# Lecture 9 - Linear Classifiers

Recall the Bayes' classifier (optimal 0-1 loss classifier)

$$\hat{y} = \underset{c}{\arg\max}\ \mathbb{P}(Y=c \mid X=x)$$

two approaches:

① discriminative : model $Y|X$

ex. KNN

$\mathbb{P}(Y=c \mid X=x) \approx$ % of nearby training $y$s of class $c$

② generative: model $X|Y$ and $Y$

Bayes' rule: $\mathbb{P}(Y=c \mid X=x) \propto P(X=x \mid Y=c) \cdot P(Y=c)$

ex. linear discriminant analysis (today)

LDA

---

## Defn : Linear Classifier

Bayes' say $\hat{y} = \underset{c}{\arg\max}\ P(Y=c \mid X=x)$

More generally $\hat{y} = \underset{c}{\arg\max}\ \delta_c(x)$

discriminant function

More generally $y = {}_c \quad \delta_c(x)$

discrimina function

intuition: higher = better

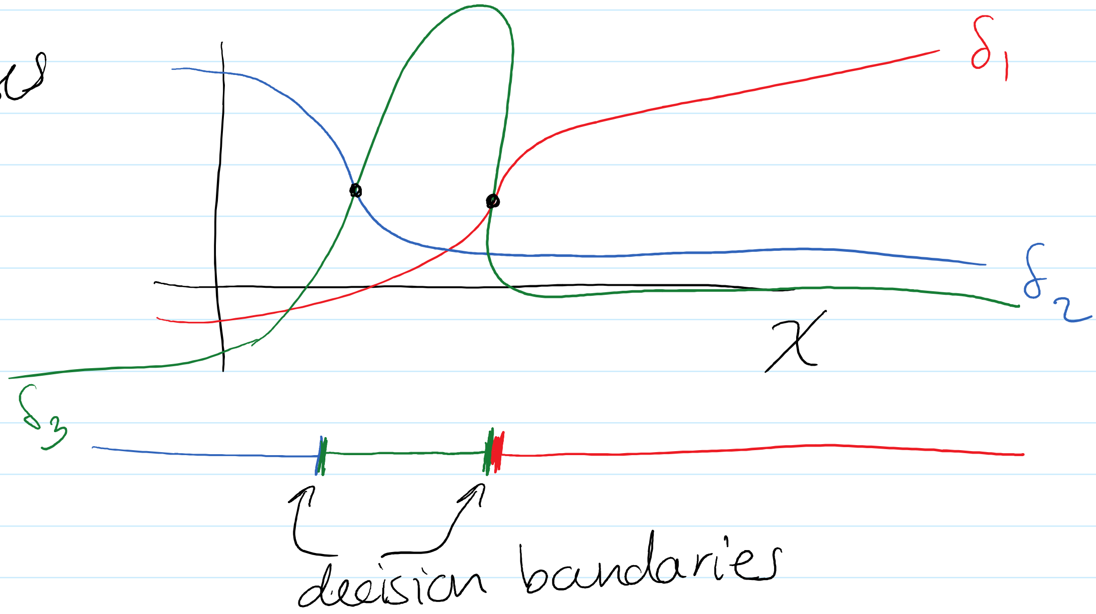
**Ex.** Bayes classifier is a particular example

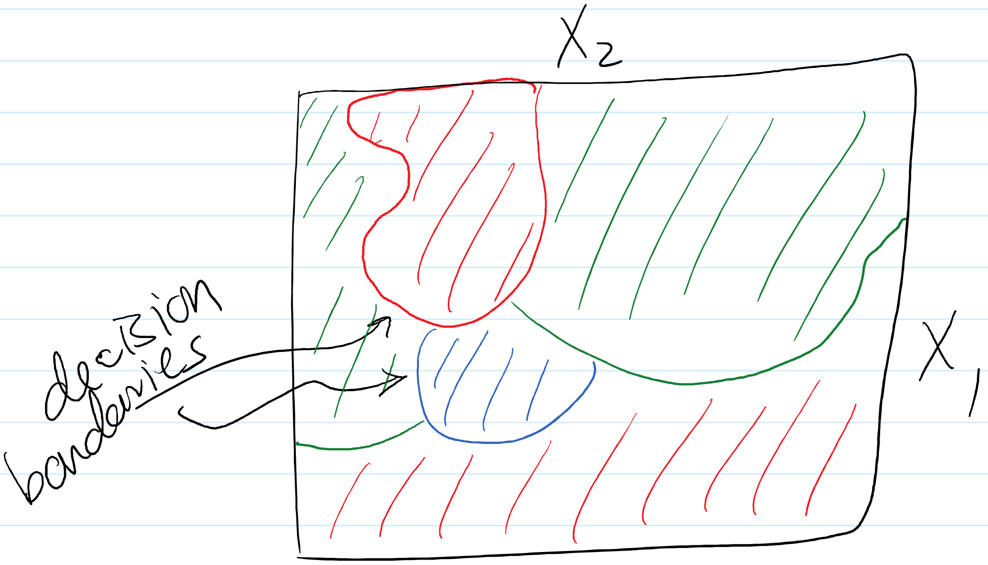
$$\delta_c(x) = P(Y=c \mid X=x)$$

**Ex.** $\delta_c(x) = \overset{\text{intercept}}{\beta_{0c}} + \overset{\text{slope}}{\beta_c^T} x$

**Ex.** $\delta_c(x) = \exp(\tanh(-\log(\alpha_c x^2)))$

3 classes





⋮

A **linear classifier** is a classifier whose discriminant functions can be transformed to linear functions (of $x$) by a increasing (monotone) transformation.

Ex. If the $\delta_c(x)$ is linear in $x$

i.e. $\delta_c(x) = \beta_{0c} + \beta_c^T x$

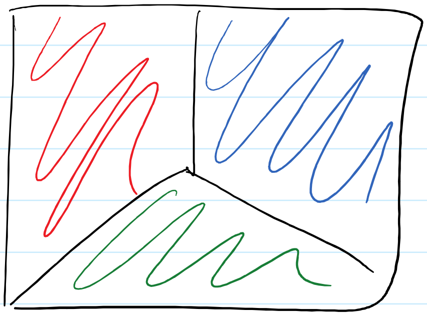
then the classifier is linear.

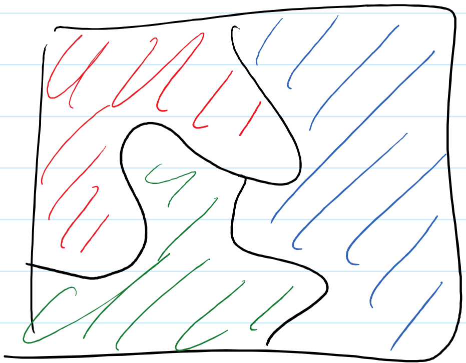
Ex. there is a monotone non-decreasing transformation $T$ so that

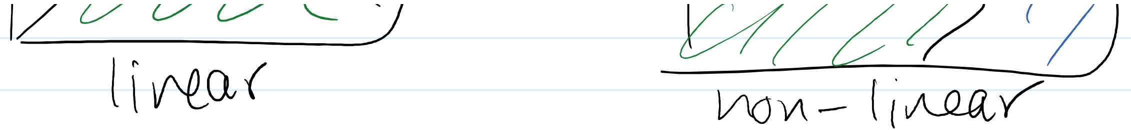
$$T(\delta_c(x)) = \beta_{0c} + \beta_c^T x$$

Ex. $T(x) = \log(x)$ or $T(x) = \exp(x)$

⊛ The decision boundaries of linear classifiers are linear. ⊛


linear


non-linear



---

Two classes: $\{x \mid \delta_1(x) = \delta_2(x)\}$

← decision boundary

Condition:

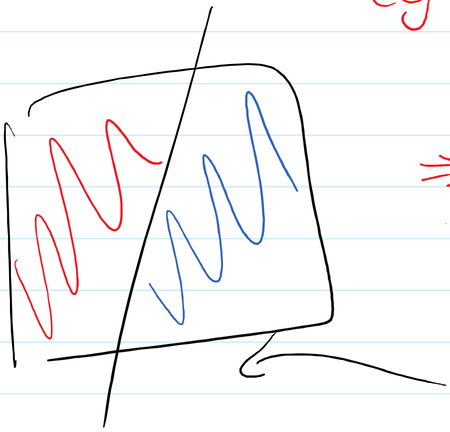
$$\delta_1(x) = \delta_2(x)$$

← if $\delta_1$ and $\delta_2$ are linear in $x$ then so is the solution set

Eg. $\overbrace{\beta_{01} + \beta_1^T x}^{\delta_1(x)} = \overbrace{\beta_{02} + \beta_2^T x}^{\delta_2(x)}$

$$\Rightarrow \underbrace{(\beta_{01} - \beta_{02}) = (\beta_2 - \beta_1)^T x}_{\text{linear algebra}}$$

says solution set is a subspace



---

Notice if $T$ is increasing then

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

← adding $T$ gives EXACT same classifier → decision boundaries are the same

Ex.

| $T(P(Y=1\|X))$ | $T(P(Y=2\|X))$ | $T(P(Y=3\|X))$ |
|---|---|---|
| $.5^2$ | $.3^2$ | $.2^2$ |

$T(x) = x^2$

$T(x)=x^2$ $\quad .5 \;|\; .3 \;|\; .2$ $\quad\rightarrow$

are the same

$\uparrow \hat{y}=1$

So if $\exists\, T$ that makes $\delta_c$s linear then

$$\hat{y} = \arg\max_c \delta_c(x) = \arg\max_c \underbrace{T(\delta_c(x))}_{\text{linear}}$$

another way ex

decision boundary defined $T(\delta_1(x)) = T(\delta_2(x))$

---

## Linear Discriminant Analysis (LDA)

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

Need: model $X|Y$ and $Y$

$\uparrow$ denom doesn't depend on c

$$\text{LDA}: \begin{cases} X|Y & \text{as Gaussian (Normal)} \\ Y & \text{as discrete} \end{cases}$$

$\arg\max_y f(y) = \arg\max_y k \cdot f(y)$

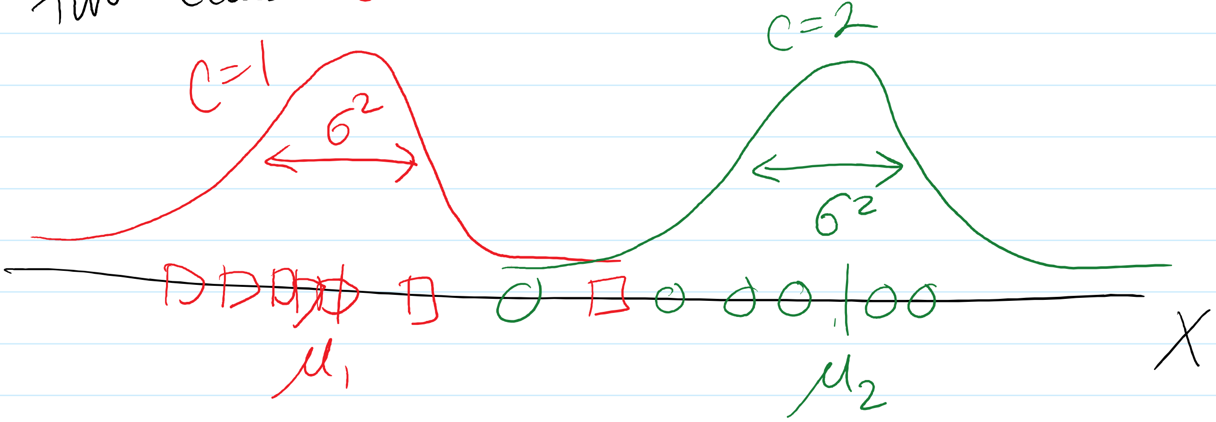
---

Ex Let $x \in \mathbb{R}$ i.e. $p=1$

LDA: $X|Y=c \sim N(\mu_c, \sigma^2)$

$$\underbrace{P(Y=c)=\pi_c \text{ where } \sum_c \pi_c = 1, \pi_c \geq 0}$$

Ex. two-class $c=1$ or $c=2$



$\boxed{P=1}$

$$X|Y=c \sim N(\underset{\sim}{\mu_c}, \underset{\sim}{\sigma^2})$$

$$P(Y=c)=\underset{\sim}{\pi_c} \leftarrow$$

learn them (estimate) from training data

$$\delta_c(x) = P(Y=c|X=x)$$

$$\downarrow$$

really use

$$\delta_c(x) \leftarrow P(X=x|Y=c)P(Y=c)$$

$$= \underbrace{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(x-\mu_c)^2\right)}\underbrace{\pi_c} \leftarrow$$

$$\delta_c(x) \leftarrow \log(\underset{\sim}{\quad\quad})$$

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

$$= \underbrace{-\frac{1}{2}\log(2\pi\sigma^2)}_{\text{ignore b/c doesn't depend on } c} - \frac{1}{2\sigma^2}(x-\mu_c)^2 + \log\pi_c$$

$$\delta_c(x) \longleftarrow -\frac{1}{2\sigma^2}(\underbrace{x^2}_{\text{ignore}} - 2x\mu_c + \mu_c^2) + \log\pi_c$$

$$\boxed{\delta_c(x) = -\frac{\hat{\mu}_c^2}{2\hat{\sigma}^2} + \frac{x\hat{\mu}_c}{\hat{\sigma}^2} + \log\hat{\pi}_c}$$

reality: need to estimate $\mu_c$s, $\sigma^2$, $\pi_c$s

How: $\hat{\mu}_c$ = mean of training $x_n$s in class $c$

$\hat{\sigma}^2$ = pooled variance

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

---

Notice: For LDA:

$$\delta_c(x) = \overbrace{\boxed{-\frac{\hat{\mu}_c^2}{2\hat{\sigma}^2}}}^{\hat{\beta}_{0c}} + \underbrace{\boxed{\frac{\hat{\mu}_c}{\hat{\sigma}_2}}}_{\hat{\beta}_c} x \overbrace{\boxed{+\log\hat{\pi}_c}}^{\hat{\beta}_{0c}}$$

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

So LDA is linear.