Generally build a classifier w/ discriminant functions δ_c for each of my classes $c = 1, \dots, K$ so that

$\delta_c(x)$ = large if x signifies that the assoc. y is likely to be in class c

So

$$\hat{y} = \arg\max_c \delta_c(x)$$

we said a Bayes' classifier

$$\delta_c(x) = \mathbb{P}(Y=c \mid X=x)$$

<u>Defn</u>: A classifier is linear if some (increasing) function of the δ_c's is linear.

$\Rightarrow$ In this case the decision boundaries btwn classes are lines.

<u>Linear Discriminant Analysis</u> (linear classifier)

$$\delta_c(x) = \mathbb{P}(Y=c \mid X=x)$$

$$\propto \underbrace{\mathbb{P}(X=x \mid Y=c)}_{N(\mu_c, \sigma^2)} \underbrace{\mathbb{P}(Y=c)}_{\pi_c}$$

Showed last time w/ algebra

$$\delta_c(x) = x\frac{\mu_c}{\sigma^2} - \frac{\mu_c^2}{2\sigma^2} + \log(\pi_c)$$

$$\delta_c(x) = x\underbrace{\frac{\mu_c}{\sigma^2}}_{\beta_c} - \underbrace{\frac{\mu_c^2}{2\sigma^2} + \log(\pi_c)}_{\beta_{0c}}$$

$x \in \mathbb{R}$

Unknown parameters μ_c, π_c, σ^2

we estimate these using training data

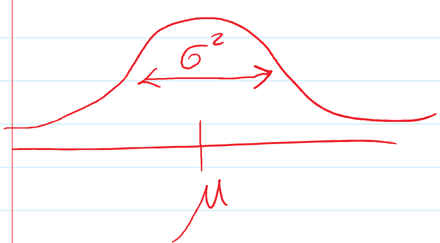
$\hat{\mu}_c$ = mean of x_ns in class c

$\hat{\pi}_c$ = % of x_ns in class c

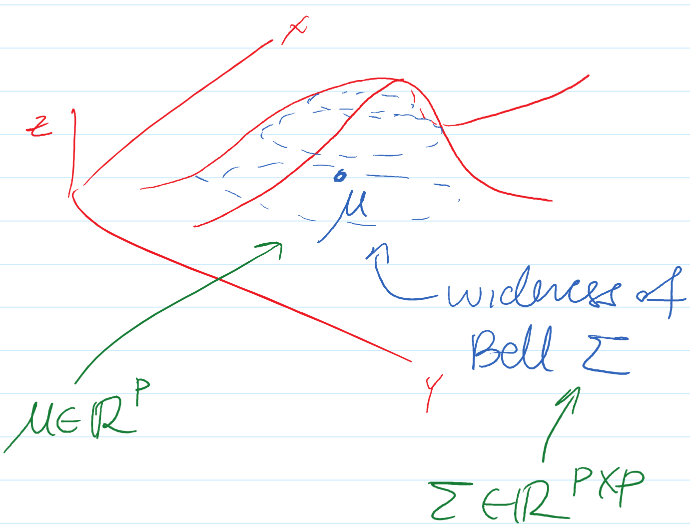
$\hat{\sigma}^2$ = pooled variance
= calc. sample var. separately for each class and then take weighted avg.

We can expand to when $x \in \mathbb{R}^p$ using multivariate normal distribution

Univariate



Multivariate



$\Sigma_{ij} = Cov(x_i, x_j)$

PDFs

Univariate:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

Univariate:

$$f(x) = \underbrace{\frac{1}{\sqrt{2\pi\sigma^2}}}_{(2\pi)^{-1/2}(\sigma^2)^{-1/2}} \exp\Big(\underbrace{-\frac{1}{2\sigma^2}(x-\mu)^2}_{-\frac{1}{2}(x-\mu)(\sigma^2)^{-1}(x-\mu)}\Big)$$

Multivariate:

$$f(x) = (2\pi)^{-p/2} \det(\Sigma)^{-1/2} \exp\Big(\underbrace{-\frac{1}{2}\underbrace{(x-\mu)^T}_{1\times p}\underbrace{\Sigma^{-1}}_{p\times p}\underbrace{(x-\mu)}_{p\times 1}}_{1\times 1}\Big)$$

$$\delta_c(x) = \underbrace{P(X=x \mid Y=c)}_{N(\mu_c, \Sigma)} \underbrace{P(Y=c)}_{\pi_c}$$

$\mu_c \in \mathbb{R}^{p\times 1}$ ↑ $\Sigma \in \mathbb{R}^{p\times p}$

$$= (2\pi)^{-p/2} \det(\Sigma)^{-1/2} \exp\Big(-\frac{1}{2}(x-\mu_c)^T \Sigma^{-1}(x-\mu_c)\Big)\pi_c$$

$$\delta_c(x) \leftarrow \log \delta_c(x) = -\frac{1}{2}(x-\mu_c)^T\Sigma^{-1}(x-\mu_c) + \log \pi_c$$

$$= -\frac{1}{2}(x^T\Sigma^{-1} - \mu_c^T\Sigma^{-1})(x-\mu_c) + \log \pi_c$$

$$= -\frac{1}{2}(\cancel{x^T\Sigma^{-1}x} - x^T\Sigma^{-1}\mu_c + \mu_c^T\Sigma^{-1}\mu_c$$

$$- \mu_c^T\Sigma^{-1}x) + \log \pi_c$$

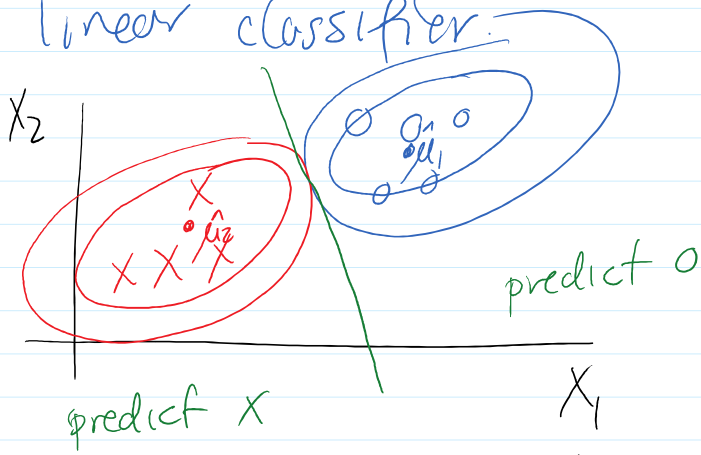
$$\boxed{\delta_c(x) = x^T\underbrace{\Sigma^{-1}\mu_c}_{\beta} \underbrace{- \frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c + \log \pi_c}_{\beta_0}}$$

$$= x^T\beta + \beta_0$$

↑ linear!

So even in the multivariable $(p > 1)$ this

So even in the multivariable ($P > 1$) this is a linear classifier.



How do I estimate $\hat{\mu}_c$ and $\hat{\Sigma}$?

$\hat{\mu}_c$ = mean vector of x_n's in class c

$\hat{\Sigma}_{ij}$ = pooled covariance btwn x_{ni}'s and x_{nj}'s.

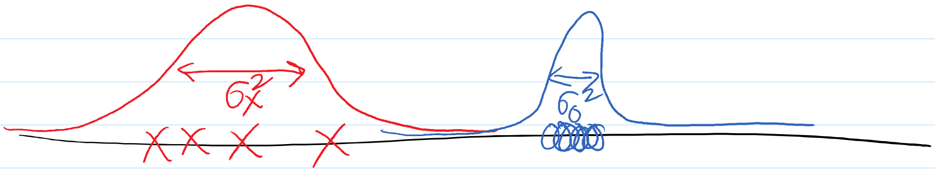
Just like in regression I can always create more dimensions w/ transformations (grow P)

$$X = \begin{bmatrix} | & | & | \\ X & X^2 & \log X \\ | & | & | \end{bmatrix}$$

Quadratic Discriminant Analysis (not linear)

LDA: assumes equal variances $(\hat{\mu}_c, \hat{\pi}_c, \Sigma)$

QDA: relax this, classes can have different vars.
$(\hat{\mu}_c, \hat{\pi}_c, \Sigma_c)$



Discr. fn for QDA:

$$\delta_c(x) = \underbrace{\mathbb{P}(X = x \mid Y = c)}_{N(\mu_c,\, \Sigma_c)} \underbrace{\mathbb{P}(Y = c)}_{\pi_c}$$

For $p = 1$

quadratic

$$\delta_c(x) = -\log \sigma_c - \frac{(x-\mu_c)^2}{2\sigma_c^2} + \log(\pi_c)$$

For $p > 1$

$$\delta_c(x) = -\log \det \Sigma_c - \frac{1}{2}(x-\mu_c)^T \Sigma^{-1} (x-\mu_c) + \log \pi_c$$

	LDA	v. QDA
# parameters	$(K-1)(p+1) \approx Kp$	$(K-1)\left(\frac{p(p+3)}{2}+1\right) \approx Kp^2$

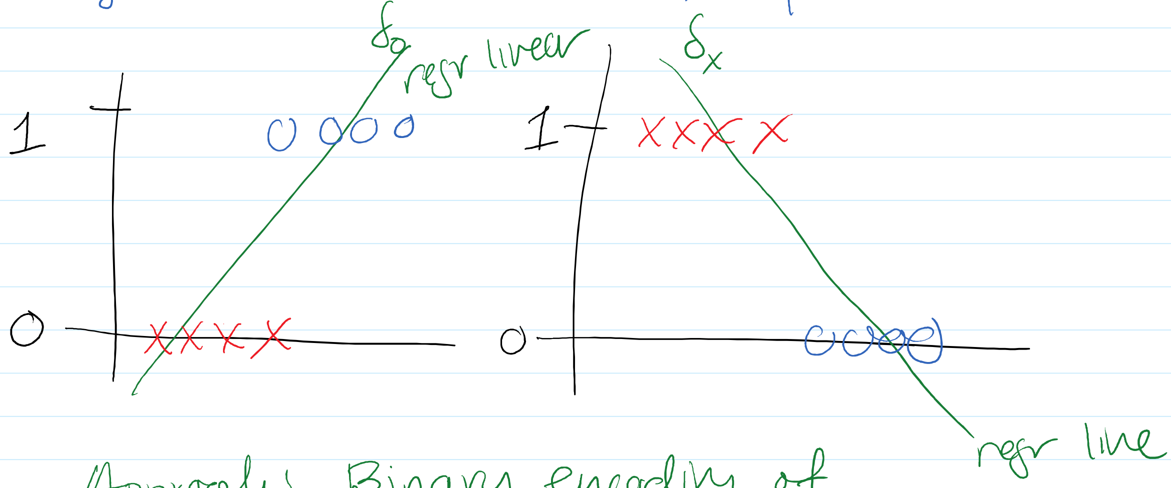
much more flex.

Linear classifiers: very simply is a linear classifier

$$\delta_c(x) = \beta_{0c} + \beta_c' x$$

LDA estimates $\hat{\beta}_{0c}, \hat{\beta}_c$ using training data and an assumption of Normality.

Why not use regression to find $\hat{\beta}$s?



<u>Approach:</u> Binary encoding of ys and fit regr line to this and use as discr. fns.

$$\delta_c(x) = \hat{\beta}_{0c} + \hat{\beta}_c^T x$$

obtained from LS regress.

<u>Punchline:</u> → this isn't horrible if # classes (K) is small

→ if $K=2$ (w/ some caveats) this will give me LDA.

→ when K is large we have <u>masking problems</u>

→ when K is large we have
masking problems