Classification

Thursday, September 19, 2024 3:29 PM

Outline:

- Stat. Learning
  - supervised
    - regr.
    - $\boxed{\text{classification}}$
  - unsupervised

Classification

$\underset{\sim}{x} \in \mathbb{R}^p, \; y \in \mathcal{C}$

$$\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$$

possible classes

Goal: build $\hat{f}$ so that $y \approx \hat{f}(\underset{\sim}{x})$

Loss function for class. 0-1 loss

$$L(y, \hat{f}(x)) = \mathbb{1}(y \neq \hat{f}(x))$$

$$L(y, \hat{f}(\underset{\sim}{x})) = \mathbb{1}(y \neq \hat{f}(\underset{\sim}{x}))$$

$$= \begin{cases} 1 & y \neq \hat{f}(\underset{\sim}{x}) \\ 0 & y = \hat{f}(\underset{\sim}{x}) \end{cases}$$

What is f^*?

$$f^*(\underset{\sim}{x}) = \underset{c}{\operatorname{argmin}}\, E[L(Y, c) \mid \underset{\sim}{x}]$$

$$= \underset{c}{\operatorname{argmin}}\, E[\mathbb{1}(Y \neq c) \mid \underset{\sim}{x}]$$

<u>Claim</u>: $E[\mathbb{1}(\text{statement})] = P(\text{statement})$

$$= \underset{c}{\operatorname{argmin}}\, P(Y \neq c \mid \underset{\sim}{x})$$

$$= \underset{c}{\operatorname{argmin}}\, 1 - P(Y = c \mid \underset{\sim}{x})$$

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

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

Bayes' classifier

Ex. 3-class : Cats, Dogs, People

$\underset{\sim}{X}$ = image

Calc: $P(Y=\text{cat} \mid \underset{\sim}{x})$, $P(Y=\text{dog} \mid \underset{\sim}{x})$, $P(Y=\text{people} \mid \underset{\sim}{x})$

Pick class w/ largest prob.

Reality: don't know joint of $\underset{\sim}{X}$, Y, but can try to est. $P(Y=c \mid \underset{\sim}{x})$.

Simple Way:

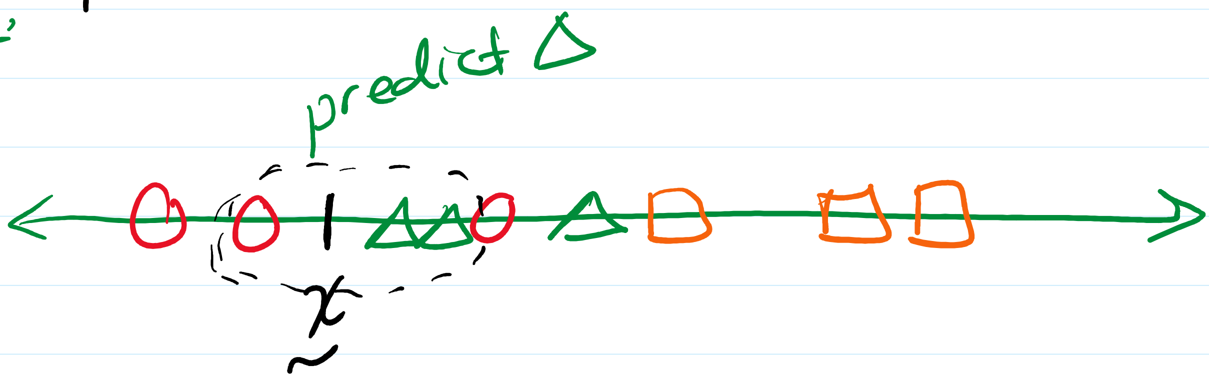
$\hat{P}(Y=c \mid \underset{\sim}{x})$ = % of training y_ns in class c w/ corresp. $\underset{\sim}{x}_n$s near $\underset{\sim}{x}$.

w/ corresp. $\underset{\sim}{x}_n$s near $\underset{\sim}{x}$.

$$= \frac{1}{K} \sum_{\underset{\sim}{x}_n \in N_K(\underset{\sim}{x})} \mathbb{1}(y_n = c)$$

$$\hat{f}(\underset{\sim}{x}) = \arg\max_c \hat{P}(Y = c \mid \underset{\sim}{X})$$

Ex, $K = 3$



Correct Scale:

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

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

$$Z_{train} = \frac{X_{train} - \hat{\mu}}{\hat{\sigma}}$$

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

Bayes' Classifier says to look at

$$P(Y = c \mid \underset{\sim}{x})$$

① <u>discriminative models</u>

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

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

e.g. KNN, logistic regression, classification trees

② <u>generative model:</u>

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

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

$$= \frac{[\text{illegible}]}{P(\underset{\sim}{X} = \underset{\sim}{x})}$$

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

model both $\underset{\sim}{x} \mid Y=c$ and Y.

e.g. LDA/QDA, naive bayes'

Linear Classifiers

Bayes' classifier:

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

more generally:

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

(discriminat fns

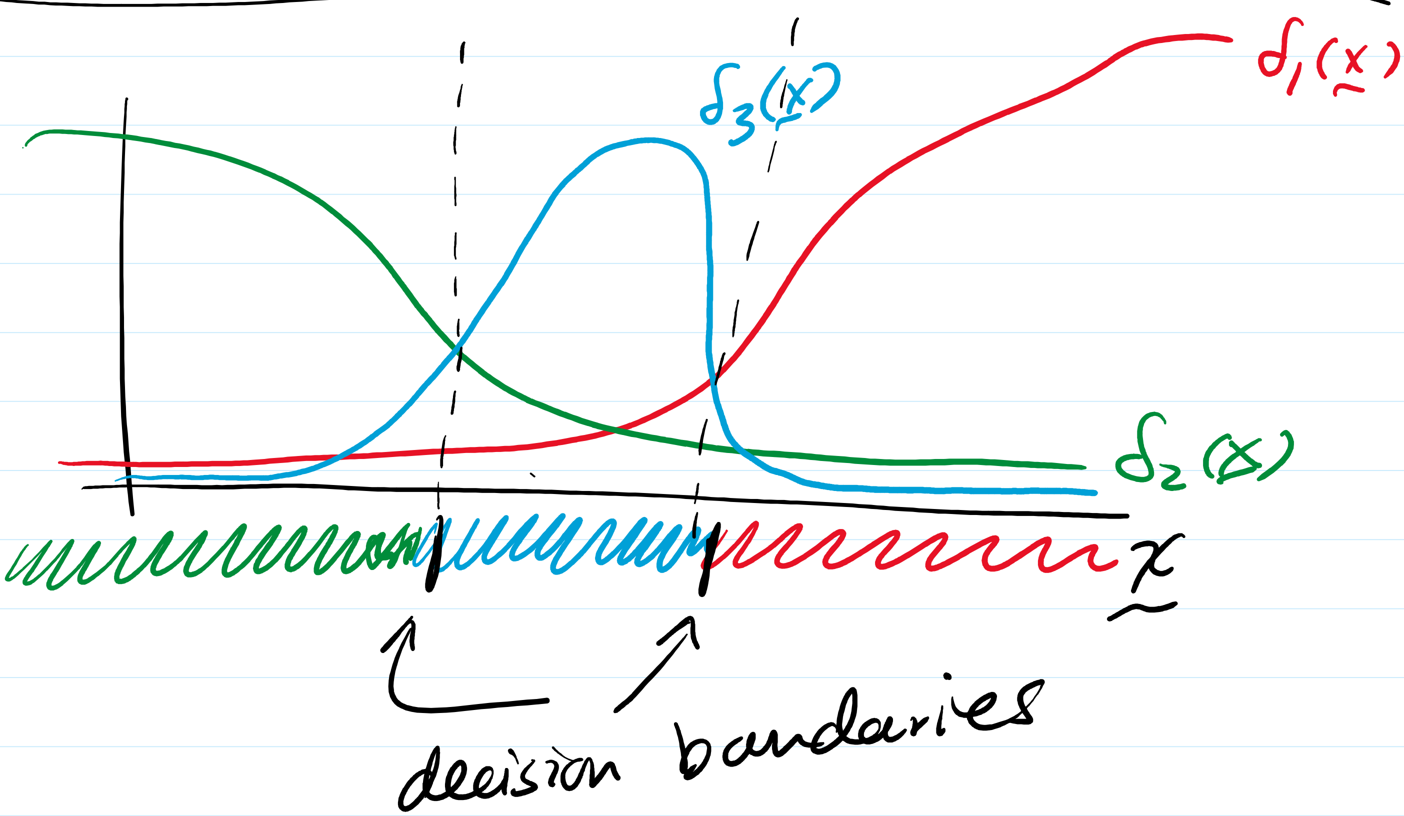
Ex, $\delta_c(\underset{\sim}{x}) = \hat{P}(Y=c \mid \underset{\sim}{X})$

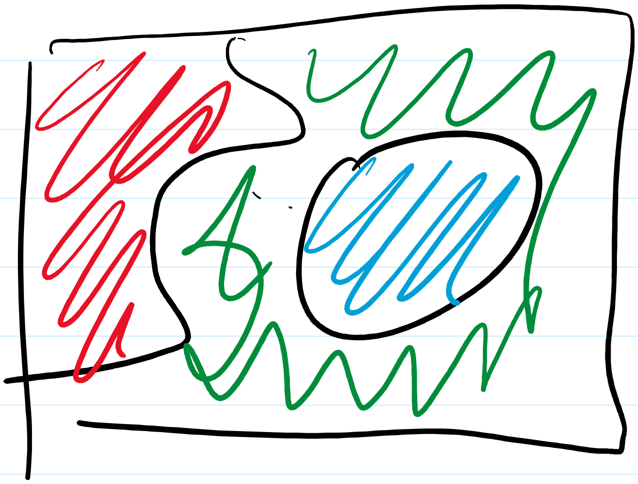
Ex, $\delta_c(\underset{\sim}{x}) = \alpha_c + \beta_c^T \underset{\sim}{x}$

idea: $\delta_c(\underset{\sim}{x})$ large if $\underset{\sim}{x}$ corresp. to class c.

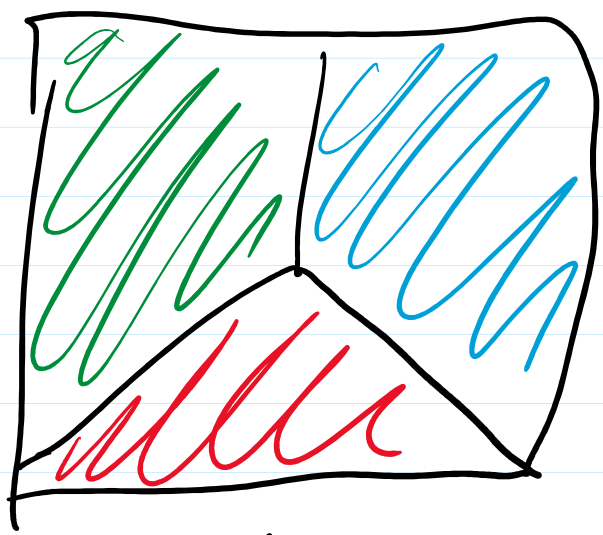
$\underline{Ex}.$ $\delta_c(\underset{\sim}{x}) = \alpha_c + \beta_c^T \underset{\sim}{x}$

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





non-lin.
dec. boundaries



lin. decision
boundaries

dec. bandaries bandaries

Defn: A lin. class. has lin. dec. bandaries

Thrm: The disc. fns of a lin. class. can be transformed to lin. fns. by some inc. transf. (iff)

Reason:

Decision bandary for 2-class problem.

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

If δs are linear:

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

bandary is soln space of

boundary is soln space of

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

which is a lin. subspace.

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

inc. fn.

boundary:

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

has h^{-1}

get back orig. lin syst.

Another way:

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

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

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA)

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

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

up to some inc. transf.

shared

LDA

$$\left\{ \begin{array}{l} \underset{\sim}{X} \mid Y = c \sim N(\mu_c, \Sigma) \\ \\ Y \text{ is discrete} \end{array} \right.$$

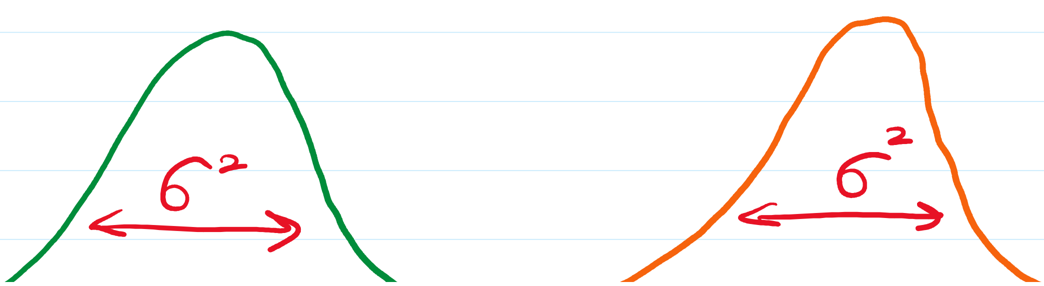
deps on class

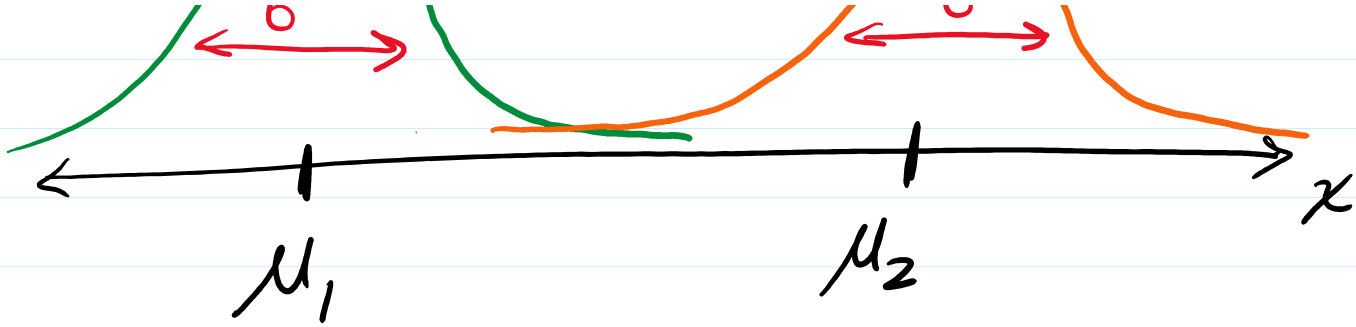
Y	1	2	...	K
$P(Y=c)$	π_1	π_2	...	π_K

$$\pi_i \geq 0, \quad \sum_{i=1}^{K} \pi_i = 1.$$

Ex, $K=2$

$\underset{\sim}{x}$ is 1-dim'l





To "learn" LDA I just need to learn the parameters:

$$\pi_1, \dots, \pi_K, \mu_1, \dots, \mu_K, \Sigma$$