Lecture 9: Classification

$$f^*(\underset{\sim}{x}) = \underset{c}{\operatorname{argmax}}\ \mathbb{P}(Y=c \mid \underset{\sim}{X}=\underset{\sim}{x})$$

One way to build classifier is approx these probs, and then

$$\hat{f}(\underset{\sim}{x}) = \underset{c}{\operatorname{argmax}}\ \hat{\mathbb{P}}(Y=c \mid \underset{\sim}{X}=\underset{\sim}{x})$$

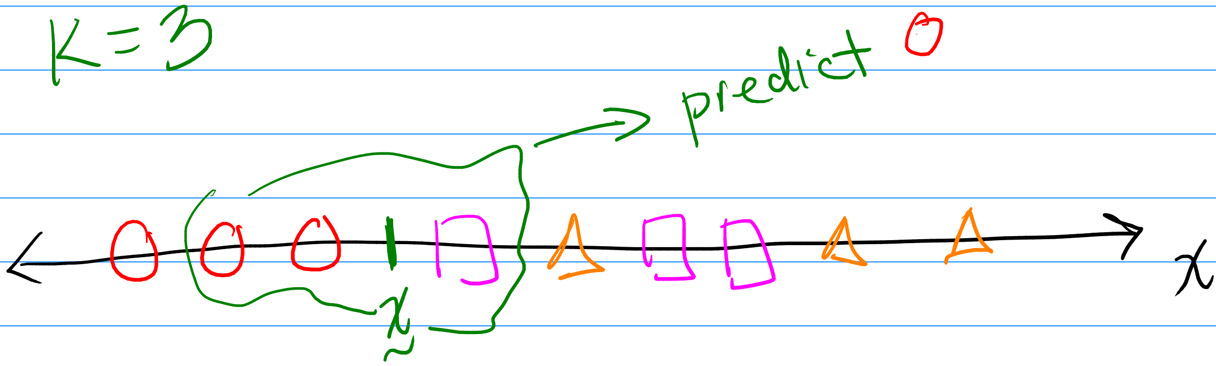
estimate

Simple way: KNN classification

$$\hat{\mathbb{P}}(Y=c \mid \underset{\sim}{X}=\underset{\sim}{x}) \approx$$ % of training y_ns in class c w/ $\underset{\sim}{x}_n$s near $\underset{\sim}{x}$

$$= \frac{1}{K} \underbrace{\sum_{\underset{\sim}{x}_n \in N_k(\underset{\sim}{x})} \mathbb{1}(y_n=c)}_{\#\text{ pts in class } c}$$

$K=3$



$$\underset{\sim}{x}_1 = (x_{11}, 10x_{12})$$

$$\underset{\sim}{x}_2 = (x_{21}, 10x_{22})$$

$$\|\underset{\sim}{x}_1 - \underset{\sim}{x}_2\| = \sqrt{(x_{11}-x_{21})^2 + 100(x_{12}-x_{22})^2}$$

$$\approx \sqrt{(x_{12}-x_{22})^2}$$

Correct Re-scaling

$$\hat{\mu} = \text{mean(train)}$$

$$\hat{\sigma} = \text{sd(train)}$$

$$z_{train} = \frac{x_{train} - \hat{\mu}}{\hat{\sigma}}$$

$$z_{test} = \frac{x_{test} - \hat{\mu}}{\hat{\sigma}}$$

Bayes' classifier says look at $\mathbb{P}(Y = c \mid \underset{\sim}{X} = \underset{\sim}{x})$

(1) discriminative models

→ model $\mathbb{P}(Y = c \mid \underset{\sim}{X} = \underset{\sim}{x})$ directly

i.e model $Y \mid \underset{\sim}{x} = \underset{\sim}{x}$

↳ KNN, logistic regression, classification trees

② generative models

Bayes' rule: $\mathbb{P}(Y=c \mid \underset{\sim}{X}=\underset{\sim}{x}) = \dfrac{\mathbb{P}(\underset{\sim}{X}=x \mid Y=c)\,\mathbb{P}(Y=c)}{\mathbb{P}(\underset{\sim}{X}=\underset{\sim}{x})}$

$\propto \mathbb{P}(X=\underset{\sim}{x} \mid Y=c)\,\mathbb{P}(Y=c)$

model $\underset{\sim}{X} \mid Y=c$ and Y

$\hookrightarrow$ LDA/QDA, naive Bayes'

Linear Classifiers

Bayes' says $\hat{f}(\underset{\sim}{x}) = \underset{c}{\operatorname{argmax}}\ \mathbb{P}(Y=c \mid \underset{\sim}{X}=\underset{\sim}{x})$

more generally

$$\hat{f}(\underset{\sim}{x}) = \underset{c}{\operatorname{argmax}}\ \delta_c(\underset{\sim}{x})$$

discriminant functions

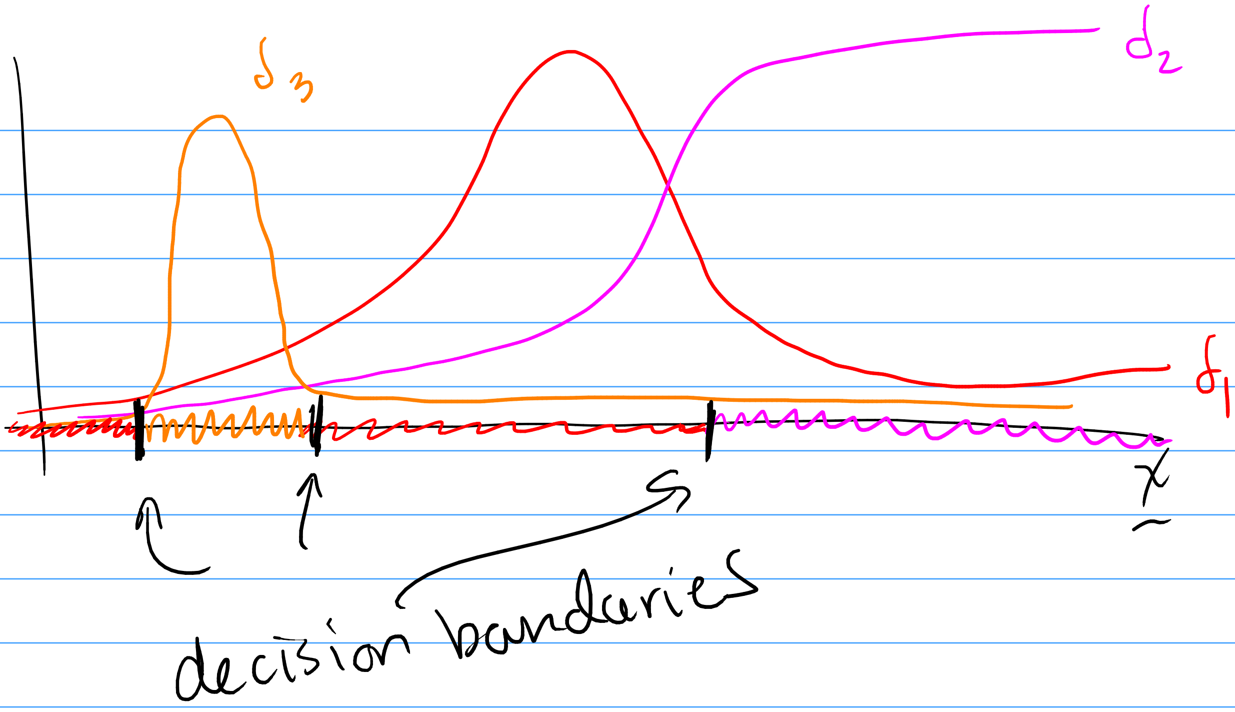
$\delta_c(\underset{\sim}{x})$ = large if $\hat{f}(\underset{\sim}{x})$ likely in class c, small otherwise

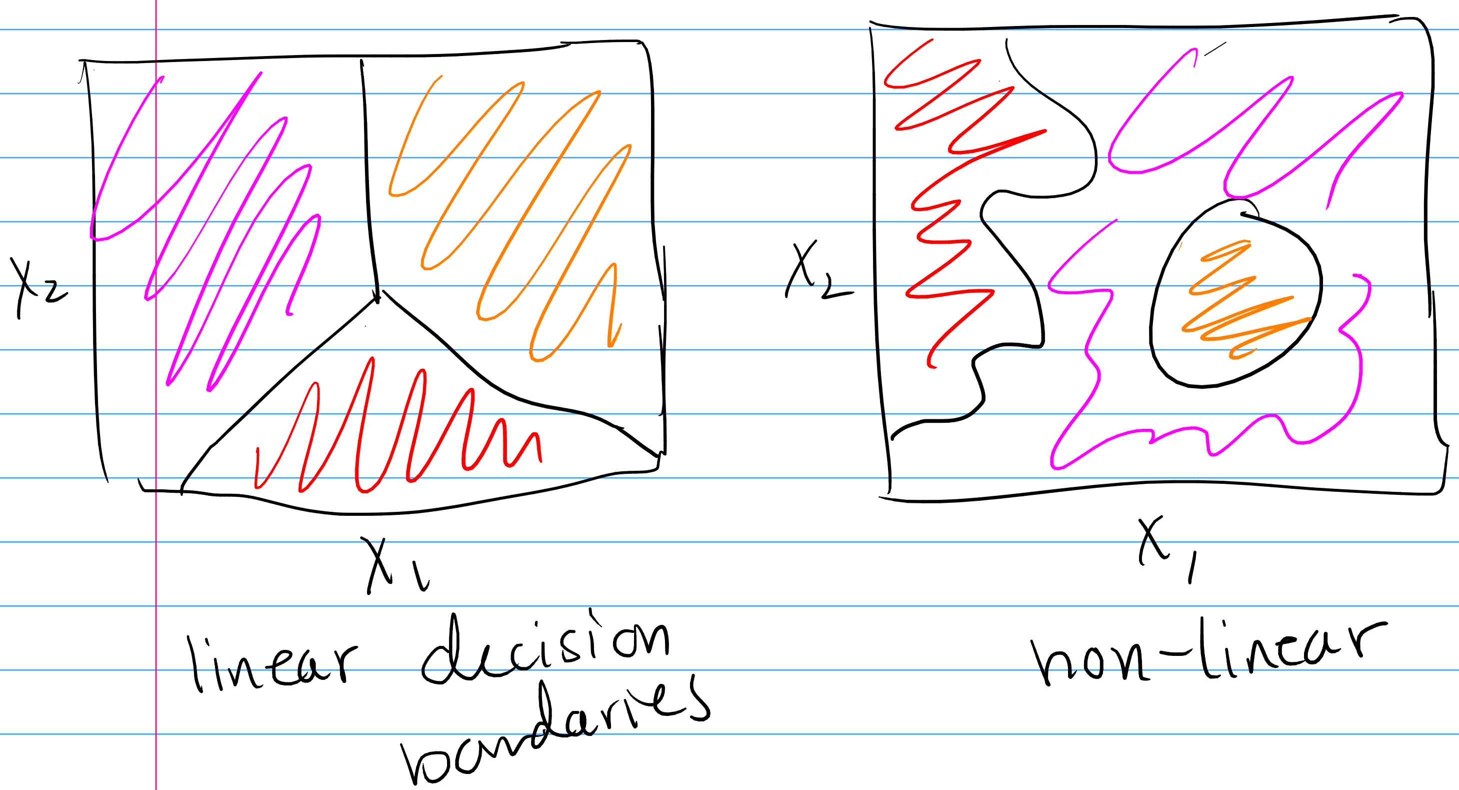
Ex. $\delta_c(\underset{\sim}{x}) = \hat{\mathbb{P}}(Y=c \mid \underset{\sim}{X}=\underset{\sim}{x})$

Ex. $\delta_c(\underset{\sim}{x}) = \log \hat{\mathbb{P}}(Y=c \mid \underset{\sim}{X}=\underset{\sim}{x})$

any increasing fn

Ex. $\delta_c(\underset{\sim}{x}) = \alpha_c + \beta_c^T \underset{\sim}{x}$





<u>Defn</u>: <u>Linear classifers</u> have linear decision boundaries

<u>Defn</u>: The discriminant functions δ_c of a <u>linear classifier</u> can be transformed to linear functions via increasing transformations (vice-versa)

Ex. $\delta_c(\underset{\sim}{x}) = \alpha_c + \beta_c^T \underset{\sim}{x}$

gives a linear classifier

Ex. $\delta_c(\underset{\sim}{x}) = e^{\alpha_c + \beta_c^T \underset{\sim}{x}}$

is also linear

Reason:

$$\hat{f}(\underset{\sim}{x}) = \underset{c}{\operatorname{argmax}}\, \delta_c(\underset{\sim}{x})$$

can depend on $\underset{\sim}{x}$ but not c

$$= \underset{c}{\operatorname{argmax}}\, T(\delta_c(\underset{\sim}{x}))$$

T increasing

Why do linear classifiers have linear decision boundaries?

For two class problem $Y = 1$ or 2

decision boundary: $\{ \underset{\sim}{x} : \delta_1(\underset{\sim}{x}) = \delta_2(\underset{\sim}{x}) \}$

$\delta_c(\underset{\sim}{x}) = \alpha_c + \beta_c^T \underset{\sim}{x}$

then $\delta_1(\underset{\sim}{x}) = \alpha_1 + \beta_1^T \underset{\sim}{x} = \alpha_2 + \beta_2^T \underset{\sim}{x} = \delta_2(\underset{\sim}{x})$

$$\Rightarrow (\alpha_1 - \alpha_2) + (\beta_1 - \beta_2)^T \underset{\sim}{x} = 0$$

linear system

linearizable by T

Generally,

$$\delta_1(\underline{x}) = \delta_2(\underline{x})$$

$$T(\delta_1(\underline{x})) = T(\delta_2(\underline{x}))$$

linear system ...

LDA: Linear Discriminant Analysis

Model $\underset{\sim}{X} \mid Y = c$ and Y

$$\delta_c(\underset{\sim}{x}) \equiv \mathbb{P}(Y = c \mid \underset{\sim}{X} = \underset{\sim}{x}) \equiv \mathbb{P}(\underline{X} = \underline{x} \mid Y = c) \cdot \mathbb{P}(Y = c)$$

Up to some inc. transf

$\boxed{p = 1}$

deps on c

LDA: $\begin{cases} \underset{\sim}{X} \mid Y = c \sim N(\mu_c, \sigma^2) \\ Y \text{ is discrete} \end{cases}$

shared across c

Y	1	2	....	K
$\mathbb{P}(Y=c)$	π_1	π_2	...	π_K

$$\pi_c \geq 0 \quad \text{and} \quad \sum_c \pi_c = 1$$

Ex, $K=2$ and $\underset{\sim}{X}$ is 1-dim'l

Learning LDA: reduces to learning

$\pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \sigma^2$