_1), \dots, (x_m, y_m)$, the MSE method estimates the vector a as the vector that gives the best solution to the following system of m equations with $n+1$ unknowns:

$$\begin{aligned} a^T x_1 &= y_1 \\ a^T x_2 &= y_2 \\ &\vdots \\ a^T x_m &= y_m \end{aligned}$$

In matrix notation this can be written as:

$$Xa = y \qquad \text{where } X \text{ has } m \text{ rows and } n+1 \text{ columns, } a \text{ is } n+1 \text{ vector, } y \text{ is } m \text{ vector.}$$

The MSE solution for a can be computed as follows:

1. Compute the matrix $B = X^T X$.

2. Compute the vector $h = X^T y$.

3. Solve the linear system: $Ba = h$.

Example

x	y
0 0	-1
0 1	1
1 0	1
1 1	1

$$X = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad y = \begin{pmatrix} -1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$B = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix}, \quad h = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \quad \Longrightarrow \quad a = \begin{pmatrix} -1/2 \\ 1 \\ 1 \end{pmatrix}$$

This gives the following estimate:

$$y \approx -1/2 + x(1) + x(2)$$

x	y	appxox y
0 0	-1	-1/2
0 1	1	1/2
1 0	1	1/2
1 1	1	3/2

Observe that a simple threshold can now be used to determine the label.

> **Typically the output of linear discriminants is considered as a reduced dimension of the original problem. Another algorithm (e.g., thresholding or nearest neighbor) is then applied to compute the classification from the output of the linear discriminant.**