Linear models for classification

Lecture 6

$\vec{x} \rightarrow C_k, \quad k = 1, \ldots, K$

$\vec{x}$: input, D dims

C_k: discrete classes

Input space divided into decision regions by decision surfaces.

For ex., $\vec{t} = (0, 1, 0, 0, 0)$ — target variables

* 1-of-K coding scheme

$K = 5$ classes, target variable indicates class 2

Classification approaches:

(1.) Discriminant function: directly assigns $\vec{x}$ to a class, e.g. for 2 classes

$y(\vec{x}) \geq 0 \Leftrightarrow C_1, \quad y(\vec{x}) < 0 \Leftrightarrow C_2.$

(2.) Probabilistic approach:

use $p(C_k|\vec{x}) = \dfrac{p(\vec{x}|C_k)\,p(C_k)}{p(\vec{x})}$ — Bayesian framework

Previously, we focused on

$y(\vec{x}) = \vec{w}^T \vec{\varphi}(\vec{x}) + w_0$ ← explicit bias prm

Now we will consider $y(\vec{x}) = f(\vec{w}^T \vec{x} + w_0)$

or, more generally, $y(\vec{x}) = f(\vec{w}^T \cdot \vec{\varphi}(\vec{x}) + w_0)$

non-linear activation f'n

Decision surfaces ($y(\vec{x}) = \text{const}$) are given by

$\vec{w}^T\vec{x} + w_0 = \text{const}$, linear f's of $\vec{x}$

Discriminant functions

(1.) Two classes

Consider $y(\vec{x}) = \vec{w}^T\vec{x} + w_0$ [linear discriminant]

$$\begin{cases} y(\vec{x}) \geq 0 \Rightarrow C_1 \\ y(\vec{x}) < 0 \Rightarrow C_2 \end{cases}$$

$y(\vec{x}) = 0 \Leftarrow$ decision boundary (DB)

Consider $\vec{x}_A, \vec{x}_B \in DB$

$$y(\vec{x}_A) = y(\vec{x}_B) = 0 \Rightarrow \vec{w}^T(\underbrace{\vec{x}_A - \vec{x}_B}_{\text{lies on } (D-1) \text{ dim'l DB}}) = 0$$

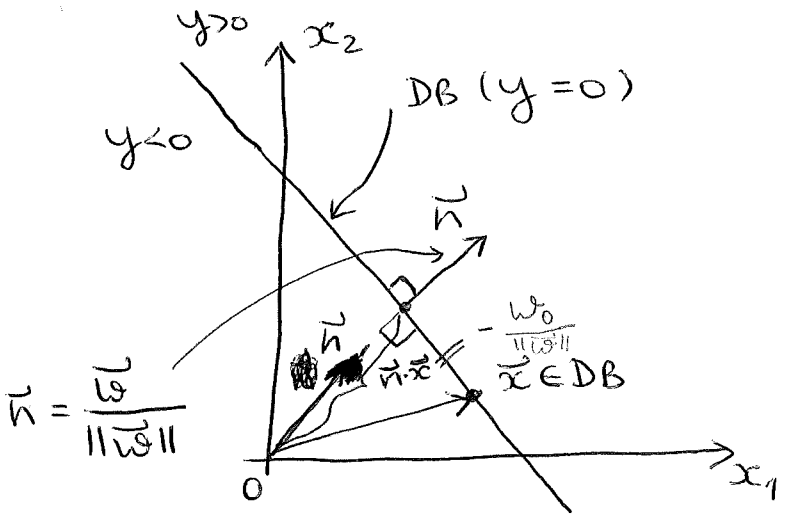
So, $\vec{w} \perp DB \Rightarrow \vec{n} = \frac{\vec{w}}{\|\vec{w}\|}$ is a unit vector $\perp DB$

Similarly, if $\vec{x} \in DB$,

$$y(\vec{x}) = \vec{w}^T\vec{x} + w_0 = 0 \text{ , or}$$

$$\underbrace{\frac{\vec{w}^T\vec{x}}{\|\vec{w}\|}}_{\text{"}\vec{n}\cdot\vec{x} = \text{normal distance from the origin to DB}} = -\frac{w_0}{\|\vec{w}\|}$$

So, w_0 determines the location of DB



D=2

Moreover, for any $\vec{x}$ (not necessarily on DB)

$$\vec{x} = \vec{x}_{\parallel DB} + \underbrace{r\vec{n}}_{\perp DB}$$

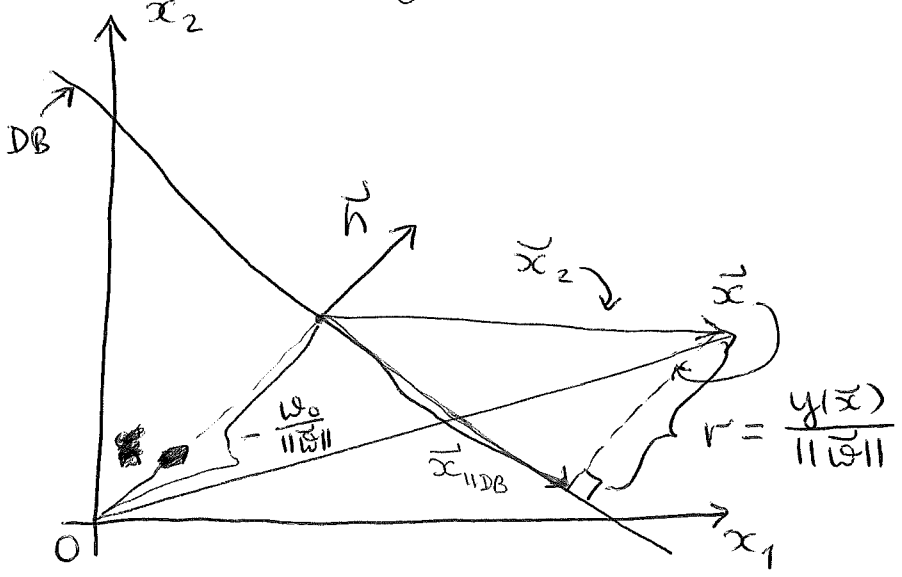
Then $\underbrace{\vec{w}^T\vec{x} + w_0}_{y(\vec{x})} = \vec{w}^T\vec{x}_{\parallel DB} + r\frac{\overbrace{\vec{w}^T\cdot\vec{w}}^{\|\vec{w}\|^2}}{\|\vec{w}\|} + w_0$, or

$$y(\vec{x}) = r\|\vec{w}\| \Rightarrow r = \frac{y(\vec{x})}{\|\vec{w}\|}.$$

↑ perpendicular distance from $\vec{x}$ to DB

Here, we used $\vec{w}^T\cdot\vec{x}_{\parallel DB} + w_0 = 0$ ← since $\vec{x}_{\parallel DB}$ is on DB by construction

Indeed,



Then

$$-\vec{n}\frac{w_0}{\|\vec{w}\|} + \underbrace{\vec{x}_2}_{\vec{x}_{2,\parallel DB} + \underbrace{\vec{x}_{2,\perp DB}}_{r\vec{n}}} = \vec{x}$$

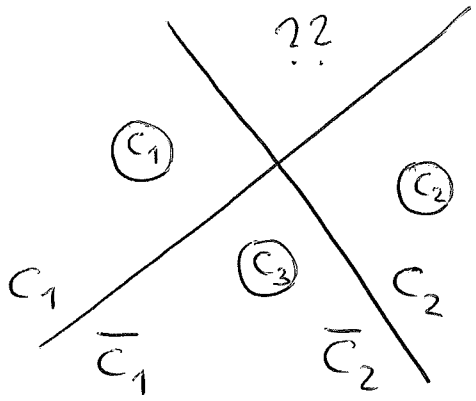
$$\vec{w}^T\cdot\vec{x} = -\underbrace{\frac{\vec{w}^T\cdot\vec{w}}{\|\vec{w}\|^2}}_{1} w_0 + \underbrace{\vec{w}^T\cdot\vec{x}_{2,\parallel DB}}_{0} + r\underbrace{\frac{\vec{w}^T\cdot\vec{w}}{\|\vec{w}\|}}_{\|\vec{w}\|}, \text{ or}$$

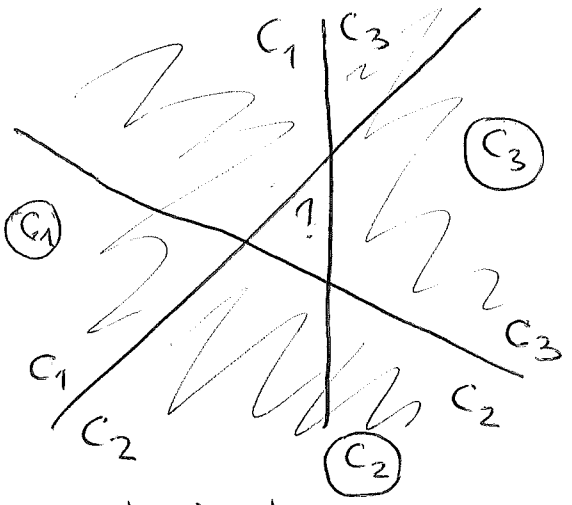
$$\underbrace{\vec{w}^T\cdot\vec{x} + w_0}_{y(\vec{x})} = r\|\vec{w}\|, \text{ as before.}$$

(2.) Multiple classes

Consider $K > 2$ classes.

Difficulties in generalizing from the $K=2$ case:





majority vote among discriminant f's

Rather, consider a single K-class discriminant:

$$y_k(\vec{x}) = \vec{w}_k^T\cdot\vec{x} + w_{k,0}$$

Assign a point to C_k if $y_k(\vec{x}) > y_j(\vec{x})$, $\forall j \neq k$.

DBs are then given by $y_k(\vec{x}) = y_j(\vec{x})$, s.t.

$$(\vec{w}_k - \vec{w}_j)^T \cdot \vec{x} + (w_{k,0} - w_{j,0}) = 0$$

Now, consider $\vec{x}_A, \vec{x}_B \in C_k$

$$\hat{\vec{x}} = \lambda \vec{x}_A + (1-\lambda)\vec{x}_B \qquad 0 \le \lambda \le 1$$

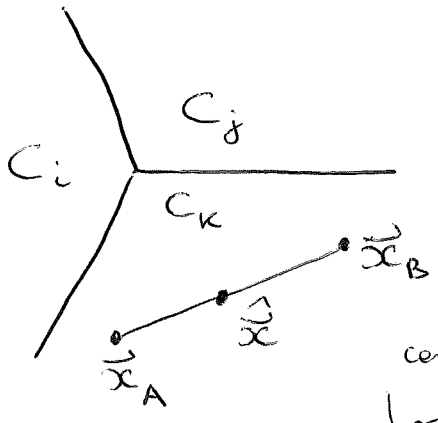
(line connecting $\vec{x}_A$ & $\vec{x}_B$)

$\Downarrow$ linearity of discriminant f's

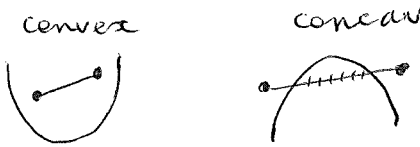
$$y_k(\hat{\vec{x}}) = \lambda \underbrace{y_k(\vec{x}_A)}_{> y_j(\vec{x}_A),\ \forall j \neq k} + (1-\lambda) \underbrace{y_k(\vec{x}_B)}_{> y_j(\vec{x}_B),\ \forall j \neq k}$$

$$y_k(\hat{\vec{x}}) > y_j(\hat{\vec{x}}) \quad , \forall j \neq k$$

So, $\hat{\vec{x}} \in C_k$ as well.



Since $\vec{x}_A, \vec{x}_B$ are arbitrary, C_k is singly connected & convex, $\forall k$.



Least squares for classification

Consider a problem with K classes, s.t. $\vec{t}$ are K-dim unit vectors.

$$y_k(\vec{x}) = \vec{w}_k^T \vec{x} + w_{k,0} \Rightarrow \vec{y}(\vec{x}) = \widetilde{W}^T \cdot \widetilde{\vec{x}}$$

(K classes)

$$\widetilde{W} = \begin{pmatrix} w_{k,0} \\ w_{k,1} \\ \vdots \\ w_{k,D} \end{pmatrix} \Big\} D+1 \text{ (\#prms) or \# entries in } \widetilde{\vec{x}}$$

k^{th} column

$$\widetilde{\vec{x}} = (1, \vec{x})^T$$

"x_0

Training set: $\{\vec{x}_n, \vec{t}_n\}$ $\quad n = 1, \dots, N$

Define $$T = \begin{pmatrix} t_{n,0} & t_{n,1} & \dots & t_{n,K-1} \end{pmatrix} \Big\} N$$

(K)

$\vec{t}_n$ is the n^{th} row of T

$$\widetilde{X} = \begin{pmatrix} 1 & & & \\ x_{n,0} & \dots & \dots & x_{n,D} \end{pmatrix} \Big\} N$$

($D+1$)

$\widetilde{\vec{x}}_n$ is the n^{th} row of $\widetilde{X}$

$\widetilde{X}\widetilde{W}$ is an $N \times K$ matrix like T

(N = #rows, K = #columns)

Then $E(\tilde{W}) = \frac{1}{2} Tr\left\{ (\tilde{X}\tilde{W} - T)^T (\tilde{X}\tilde{W} - T) \right\}$.

Indeed, $\frac{1}{2}\sum_{n,k} (\tilde{X}\tilde{W} - T)^T_{kn} (\tilde{X}\tilde{W} - T)_{nk} = \frac{1}{2}\sum_{n,k} (\tilde{X}\tilde{W} - T)_{nk} \times (\tilde{X}\tilde{W} - T)_{nk}$

$$E(\tilde{W}) = \frac{1}{2}\sum_{k=1}^{K}\sum_{n=1}^{N} \left(\underbrace{(\tilde{X}\tilde{W})_{nk}}_{\sum_{j=0}^{D} \tilde{X}_{nj}\tilde{W}_{jk} = \sum_{j=0}^{D} \tilde{x}_{n,j} w_{k,j} =} - t_{nk} \right)^2 \quad (=)$$

$$= w_{k,0} + \tilde{w}_k^T \cdot \tilde{x}$$

$$(=) \frac{1}{2}\sum_{k=1}^{K}\sum_{n=1}^{N} \left(\sum_{j=0}^{D} w_{k,j}\tilde{x}_{n,j} - t_{nk} \right)^2 .$$

Then

$$\frac{\partial E}{\partial w_{k,j}} = \frac{1}{2} 2 \sum_n \left[\sum_{j'} w_{k,j'} \tilde{x}_{n,j'} - t_{nk} \right] \tilde{x}_{n,j} = 0 \text{, yielding } \forall k, j$$

$$\underbrace{(\tilde{X}^T\tilde{X})}_{(D+1)\times(D+1)} \underbrace{\tilde{W}}_{(D+1)\times K} = \underbrace{\tilde{X}^T T}_{(D+1)\times K}$$

$(D+1)\times K$

$$\sum_n t_{nk}\tilde{x}_{n,j} = \sum_n T_{nk}\tilde{X}_{nj} = \sum_n \tilde{X}^T_{jn} T_{nk} = (\tilde{X}^T T)_{jk}$$

$$\sum_{n,j'} w_{k,j'}\tilde{x}_{n,j'}\tilde{x}_{n,j} = \sum_{n,j'} \tilde{W}_{j'k}\tilde{X}_{nj'}\tilde{X}_{nj} = \sum_{n,j'} \tilde{X}^T_{jn}\tilde{X}_{nj'}\tilde{W}_{j'k} = (\tilde{X}^T\tilde{X}\tilde{W})_{jk}$$

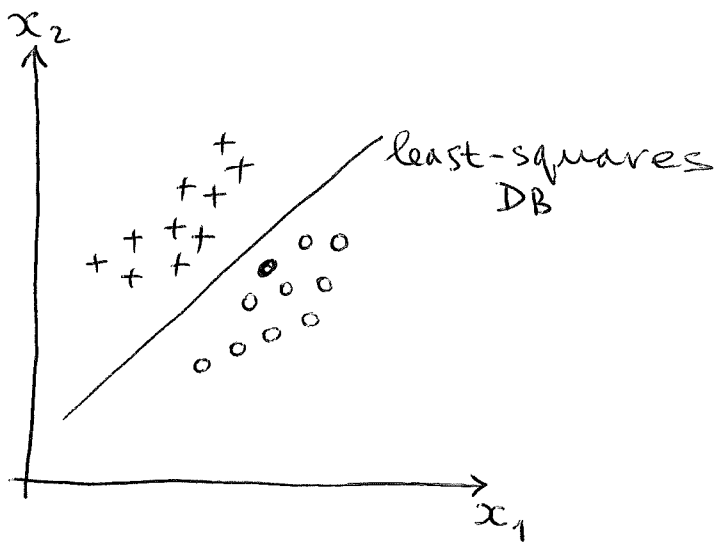
Finally, $\tilde{W} = \underbrace{(\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T}\, T$

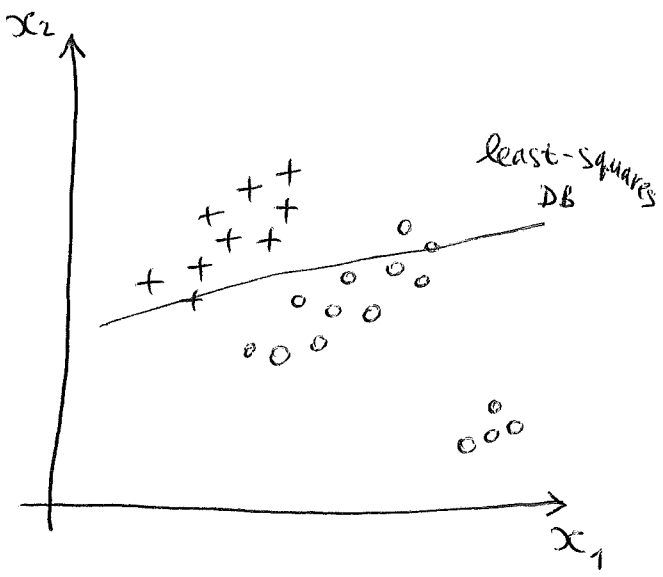
"$\tilde{X}^+$ pseudo-inverse of X"

$$\tilde{y}(\tilde{x}) = \tilde{W}^T\tilde{x} = T^T(\tilde{X}^+)^T\tilde{x} .$$

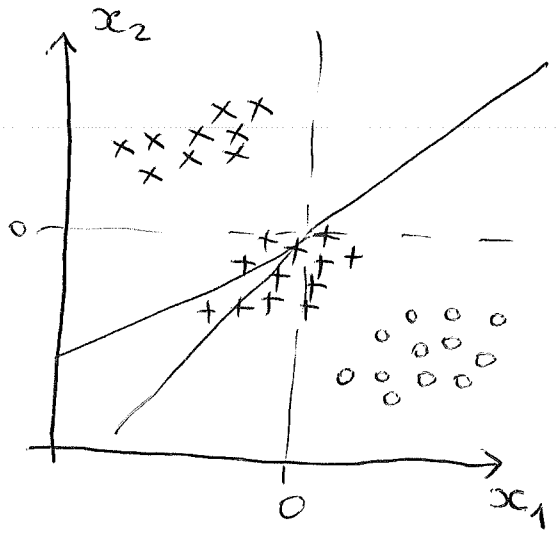
This is a closed-form solution which however is sensitive to outliers.

Ex.





And may even fail completely:



poor prediction

Indeed, least-squares assumes gaussian distr'n of $\vec{t}$'s, and binary target vectors often have non-gaussian distributions.