UNIVERSITÀ PARTHENOPE
DiST
Dipartimento di Scienze e Tecnologie

Machine Learning (part II)

Single Layer Neural Network

Angelo Ciaramella

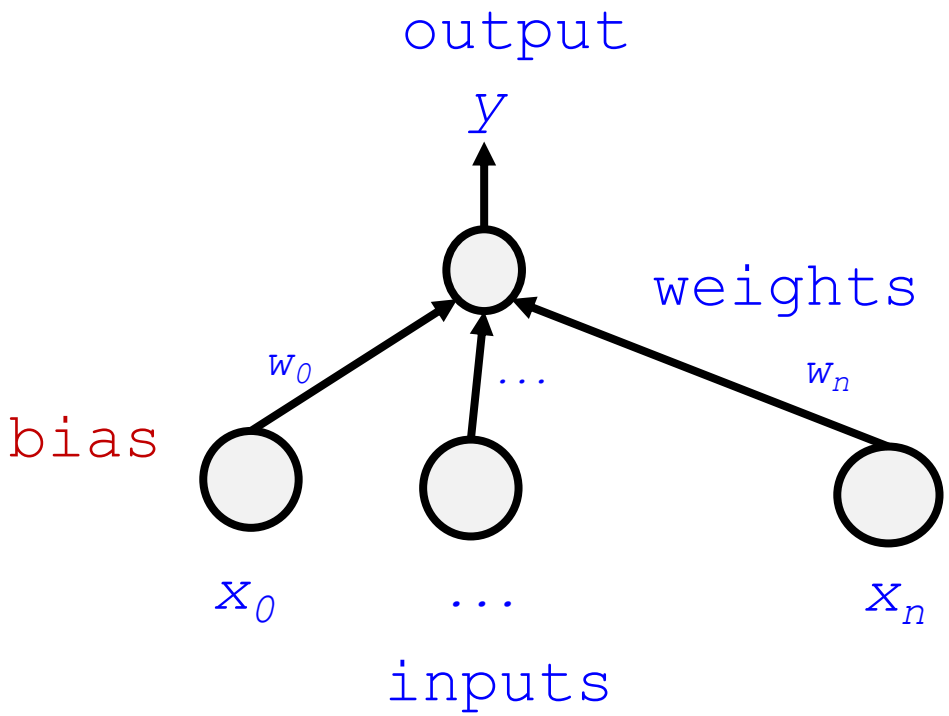
Introduction

- Linear discriminant functions
  - Linear functions of the input variables
- Generalization
  - Consider a *non-linear function*

Linear discriminant function

activation function

$$y(\mathbf{x}) = \boldsymbol{\sigma}(\mathbf{w}^T\mathbf{x} + w_0)$$

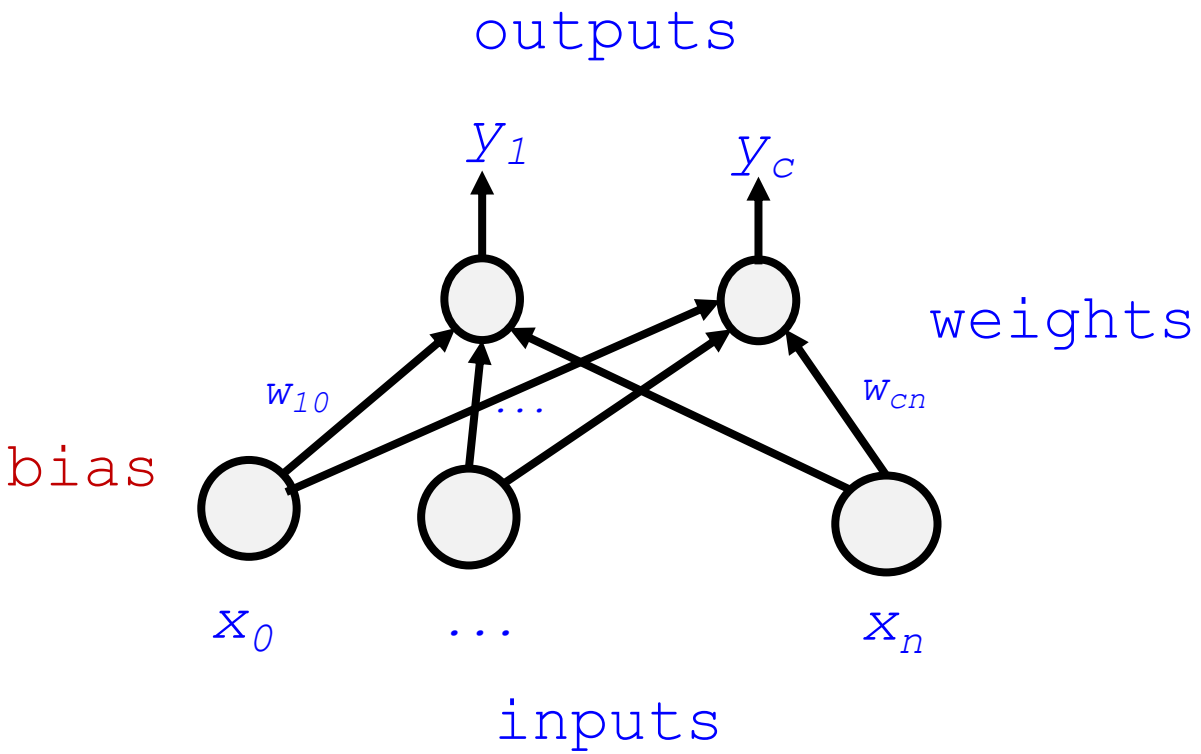


- bias is a threshold

Multiple linear discriminant

- K > 2 classes

$$y_k(\mathbf{x}) = \boldsymbol{\sigma}(\mathbf{w}_k{}^T\mathbf{x} + w_{k0})$$



- bias is a threshold

The rules of Probability

- Sum rule

joint probability

$$p(X) = \sum_{Y} p(X, Y)$$

- Product rule

$$p(X, Y) = p(Y, X)$$

$$p(X, Y) = p(Y|X)p(X)$$

- Bayes' theorem

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

Probabilistic view

- Classification
  - probabilistic view of classification
  - models with linear decision boundaries arise from simple assumptions about the distribution of the data

- Posterior probability of class C_1

a posterior probability — likelihood — a prior probability

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$

normalization factor

$$= \frac{1}{1+\exp(-a)} = \sigma(a)$$

Probabilistic view

- Making decision
  - X-ray image x
  - $P(C_1)$ probability that a person has cancer
  - $P(C_1 | x)$ probablity that a person has cancer after oberving information of X-ray

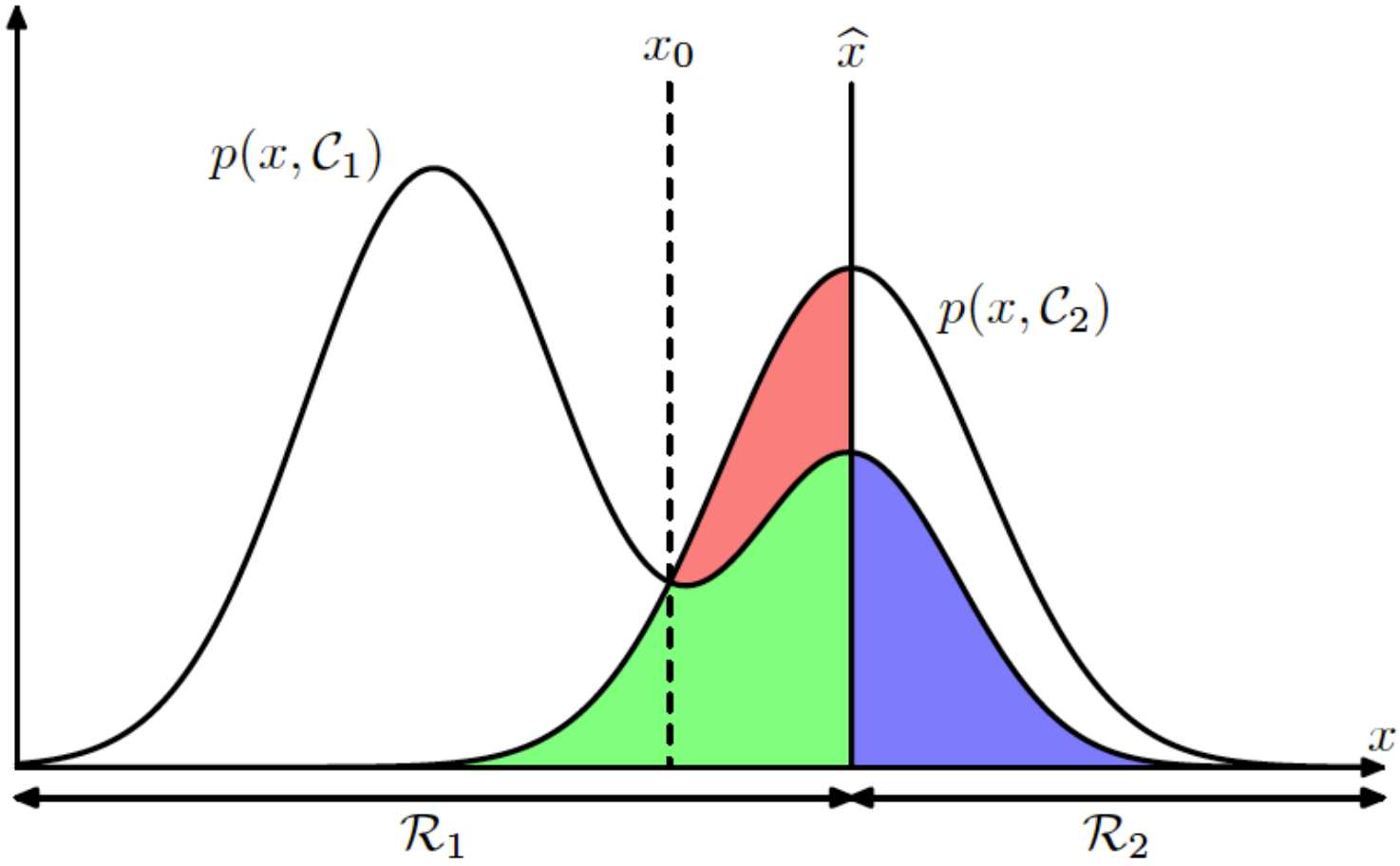
Minimizing the misclassification rate

- Decision boundaries

$$
\begin{aligned}
p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\
&= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\, \mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\, \mathrm{d}\mathbf{x}
\end{aligned}
$$

$$
\begin{aligned}
p(\text{correct}) &= \sum_{k=1}^{K} p(\mathbf{x} \in \mathcal{R}_k, \mathcal{C}_k) \\
&= \sum_{k=1}^{K} \int_{\mathcal{R}_k} p(\mathbf{x}, \mathcal{C}_k)\, \mathrm{d}\mathbf{x}
\end{aligned}
$$

Minimizing the misclassification rate



Minimizing the misclassification rate

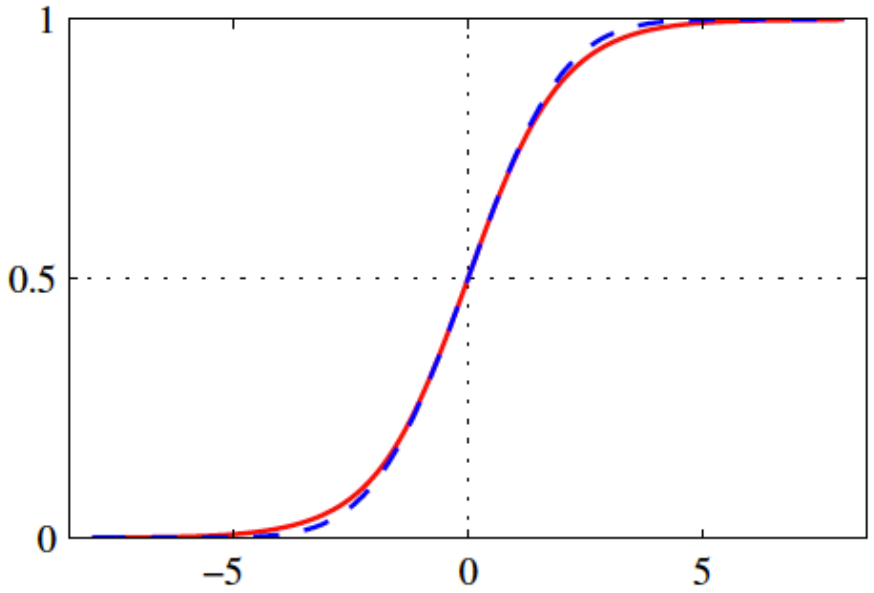
Probabilistic view

- where

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$

- and logistic sigmoid function

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$



Logistic sigmoid

- Symmetry property

$$\sigma(-a) = 1 - \sigma(a)$$

- Inverse (logit)

$$a = \ln\left(\frac{\sigma}{1-\sigma}\right)$$

Probabilistic view

- K > 2

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)}$$

$$= \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

normalized exponential or softmax function

$$a_k = \ln p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k).$$

$$a_k \gg a_j \qquad p(\mathcal{C}_k|\mathbf{x}) \simeq 1$$

$$p(\mathcal{C}_j|\mathbf{x}) \simeq 0$$

Probabilistic view

- Gaussian class-conditional density

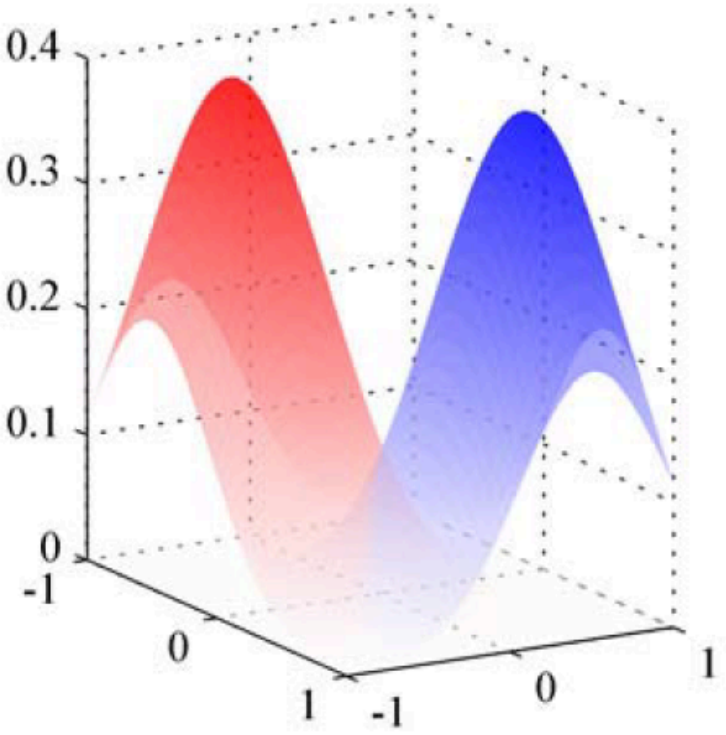
$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

- Posterior probability

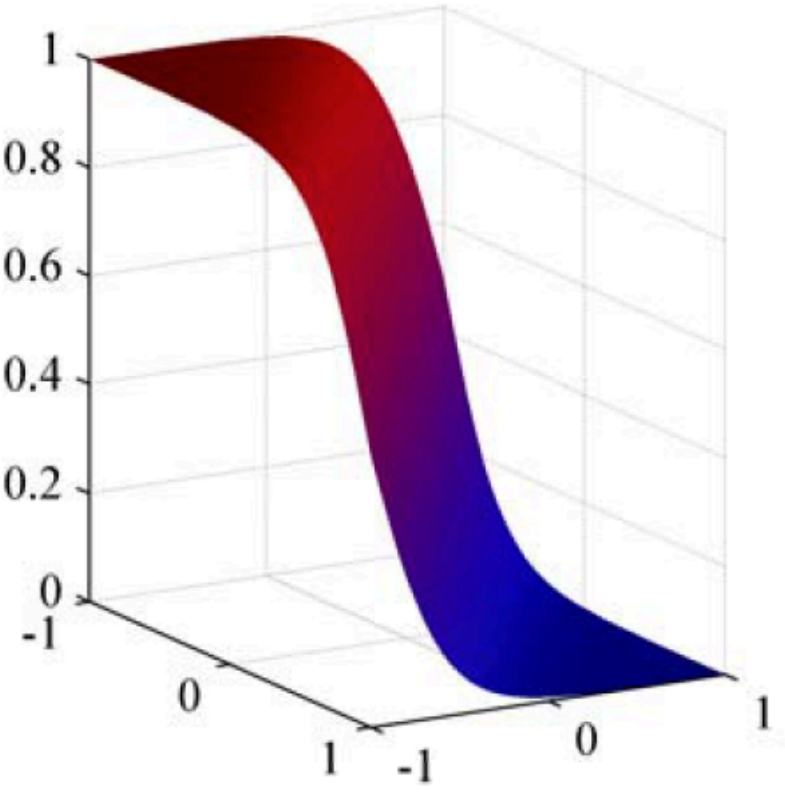
$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0)$$

$$\begin{aligned} \mathbf{w} &= \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \\ w_0 &= -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \end{aligned}$$

Probabilistic view



Class-conditional densities
for two classes



$p(C_1|\mathbf{x})$

Probabilistic view

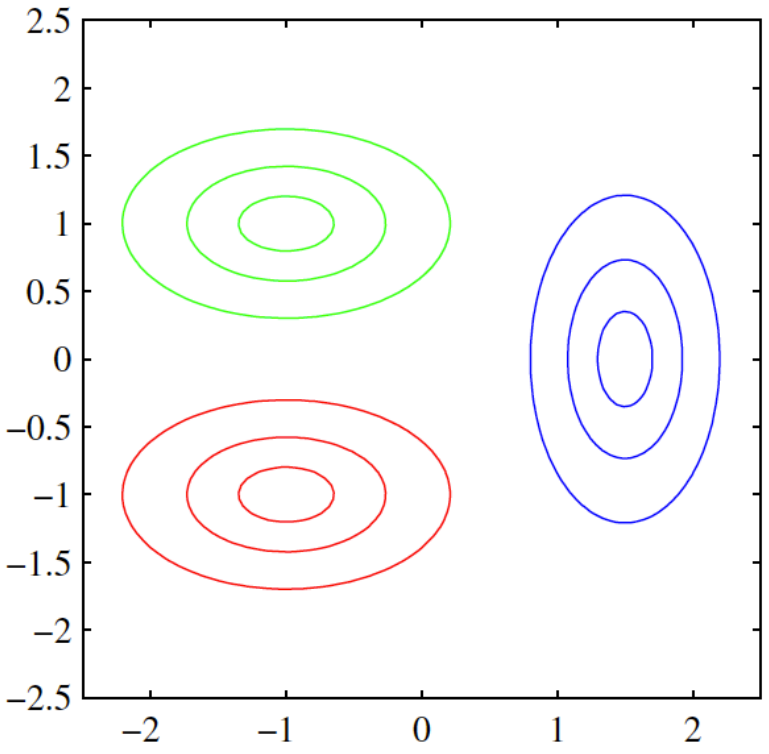
- K > 2

$$a_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

- where

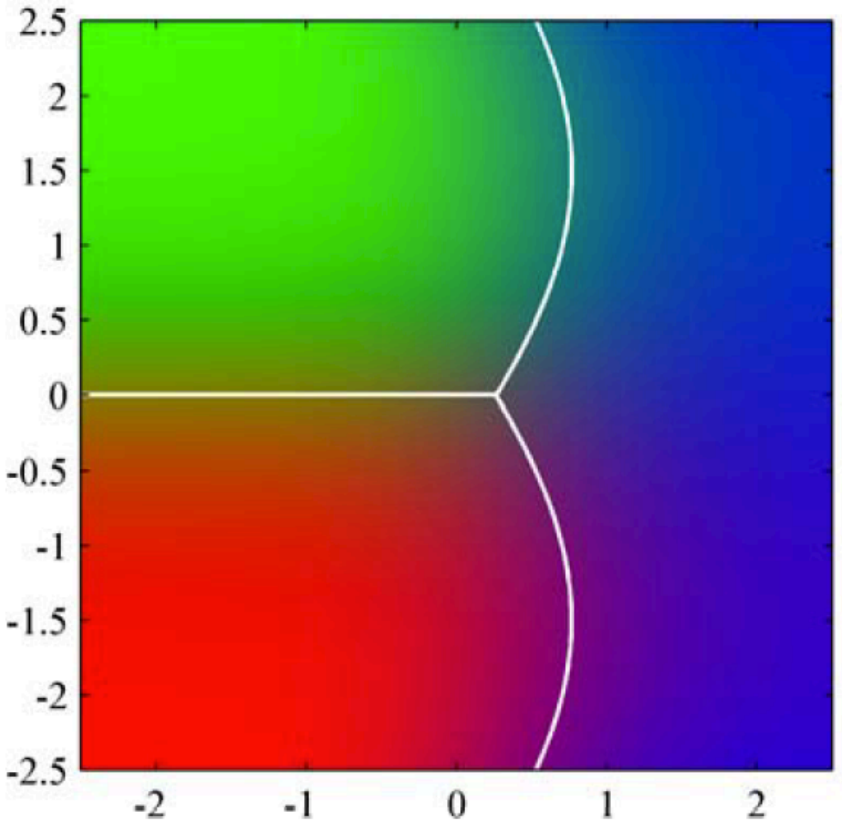
$$\begin{aligned} \mathbf{w}_k &= \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k \\ w_{k0} &= -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(\mathcal{C}_k) \end{aligned}$$

Probabilistic view



Class-conditional densities
for three classes

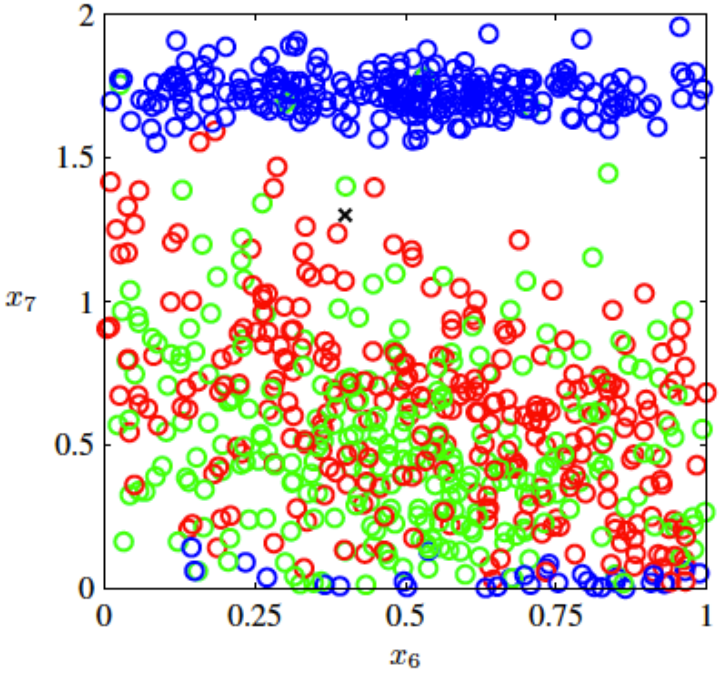
Posterior probabilities

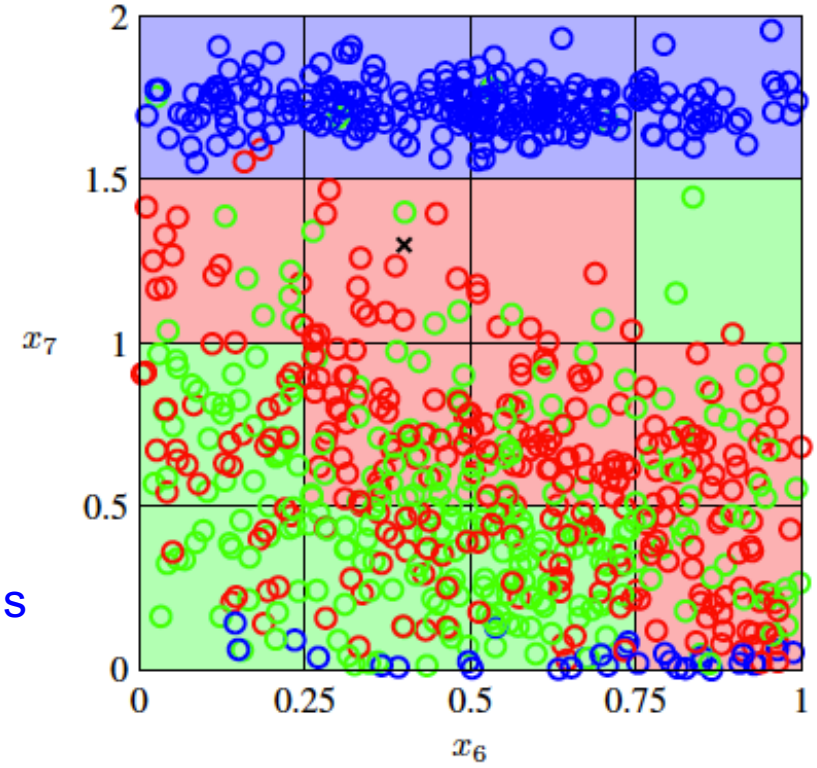


Fisher's linear discriminant

- Curse of dimensionality
  - The design of a good classifier becomes rapidly more difficult as the dimensinality of the input space increases
  - Pre-processing
    - To reduce its dimensionality
    - Fisher discriminant aims to achieve an optimal linear dimensionality reduction

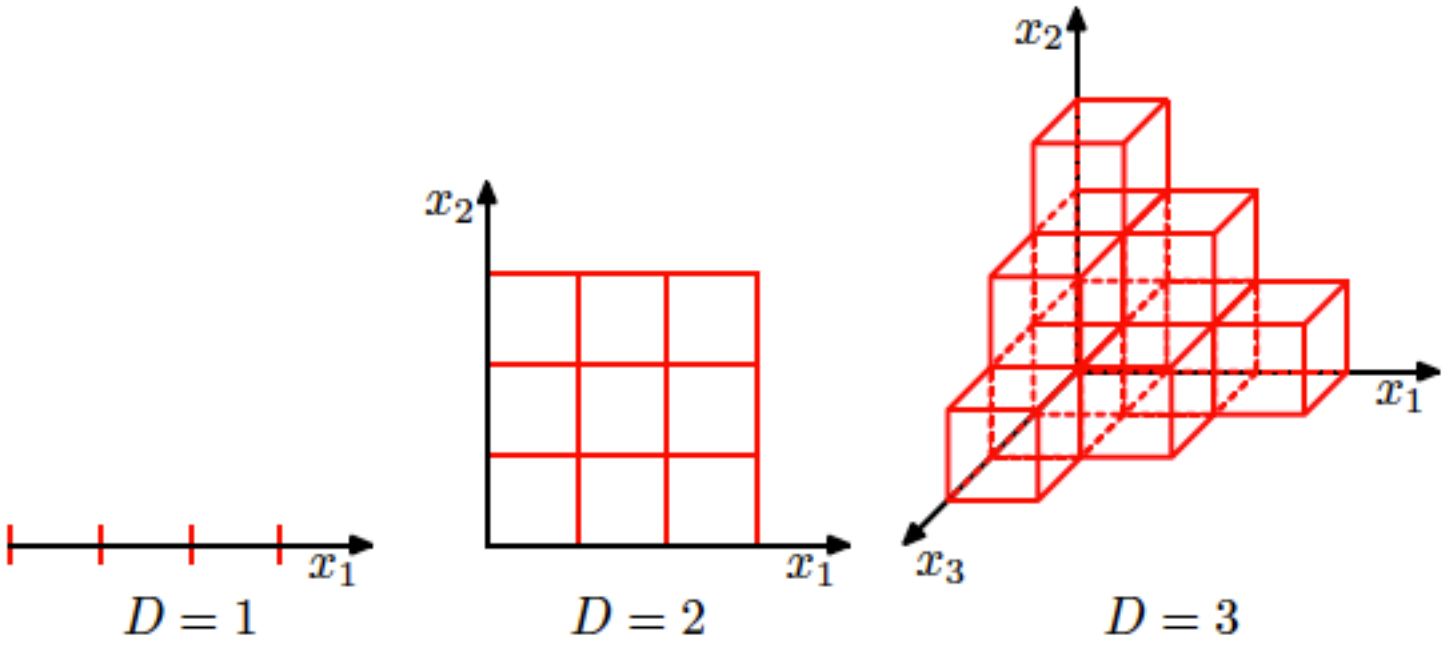
Fisher's linear discriminant





Classification by voting in the cells

Fisher's linear discriminant



The number of regions of a regular grid grows exponentially with the dimensionality of D

Fisher's linear discriminant

- Projection

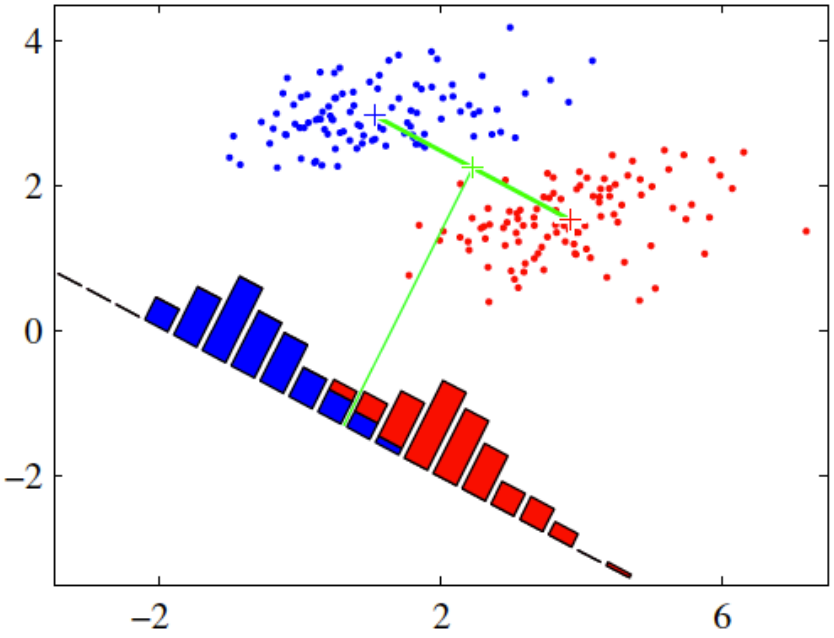
$$y = \mathbf{w}^{\mathrm{T}}\mathbf{x}. \qquad y \geqslant -w_0 \text{ as class } \mathcal{C}_1$$

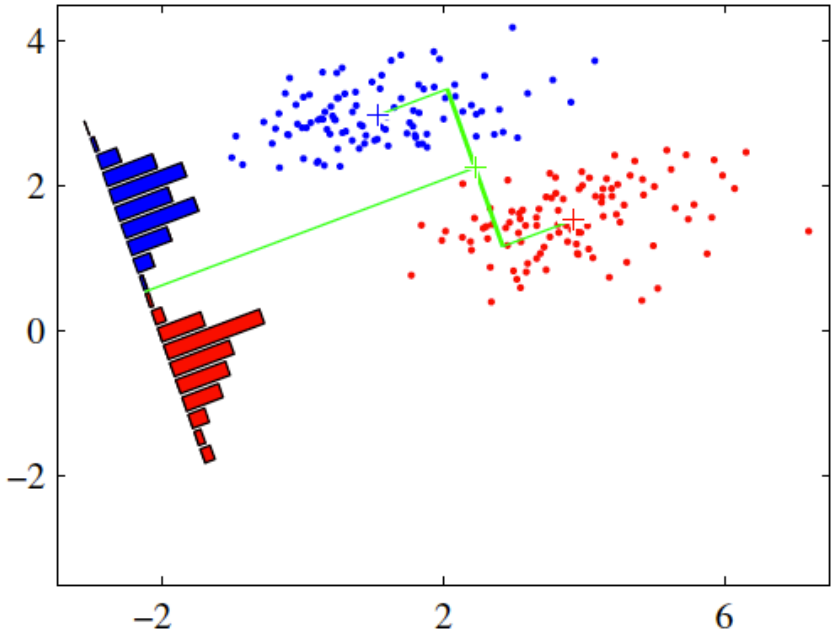
- Projection that maximizes the class separation

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{n \in \mathcal{C}_1} \mathbf{x}_n, \qquad \mathbf{m}_2 = \frac{1}{N_2}\sum_{n \in \mathcal{C}_2} \mathbf{x}_n.$$

$$m_2 - m_1 = \mathbf{w}^{\mathrm{T}}(\mathbf{m}_2 - \mathbf{m}_1)$$

Fisher's linear discriminant





Fisher directions

Fisher's linear discriminant

- within-class variance

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2$$

- Fisher criterion

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

between-class variance

within-class variance

Fisher's linear discriminant

- Fisher criterion

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}}$$

- Between-class covariance

$$\mathbf{S}_{\mathrm{B}} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}}$$

- within-class covariance

$$\mathbf{S}_{\mathrm{W}} = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^{\mathrm{T}} + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^{\mathrm{T}}$$

Fisher's linear discriminant

- Differentianting with respect to weights

direction of $(m_2 - m_1)$

$$(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w})\mathbf{S}_{\mathrm{W}}\mathbf{w} = (\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w})\mathbf{S}_{\mathrm{B}}\mathbf{w}$$

scalar factor scalar factor

$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

Fisher linear discriminant

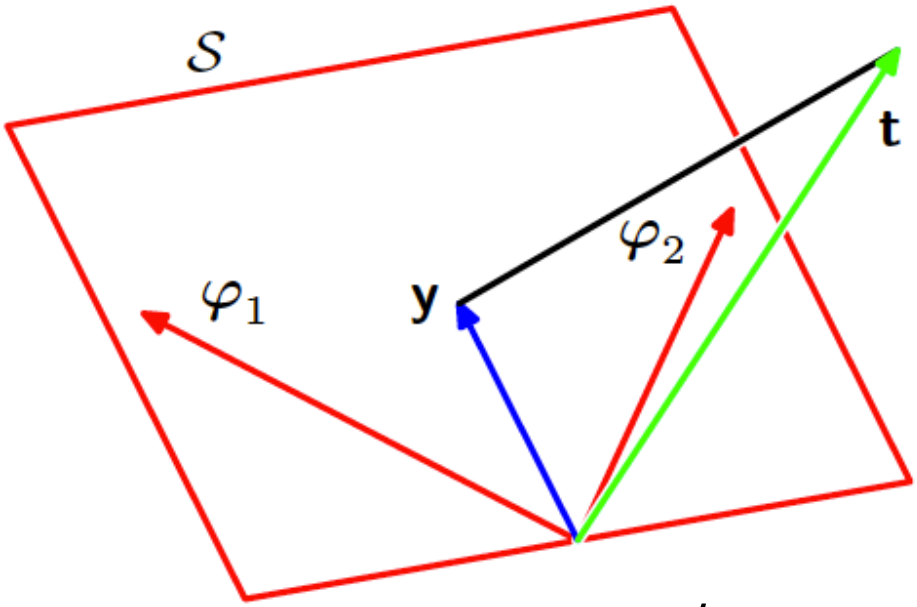
Generalization for more classes

Sum-of squares error

- Quadratic error function

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{c}\{y_k(\mathrm{x}^n;\mathrm{w}) - t_k^n\}^2$$

Geometrical representation



$$y = \sum_{j=0}^{M} w_j \phi_j^n$$

$$\frac{\partial E}{\partial w_j} = 0 = \phi_j^T(y - t)$$

$y = t_{||}$ Solution

Gradient descent

- Error function

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} E^n(\mathrm{w})$$

- Stocchastic gradient descent algorithm

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E^n$$

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \frac{\partial E^n}{\partial \mathbf{w}}$$

The perceptron algorithm

- Output

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})\right)$$

- Activation function

$$f(a) = \begin{cases} +1, & a \geqslant 0 \\ -1, & a < 0 \end{cases}$$

The perceptron algorithm

- Classification

$$\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) > 0 \qquad \mathcal{C}_1$$
$$\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) < 0 \qquad \mathcal{C}_2$$
$$t \in \{-1, +1\}$$

- Perceptron criterion

$$E_{\mathrm{P}}(\mathbf{w}) = -\sum_{n \in \mathcal{M}} \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}_n t_n$$

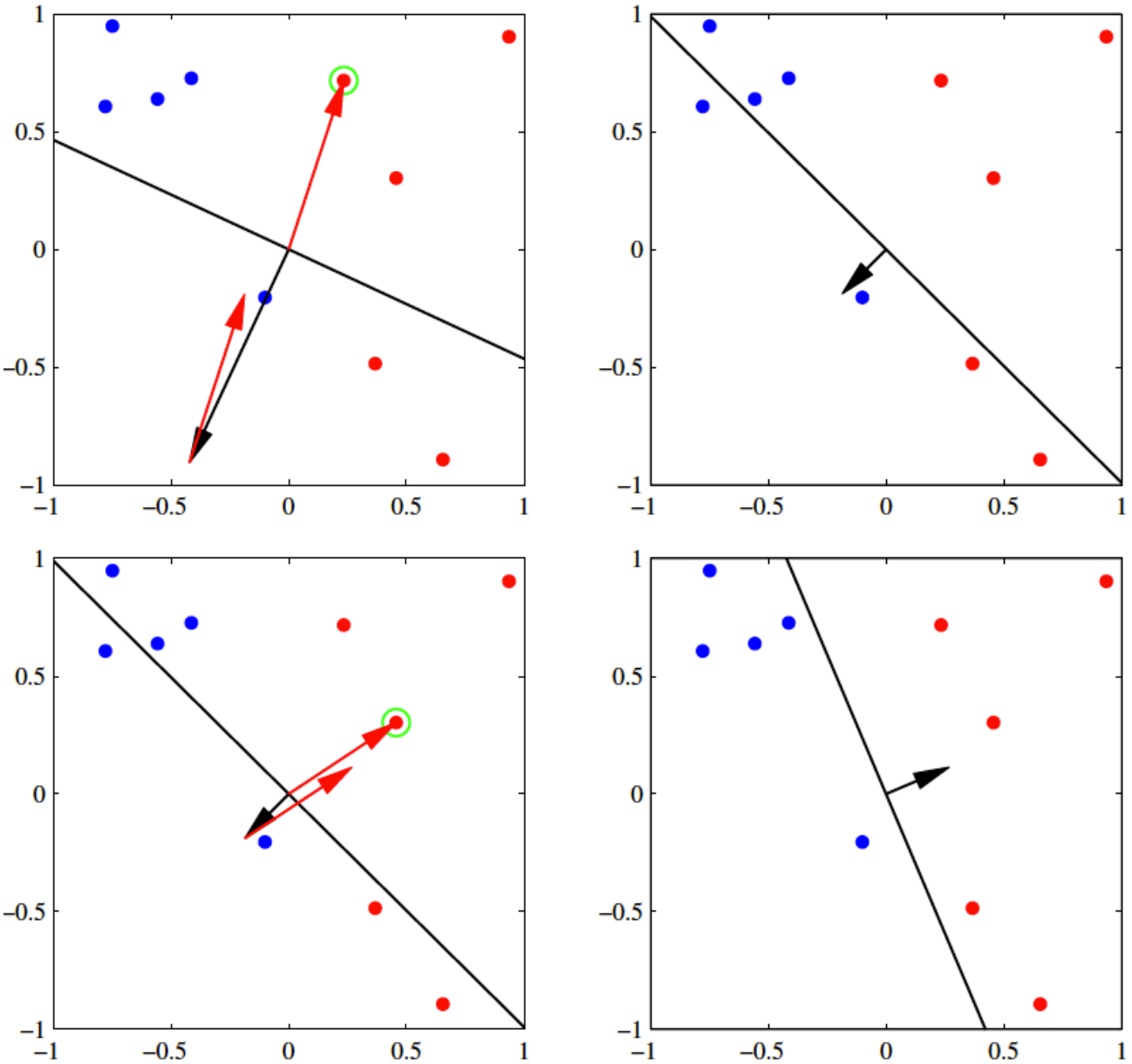
The perceptron algorithm

- Gradient descent algorithm

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_{\mathrm{P}}(\mathbf{w}) = \mathbf{w}^{(\tau)} + \eta \boldsymbol{\phi}_n t_n$$

- Perceptron convergence theorem
  - if there exists an exact solution (if the training data set is linearly separable) then the perceptron learning algorithm is guaranteed to find an exact soultion in a finite number of steps

The perceptron algorithm



Linear separability

- Linearly separble
  - The points can be classified correclty by a linear decision boundary

- No linearly separable
  - exclusive-OR (XOR) problem