Lecture 14

Thursday, October 19, 2023 10:02 AM

Bayes classifier

likelihood Prior

$$P(C_i|\vec{x}_j) = \frac{\boxed{P(\vec{x}_j|C_i)} \cdot \boxed{P(C_i)}}{P(\vec{x}_j)}$$

Full Bayes

↳ Parametric : Assume $P(\vec{x}_j|C_i)$ to normally distributed

$$P(\vec{x}_j|c_i) \sim N(\vec{x}_j \mid \underbrace{\hat{\vec{\mu}}_i, \hat{\Sigma}_i}_{\theta = \{\vec{\mu}_i, \Sigma_i\}_{i=1}^{k}})$$

$$N \propto e^{-(\vec{x}_j-\vec{\mu}_i)\Sigma_i^{-1}(\vec{x}_j-\vec{\mu}_i)/2}$$

Naive Bayes :

all attributes are independent

$$\Sigma_i = \begin{bmatrix} \sigma_{i1}^2 & & 0 \\ & \sigma_{i2}^2 & \\ & \ddots & \\ 0 & & \sigma_{id}^2 \end{bmatrix}$$

d = # of attributes

i = class

Non-parametric

$P(\vec{x}_j|c_i)$

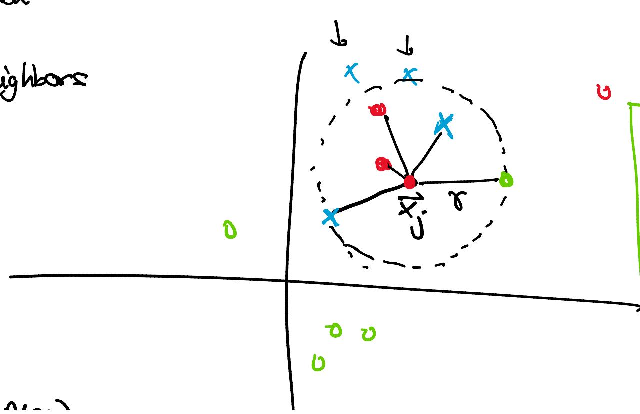
k = # of classes

K = # of nearest neighbors

$\underbrace{K\text{-NN}}$ approach

K - nearest neighbors

$K=5$



$k_1 = 2$

$k_2 = 2$

$k_3 = 1$

$K = 5$

# of pts from each

$n_1 = 3$

$n_2 = 4$

$n_3 = 5$

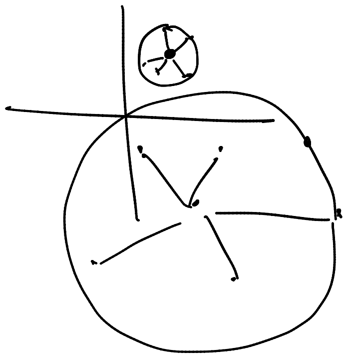
$P(c_i|\vec{x}_j) = \boxed{f(\vec{x}_j|c_i)} P(c_i)$

$K=5$

$$P(c_i|\vec{x}_j) \sim \frac{\boxed{f(\vec{x}_j|c_i)}\, P(c_i)}{P(\vec{x}_j)}$$

$$B_d(r) = \{ x_i \mid \|\vec{x}_i - \vec{x}_j\| \leq r \}$$

r: distance from $\vec{x}_j$ to its K-th NN



$$f(\vec{x}_j|c_i) = \frac{\text{fraction of points in } c_i}{\text{vol}(B_d(r))} \leftarrow$$

$=$

\# of points in $B_d(r) \equiv K$

$$\boxed{f(x_j|c_i) = \frac{k_i/n_i}{V}}$$

$$f(x_j|c_1) = \frac{2/3}{V}$$

$$f(x_j|c_3) = \frac{1/5}{V}$$

$$f(x_j|c_2) = \frac{2/4}{V}$$

$$P(c_i|\vec{x}_j) = \frac{f(x_j|c_i)\, P(c_i)}{\sum_{a=1}^{k} f(x_j|c_a) \cdot P(c_a)} = \frac{\boxed{\frac{k_i/n_i}{V}}\ \frac{n_i}{n}}{P(x_j)} = \frac{\frac{k_i}{\cancel{Vn}}}{\sum_{a=1}^{k} \frac{k_a}{\cancel{Vn}}}$$

$$\frac{k_i}{\cancel{n_i} V} \cdot \frac{\cancel{n_i}}{n} = \frac{k_i}{Vn}$$

$$P(c_i|x_j) = \frac{k_i}{\sum k_a} = \frac{k_i}{K}$$

Posterior probability $=$ fraction of points in class c_i within the ball $B_d(r)$

$$P(c_1|\vec{x}_j) = 2/5$$

K-NN classifier

$P(C_1 | \vec{x}_j) = 2/5$

$P(C_2 | \vec{x}_j) = 2/5$

$P(C_3 | \vec{x}_j) = 1/5$

K-NN classifier

K-NN classifier $(D, \vec{z}, K)$

data · any point

1) find the K nearest neighbors of $\vec{z}$

(Compute distance $\|\vec{z} - \vec{x}_i\| \ \forall \vec{x}_i \in D$

sort, pick the smallest K.)

worst case $\rightarrow \underbrace{O(n \cdot d)}$

$\underbrace{O(\underline{\log n} \cdot d)}$

Only true for small d

2) Count $K_i \ \forall \ i = 1 .. k$

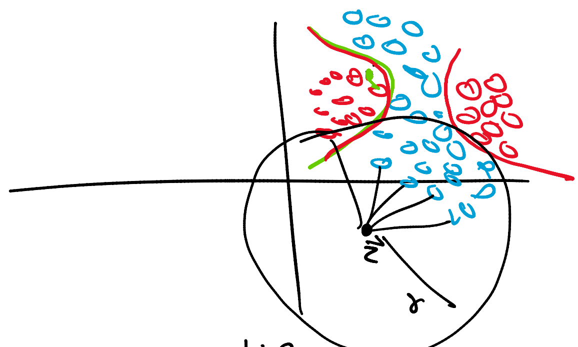
$\Downarrow$

\# of points in class C_i

3) $\hat{y} = \underset{i=1}{\overset{k}{\arg\max}} \{ K_i \}$

K-NN is actually <u>very powerful</u> ! non-linear

$\mathbb{R}^d$



$K = 1$

Small $K \leftarrow$ rough boundary "noisy"

large $K \leftarrow$ very smooth

Q1: what value of K?

Q2: what <u>norm</u>?

$L_2 = \|\vec{z} - \vec{x}_i\|^2 \ \forall \vec{x}_i \in D$

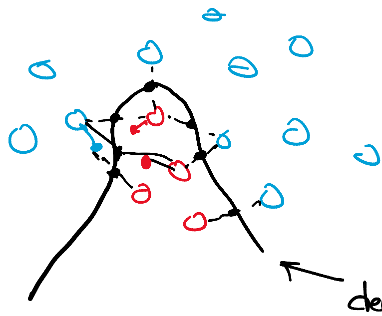
what distance metric should we use

L_p

$\cos\theta$

Q3: time!

each point requires $O(n.d)$ time to classify!



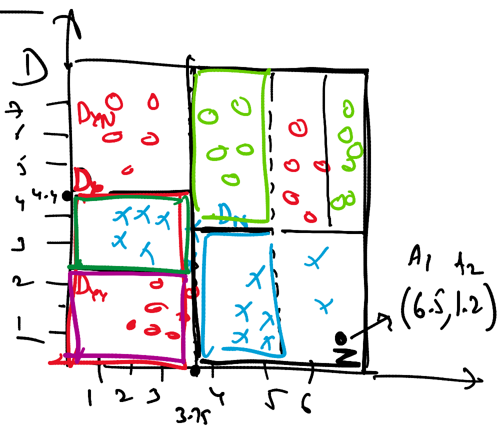
decision boundary!

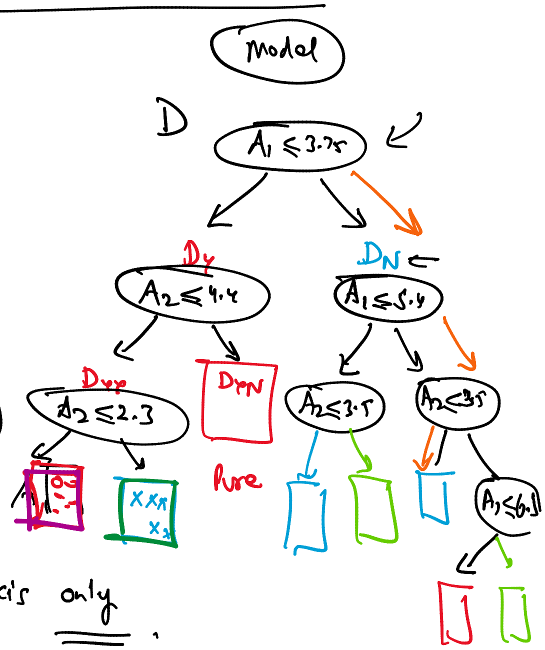
$K=1$

Decision Trees

⇓

recursively partitions into eventually "pure" regions





partition function / decision is along each axis only.

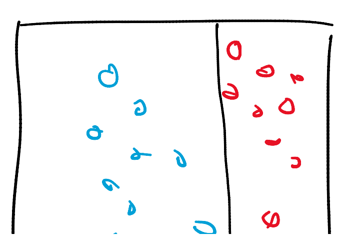
$\underbrace{\text{partition function / decision is along each axis only}}_{\text{axis-aligned hyperplane}}$

Choose the best hyperplane / decision at each step

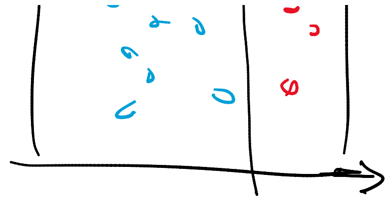
$\underbrace{\text{hyperplane / decision}}_{\text{greedy}}$

each axis acts like the 'normal' vector to the hyperplane

Z is classified as blue point!



'normal' vector to the hyperplane



---

axis aligned hyperplane

↳ all decisions are of the form

$A_i \leq v_j$

attribute value

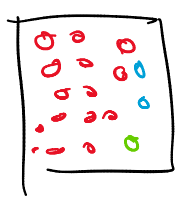
Yes No

decision

<u>at each step</u>: we have a "dataset" of interest

Decision Tree (D, stopping criteria = purity or size)
θ m

base
1) If $|D| \leq m$ stop (size)
2) If $\text{purity}(D) \geq \theta$ stop (majority class)



3) evalue all splits or decisions of the kind

$A_i \leq v_j$

$\forall i = 1 \ldots d$ (all attributes)

for all <u>mid-points</u> v_j

select the <u>best</u> (greedy)

(try all (attribute, value) combinations)

4) Use $A_i^* \leq v_j^*$ best split to partition D

$D_Y = \{ \vec{x}_a \mid x_{ai} \le v_j^* \}$

$D_N = \{ \vec{x}_a \mid x_{ai} > v_j^* \}$

5) Decision Tree (D_Y)

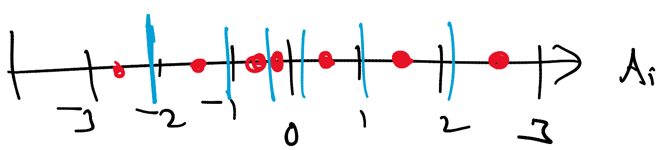
Decision Tree (D_N)

---

how to choose the best (A_i, v_j) combination

$A_i \le v_j$ decision.

D



1) find the distinct values along A_i

−2.7, −1.5, −0.8, −0.2, 0.5, 1.5, 2.6

v_j { −2.1, −1.15, −0.5, 0.15, 1, 2.05 }

try all of them

$A_i \le -2.1$

$A_i \le -1.15$

⋮

$A_i \le 2.05$

---

which of these is the best over all $A_i \le v_j$ D

Criteria: Information Gain D_Y D_N

Criteria: Information Gain $\quad D_Y \quad D_N$

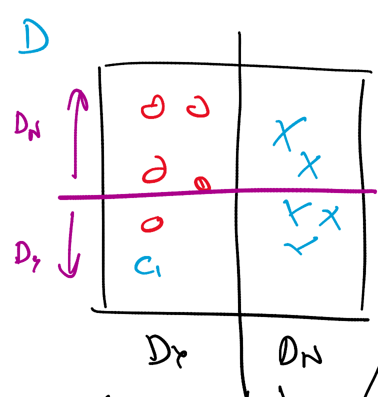
$$IG(D, D_Y, D_N) = \underbrace{H(D)}_{\text{original entropy}} - \underbrace{H(D_Y, D_N)}_{\text{joint entropy after the split}}$$

we want this value to be smaller

$$H(D) \equiv \text{entropy} = -\sum_{i=1}^{k} p_i \log p_i$$

p_i = prob of class c_i in D $= n_i/n = \dfrac{\#\text{ of points in } c_i}{\#\text{ of points in } D}$

$$H(D_Y, D_N) = \frac{n_Y}{n} H(D_Y) + \frac{n_N}{n} H(D_N)$$



c_1 = red
c_2 = blue

$P_1 = 5/10 = 0.5$

$P_2 = 5/10 = 0.5$

$-\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right)$

$-\log(0.5)$ worst possible entropy

$H(D)$

$c_1 = 5/5 = 1$

$c_2 = 0/5 = 0$

$H(D_Y) = -(1 \cdot \log 1 + 0 \cdot \log 0)$
$= 0$

$c_1 = 0/5 = 0$

$c_2 = 5/5 = 1$

$H(D_N) = 0$

$$\boxed{\begin{aligned} H(D_Y, D_N) &= \frac{5}{10} \cdot 0 + \frac{5}{10} \cdot 0 \\ &= 0 \end{aligned}}$$

$IG = -\log(0.5)$

higher

$$H(D_Y, D_N) = \frac{5}{10} \cdot 0 + \frac{5}{10} \cdot 0 = 0$$

higher

$$IG = \underline{H(D)} - \underline{H(D_Y, D_N)}$$

Gini Index:

$$\underline{G(D)} = 1 - \sum_{i=1}^{k} p_i^2$$

Gini index is 0 if we have "pure" partitions

worst case

If all $p_i = 1/k$

$$1 - \sum_{i=1}^{k} \left(1/k^2\right)$$

$$1 - k/k^2 = \frac{k-1}{k}$$

worst gini index

$$G(D) - G(D_Y, D_N)$$