# Week 6 : 4.1 – 4.5

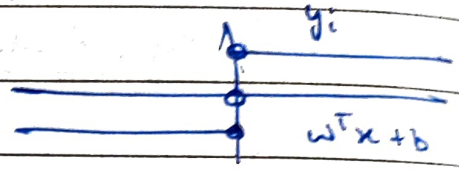
objectives :-

- Write linear classⁿ eqⁿ
- List reasons for knowing linear class.
- Write Bayesian decision fn
- Derive perception algorithm – Derive grad desc for logistic reg.
- Derive loss fn for primal SVM

- **What is linear classification ?** : ← Binary classⁿ

$x_i \in \mathbb{R}^D$ for each $x_i$ given $t_i \in \{-1, +1\}$

Find $y_i = \text{sign}(w^T x_i + b)$ $y_i \simeq t_i$

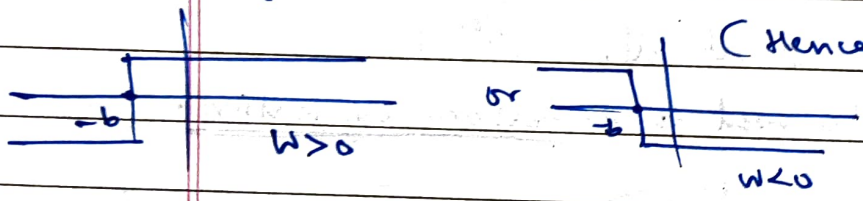



**Example 1D case :-**

$y = (wx + b)$ sign WLOG assume $w > 0 \Rightarrow$ Sign depends on $w, b$.

$= \text{sign}(x + b/w)$ [if constant is $> 0$]

(Hence $\|w\| = 1$)




Company a simple threshold.

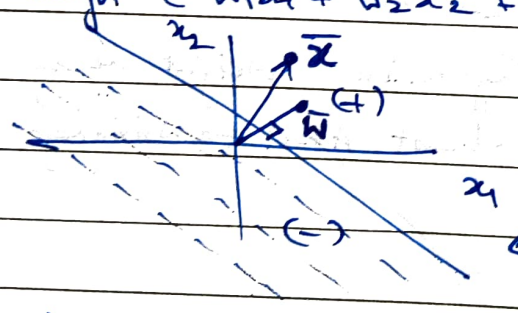
**Example 2-D case :-**

$y = \text{Sign}(w_1 x_1 + w_2 x_2 + b)$ ← dimension

$\|w\| = 1$

a line

Consider $w_1 x_1 + w_2 x_2 + b = 0$




$w^T x + b = 0$

One side +ve, one side -ve

Taking dot product with $\bar{w}$ and seeing if product $> 0$ or not

Contours if we scale $\bar{w}$ and $b$ ⟺ [Again $\|w\| = 1$]

Orienting hyperplane by $\|w\|$ around unit sphere / circle.

## (4.2) Why study linear classifiers ?

- Simplest classifiers (one of two)
- Many non linear problems can be linearized
- Natural outcome of for familiar class of class conditional densities.

## Linearising non linear problems :-

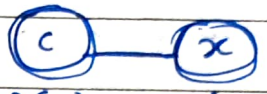
- Add desired features :
  - Powers
  - Interaction terms
  - Kernels

or use pretrained Neural Networks.

PAGE No. / DATE / /

## Bayesian decision rule for classification :

$C_1$ cats | $C_2$ dogs  $p(C|x)$




$p(C)$  $p(x|C)$

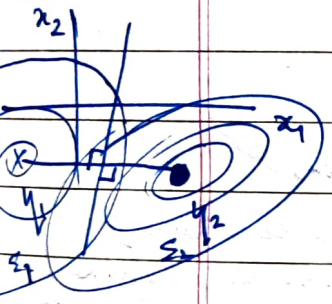
$$p(C|x) = \frac{p(x|C)\,p(C)}{p(x)} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

We may find $\frac{p(C_1|x)}{p(C_2|x)}$ and if $> 1 \Rightarrow C_1$

$< 1 \Rightarrow C_2$ [for example]

(Assuming equal risk)

## Gaussian class conditional in D- Dimensions :




$p(x|C_1)$  $p(x|C_2)$ both modelled as gaussians

$$p(x|C_j) = \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)\right)$$

D dimensions

[General setting]

Boundary may or may not be linear.

## We wish to look at :

$$\log\left(\frac{p(x|C_1)\,p(C_1)}{p(x|C_2)\,p(C_2)}\right)$$ compared with 0.

$$\ln[p(x|C_1)] + \ln[p(C_1)] - \ln[p(x|C_2)] - \ln[p(C_2)]$$

(If $\sigma_i$ & $\sigma_j$ are equal, we can cancel out the constant)

$$\Rightarrow \ln(p(x|C_1)) + \ln(p(C_1)) - \ln(p(x|C_2)) - \ln(p(C_2))$$

$$= -\frac{1}{2}(x-\mu_1)^T \Sigma^{-1} (x-\mu_1) + K_1$$

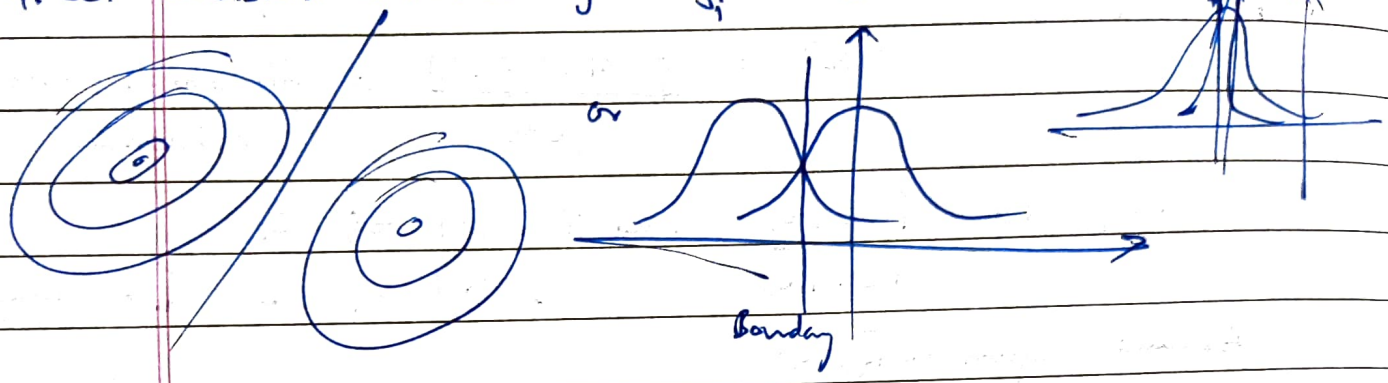
$$- \frac{1}{2}(x-\mu_2)\Sigma^{-1}(x-\mu_2) - K_2$$

We get a straight line if $\Sigma_1 = \Sigma_2 = \Sigma$.

In general we will (not) get straight line

## 4.3 Algorithm to build a Bayesian Classifier:

1. Assume parametric forms of $p(x|C_i)$
2. Set $p(C_i) = N_i/N$
3. Find the parameters of $p(x|C_i)$ using methods such as ML.
4. Set decision criteria $y = \arg_i \max \left( p(x|C_i)\, p(C_i) \right)$




## 4.4 Perceptron learning Algorithm & loss:-

$$L(w) = -\sum_{i \in M} (w^T x_i + b)\, t_i$$

$> 0$ for +ve  $< 0$ for -ve

(Misclassified  (We wish to achieve zero loss)

If points are not lin. sep no other zero loss.


**Hinge loss:-**

$\max(0, -(w^T x_i + b)\, t_i)$  Convex fns can do gradient descent

$w \leftarrow w - \eta \nabla_w L(w)$

$w + \eta\, x_i\, t_i \quad i \in M$ (Misclassified)

Infinite solutions if we have a lin. sep problem.

## 4.5 Linear classification - Logistic Regression:

PAGE No. / DATE / /

Circling back to Bayesian Decisions:

$$a \triangleq \log\left(\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}\right)$$

We want to estimate $P(C_1|x) = \sigma(a) = \frac{1}{1+\exp(-a)}$

[Now 'a' is a linear fn of x only if both are Gaussian and have the same $\Sigma$.]

$$= \frac{P(x|C_1)P(C_1)}{\sum_i P(x|C_i)P(C_i)}$$

### Gradient Descent for logistic regression:

$p(t|w)$ $[t \in \{0,1\}]$ maximise

$$= \prod_{i=1}^{N} y_i^{t_i}(1-y_i)^{1-t_i}$$ (iid hence product)

When $t=0$ / $t=1$ $\quad y_i = P(t_i=1 \mid x_i, w)$

We take log transformation as log is ~~convex~~ monotonic & then we may deal with $\Sigma$ terms instead of $\Pi$ terms

We get:

$$\therefore \min_w \left[-\sum_{i=1}^{N} t_i \log(y_i) + (1-t_i)\log(1-y_i)\right]$$

$y_i = w^T x_i + b$ if assume Gaussian & same $\Sigma$

Cross Entropy loss between $t_i$ & $y_i$ (Minimise cross entropy)

$$-\int \log(q)\, p\, dx$$

$$\nabla_w L = \sum_i (y_i - t_i)x_i$$ This + gradient descent = solution.

$$y_i = \sigma(w^T x_i + b) + \lambda_i \|w\|_2^2$$ (Hence regularization can be done.)

[Generalized linear Models]