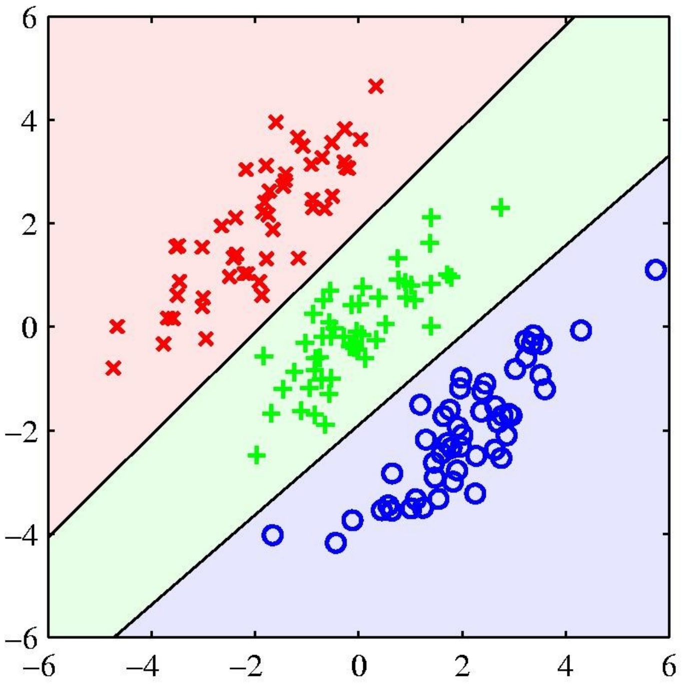
Linear Classification

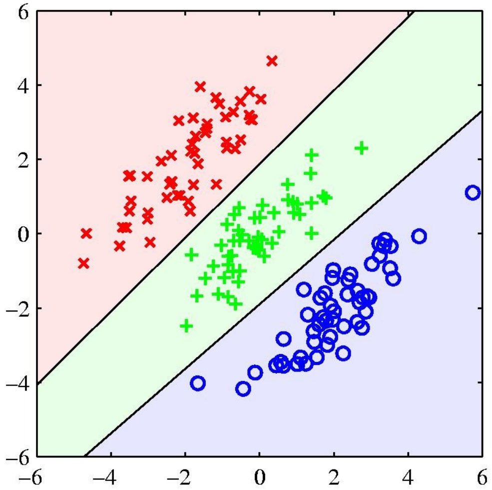


Machine Learning; Tue Apr 24, 2007

Motivation

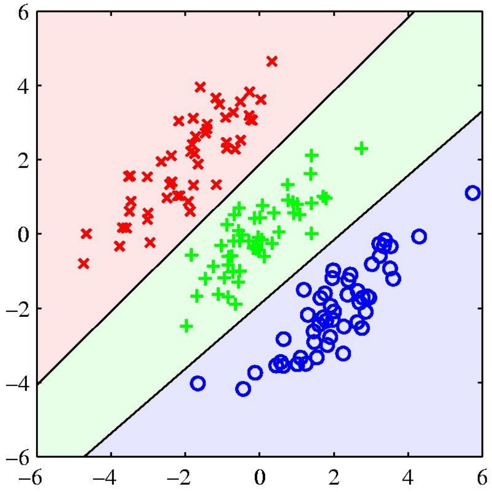
Problem: Our goal is to "classify" input vectors x into one of k classes. Similar to regression, but the output variable is discrete.

In ***linear classification*** the input space is split in (hyper-)planes, each with an assigned class.



Activation function

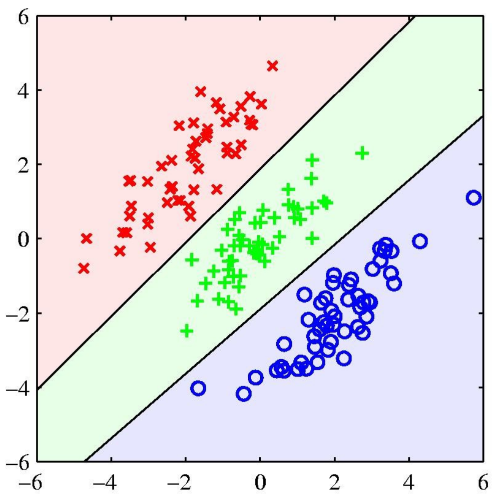
$$y(\mathbf{x}) = f\left(\mathbf{w}^T\mathbf{x}\right)$$



Non-linear function assigning a class.

Activation function

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

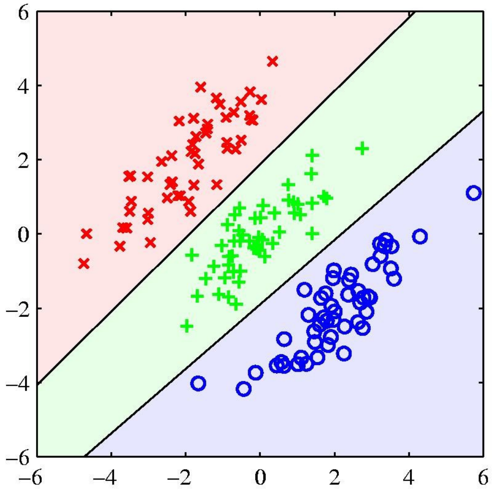


Non-linear function assigning a class.

$$\mathcal{C}_1 \text{ if } y(\mathbf{x}) > C$$
$$\mathcal{C}_2 \text{ if } y(\mathbf{x}) \leq C$$

Activation function

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

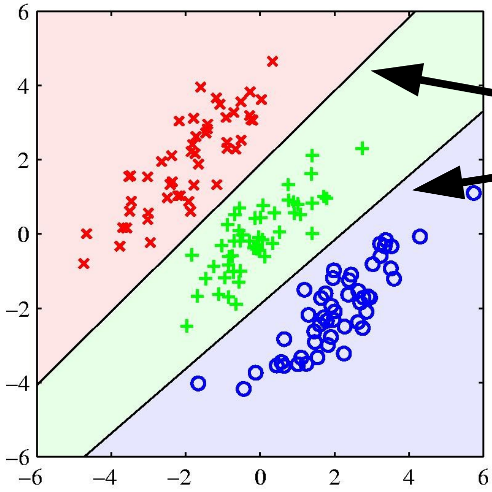


Non-linear function assigning a class.

Due to f the model is ***not*** linear in the weights.

Activation function

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



Decision surface

Non-linear function assigning a class.

Due to f the model is ***not*** linear in the weights.

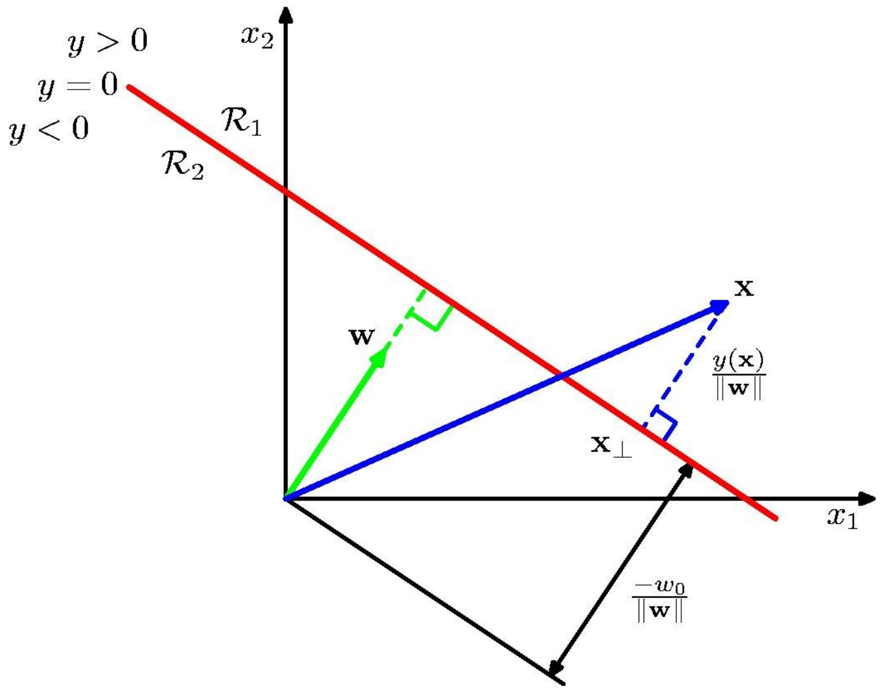
The decision surfaces ***are*** linear in w and x.

Discriminant functions

A simple linear discriminant function:

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

$$\begin{cases} \mathcal{C}_1 & y(\mathbf{x}) \geq 0 \\ \mathcal{C}_2 & y(\mathbf{x}) < 0 \end{cases}$$



Discriminant functions

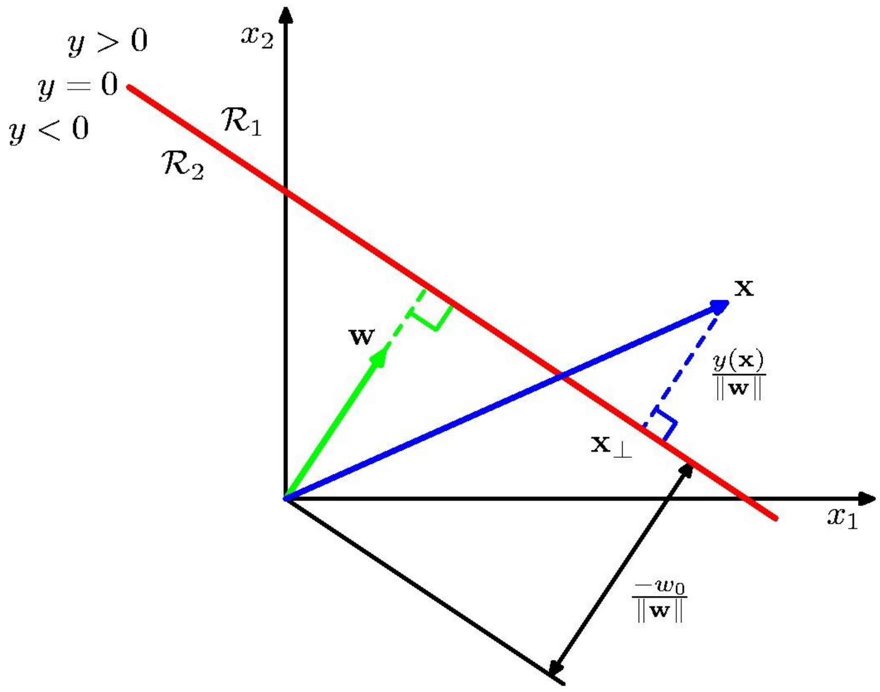
A simple linear discriminant function:

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

$$\begin{cases} \mathcal{C}_1 & y(\mathbf{x}) \geq 0 \\ \mathcal{C}_2 & y(\mathbf{x}) < 0 \end{cases}$$

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

$$\mathrm{argmax}_k y_k(\mathbf{x})$$



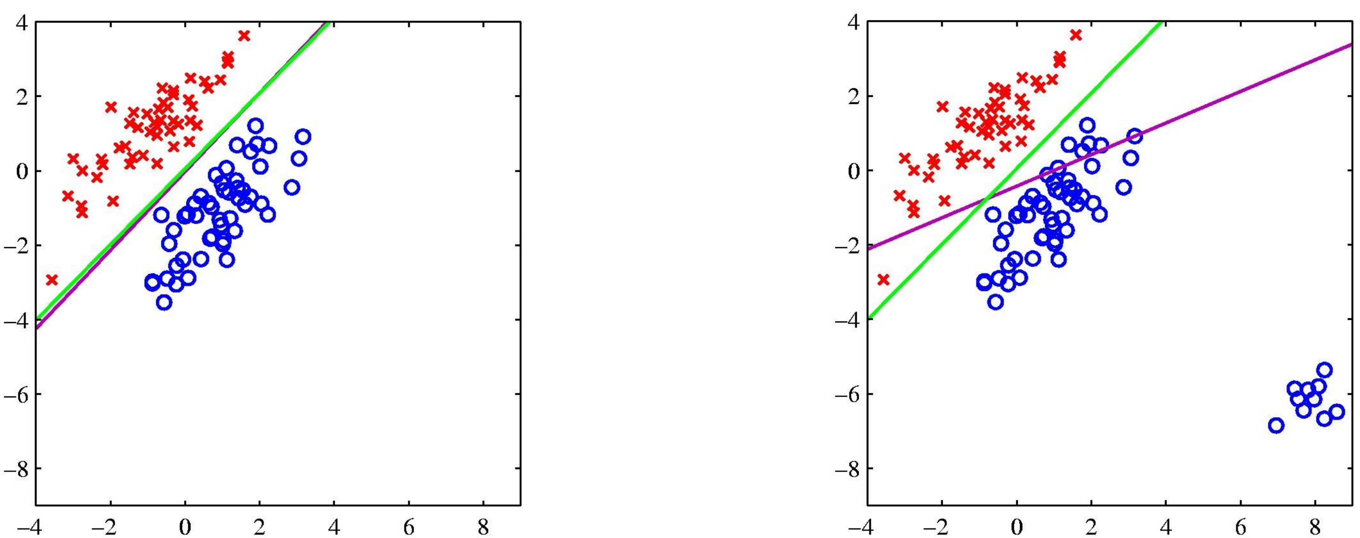
Least square training

Target vectors as bit vectors.
Classification vectors the h functions.

$$E_D(\mathbf{w}) = \sum_{n=1}^{N} \left(t_n - \sum_{k=1}^{K} y_k(\mathbf{x}_n, \mathbf{w}) \right)^2$$

Least square training

Least square is appropriate for Gaussian distributions, but has major problems with discrete targets…

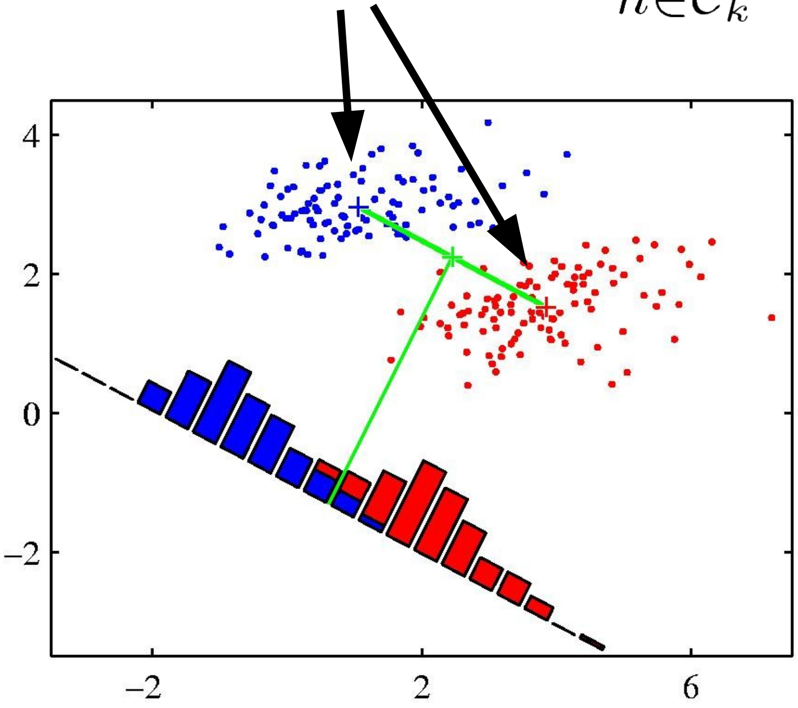


Fisher's linear discriminant

Consider the classification a projection:

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{x}_n$$

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



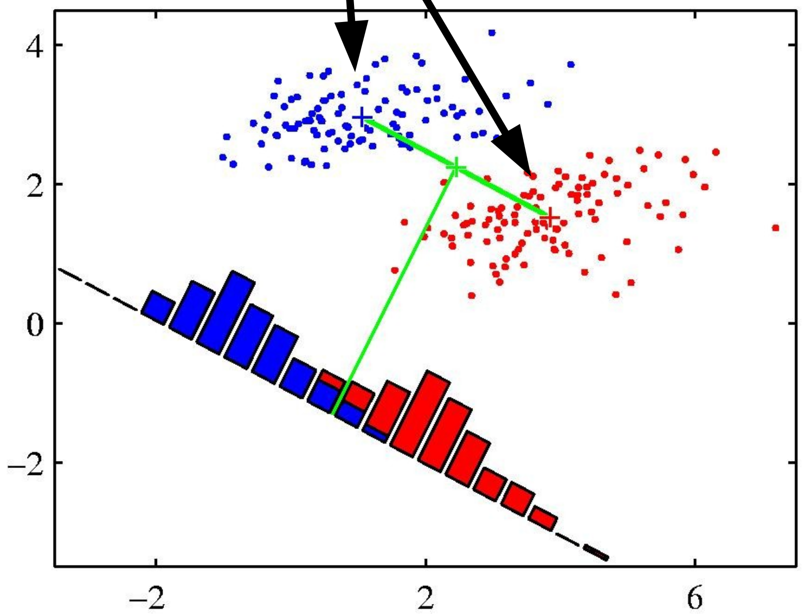
Approach: maximize this distance

Fisher's linear discriminant

Consider the classification a projection:

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{x}_n$$

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



Approach: maximize this distance

But notice the large overlap in the histograms.

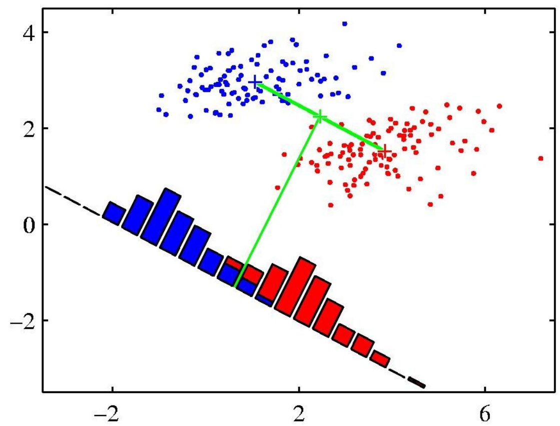
The variance in the projection is larger than it need be.

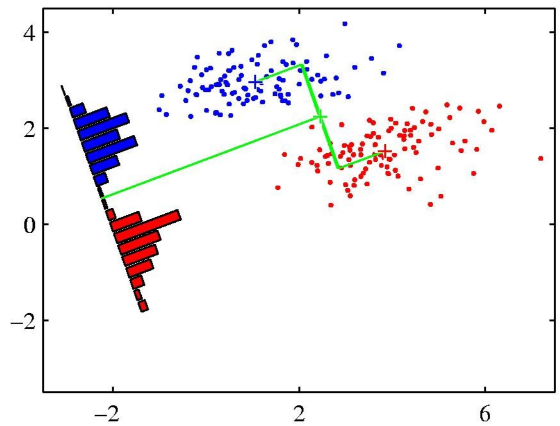
Fisher's linear discriminant

Maximize difference in mean ***and*** minimize within-class variance:

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

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



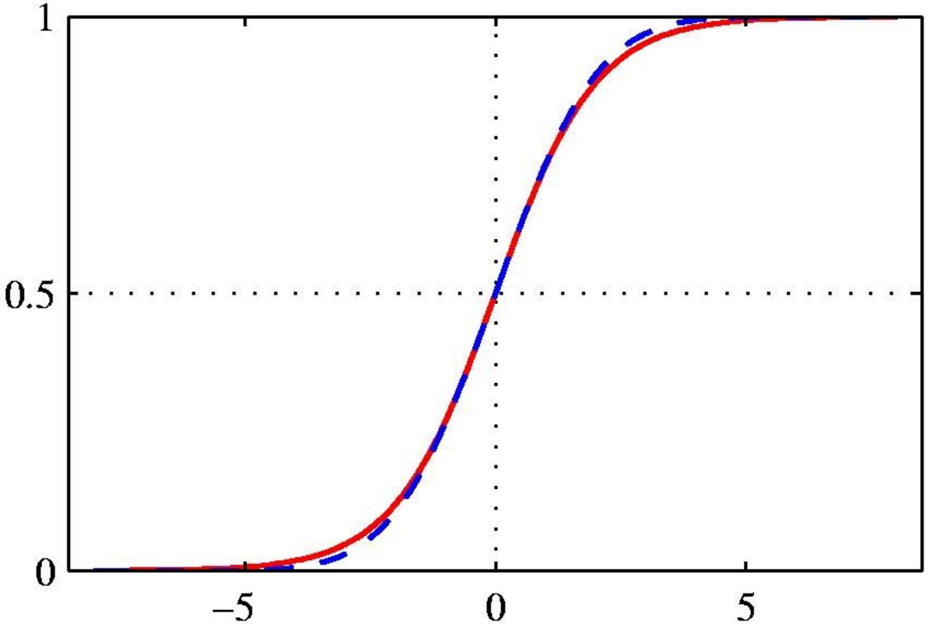


Probabilistic models

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

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

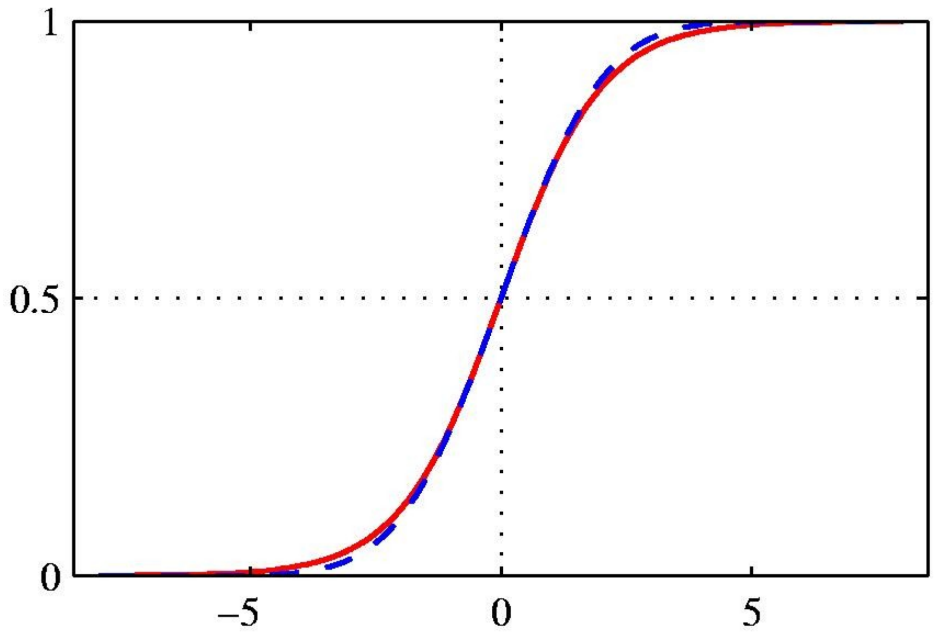
$$a = \ln \frac{p(\mathbf{x} \mid \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x} \mid \mathcal{C}_2)p(\mathcal{C}_2)}$$



Probabilistic models

Approach: Define conditional distribution and make decision based on the sigmoid activation

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



Probabilistic models

Particularly simple expression for Gaussian regression

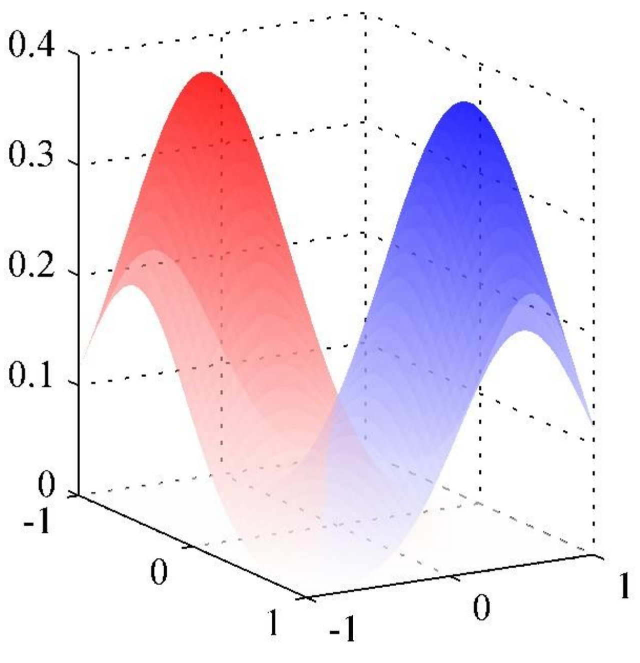
$$p(\mathbf{x} \mid \mathcal{C}_1) = N(\mathbf{x} \mid \mu_1, \sigma^2)$$

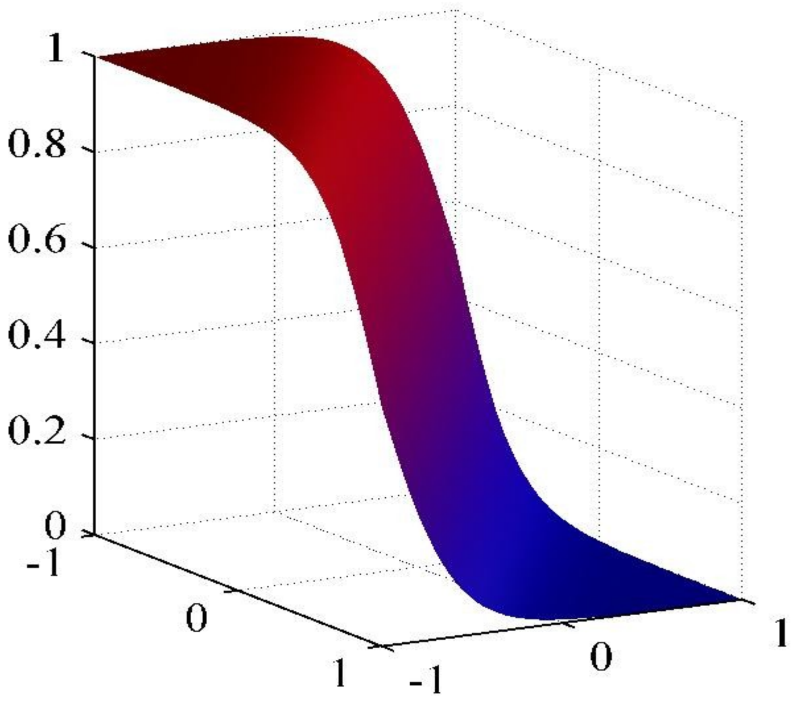
$$p(\mathcal{C}_1 \mid \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$$

$$a = \ln \frac{p(\mathbf{x} \mid \mathcal{C}_1) p(\mathcal{C}_1)}{p(\mathbf{x} \mid \mathcal{C}_2) p(\mathcal{C}_2)} = \ln \frac{\exp\left(-\frac{1}{2}(\mathbf{x} - \mu_1)^T \Sigma^{-1} (\mathbf{x} - \mu_1)\right)}{\exp\left(-\frac{1}{2}(\mathbf{x} - \mu_2)^T \Sigma^{-1} (\mathbf{x} - \mu_2)\right)} + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

$$\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_2) \qquad w_0 = -\frac{1}{2}\mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2}\mu_2^T \Sigma^{-1} \mu_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

Probabilistic models





Maximum likelihood estimation

Assume observed iid $\mathcal{D} = \{(\mathbf{t}_n, \mathbf{x}_n)\}$

$$p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\mathbf{x}_n \mid \mathcal{C}_1) = \pi N(\mathbf{x}_n \mid \mu_1, \Sigma)$$

$$p(\mathbf{x}_n, \mathcal{C}_2) = p(\mathcal{C}_2)p(\mathbf{x}_n \mid \mathcal{C}_2) = (1 - \pi)N(\mathbf{x}_n \mid \mu_2, \Sigma)$$

$$p(\mathcal{D} \mid \pi, \mu_1, \mu_2, \Sigma) =$$

$$\prod_{n=1}^{N} [\pi N(\mathbf{x}_n \mid \mu_1, \Sigma)]^{t_n} [(1 - \pi)N(\mathbf{x}_n \mid \mu_2, \Sigma)]^{1-t_n}$$

Maximum likelihood estimation

$$\log \mathrm{lhd}(\pi) \propto \sum_{n=1}^{N} \{t_n \log \pi + (1 - t_n) \log(1 - \pi)\}$$

$$\hat{\pi} = \frac{N_1}{N}$$

$$\log \mathrm{lhd}(\mu_1) \propto \sum_{n=1}^{N} t_n (\mathbf{x}_n - \mu_1) \Sigma^{-1} (\mathbf{x}_n - \mu_1)$$

$$\hat{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n \mathbf{x}_n \qquad \hat{\mu}_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1 - t_n) \mathbf{x}_n$$

Logistic regression

We can also directly express the class probability as a sigmoid (without implicitly having an underlying Gaussian):

$$p(\mathcal{C}_1 \mid \phi) = \sigma(\mathbf{w}^T \phi)$$

Logistic regression

We can also directly express the class probability as a sigmoid (without implicitly having an underlying Gaussian):

$$p(\mathcal{C}_1 \mid \phi) = \sigma(\mathbf{w}^T \phi)$$

The likelihood:

$$p(\mathcal{D}_1 \mid \mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n} (1 - y_n)^{1-t_n} \qquad y_n = \sigma(\mathbf{w}^T \phi(\mathbf{x}_n))$$

Logistic regression

We can maximize the log likelihood...

$$\log p(\mathcal{D} \mid \mathbf{w}) = \sum_{n=1}^{N} \{t_n \log y_n + (1 - t_n) \log(1 - y_n)\}$$

$$\nabla \log p(\mathcal{D} \mid \mathbf{w}) = \sum_{n=1}^{N} \left\{ t_n \frac{y_n' \phi(\mathbf{x}_n)}{y_n} - (1 - t_n) \frac{y_n' \phi(\mathbf{x}_n)}{1 - y_n} \right\}$$

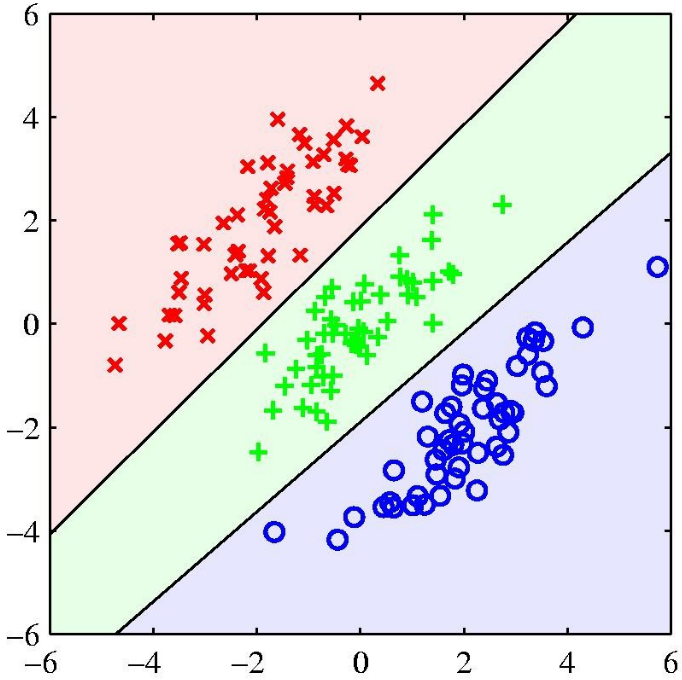
$$= \sum_{n=1}^{N} (t_n - y_n) \phi(\mathbf{x}_n)$$

$$y_n' = y_n(1 - y_n)$$

Logistic regression

Here we only estimate M weights, not M for each mean plus $O(M^2)$ for the variance in the Gaussian approach.

Summary



- Classification models
  - Linear decision surfaces
  - Geometric approach
    - Maximizing distance of means and minimizing variance
  - Probabilistic approach
    - Sigmoid functions
    - Logistic regression