Bayes Classifier

$$P(c_i \mid \vec{x}) = \frac{P(\vec{x} \mid c_i) \cdot P(c_i)}{\underbrace{\sum_{a=1}^{k} P(\vec{x} \mid c_a) \cdot P(c_a)}_{P(\vec{x})}}$$

test point → $\vec{x}$

Likelihood : $P(\vec{x} \mid c_i)$ ← maximum likelihood

Prior : $P(c_i)$

Lazy classifier

$$\hat{y} = \arg\max_{c_i} \{ P(c_i \mid \vec{x}) \}$$

D → $D_{training}$ → D_1^{train}, D_2^{train}, …, D_k^{train}

D → $D_{testing}$ → $\vec{x}$

Full Bayes	Naive Bayes
$\vec{\mu}_i$, Σ_i	diagonal Σ_i

$$\underbrace{\qquad\qquad}_{P(\vec{x} \mid c_i)}$$

$\vec{x} = (x_1, x_2, \ldots x_d)^T$

A_1 A_2 . . . A_d

K Classes: μ_1 σ_1^2 μ_2 σ_2^2 . . .

mean & variance for each class

Numeric:

$N(\vec{x} \mid \vec{\mu}_i, \vec{\Sigma}_i)$

PDF
Prob Density Function

$$\prod_{j=1}^{d} P(x_j \mid c_i)$$

$$\prod_{j=1}^{d} N(x_j \mid \mu_i^{(j)}, \sigma_i^{(j)2})$$

Categorical:

PMF: Probability Mass Function

per class

Color Make Class

$\vec{x}$

Make

F T

$P(\text{Color} = +)$

	Color	Make	Class
$\vec{x}_1$	R	Ford	Y
$\vec{x}_2$	B	Toyota	N
$\vec{x}_3$	G	T	Y
$\vec{x}_4$	R	T	Y
$\vec{x}_5$	B	F	N
			:

Color : $\{R, B, G, Y\}$ 3

Make : $\{\text{Ford}, \text{Toyota}\}$ 2

6

$A_1, A_2, \ldots A_d$

$\underbrace{2 \quad 2 \ldots \quad 2}_{2^d}$

$n << 2^d$

		Make F	Make T	$P(\text{Color} = x)$
Color	R	n_{RF} 1	n_{RT} 1	2
	B	1	1	2
	G	0	1	1
	Y	0	0	0
		2	3	$n=5$

row marginal

Contingency table

PMF : divide each entry by n.

$$\hat{P}(\vec{x} = (R, F)) = \frac{n_{RF}}{n} = \frac{1}{5} = 0.2$$

Laplace correction

	n_i		new n_i	p_i
R	2	+1	3	1/3
B	2	+1	3	1/3
G	1	+1	2	2/9
Y	0	+1	1	1/9

$$n = \underbrace{5 + 4}_{9}$$

$$P(Y \mid c_i) = 0$$

$$\boxed{P_R = \left(\frac{n_i + 1}{\underbrace{n}_{5} + \underbrace{\#\text{vals for Attribute}}_{4}} \right)^2}$$

Prior distribution on the color

α_R — $P(R) = 0.3$

α_B — $P(B) = 0.5$

α_G — $P(G) = 0.15$

α_Y — $P(Y) = 0.05$

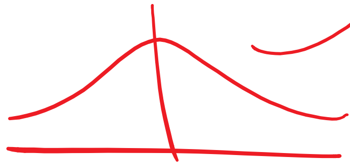
$\Sigma\alpha \quad 1$

Fixed Prior

$$P(x = R) = \frac{n_i + \alpha_R}{n + \underbrace{\Sigma\alpha}_{=1}}$$

$\Sigma \alpha$ 1 Prior

Prior distributions



Bayesian Modeling

Likelihood $P(\vec{x} \mid c_i)$ $\leftarrow$ Normal Distribution

Parametric Approach.

jointly (multivariate) Σ_i

naive/independent univariate σ_i^2 per attr

$\underbrace{\qquad\qquad\qquad}_{k \text{ classes}}$

---

non-parametric approach

$\rightarrow$ KNN classifier

k-nearest neighbor

$k \leftarrow$ input parameter

# of nearest neighbors

$$\boxed{P(c_i \mid \vec{x}) = \frac{p(\vec{x} \mid c_i) \cdot p(c_i)}{P(\vec{x})}}$$

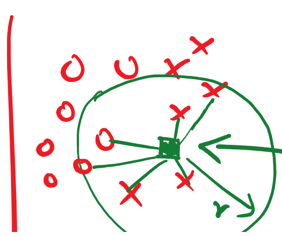
test

lazy learner (no model building required)

$K = 6$

$K_o = 2$

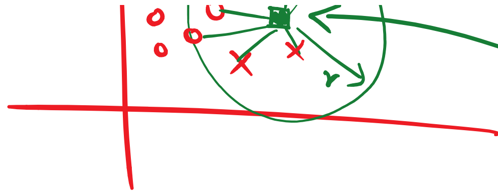
$K_x = 4$



$k = 6$

test

$K_o = 2$
$K_x = 4$



test case $\vec{x}$

$\hat{y} = \text{Cross}$

$\underbrace{B_r(\vec{x})}_{\text{d-dimensional hyperball}}$

$Vol(B_r(\vec{x})) = V$

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

estimated PDF
no assumption about the shape

n_i : # of points in class i in D_{training}

$P(c_i) = n_i/n$

k_i : # points in class c_i among the K nearest neighbors

$$P(c_i|\vec{x}) = \frac{f(\vec{x}|c_i) \cdot P(c_i)}{\sum_{a=1}^{k} f(\vec{x}|c_a) \cdot P(c_a)}$$

$$\left(\frac{k_i}{\cancel{n_i} \cdot V}\right) \cdot \frac{\cancel{n_i}}{n} = \frac{k_i}{nV}$$

$$= \frac{k_i/\cancel{nV}}{\sum_{a=1}^{k} \frac{k_a}{\cancel{nV}}}$$

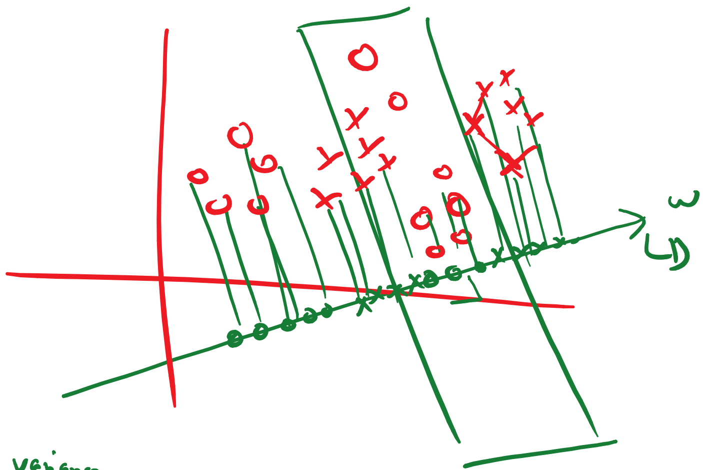
$$P(c_i|\vec{x}) = \frac{k_i}{\sum k_a} = \frac{k_i}{K}$$

KNN Classifier : $\arg\max P(c_i|\vec{x})$

$$= \arg\max \left\{\frac{k_i}{K}\right\}$$

$$\boxed{P(c_i|\vec{x}) = \arg\max_{c_i} \{k_i\}}$$

KNN ← easily handle non-linear decision boundaries



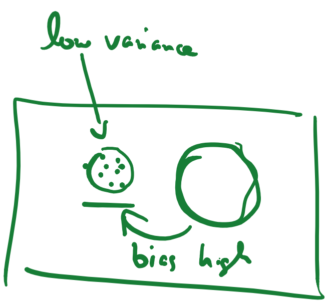
hyperplane

high bias

k = 1

high variance
low bias

Bias Variance Tradeoff



low bias
low variance

goal

Variance: perturb the training set
→ decision boundary change

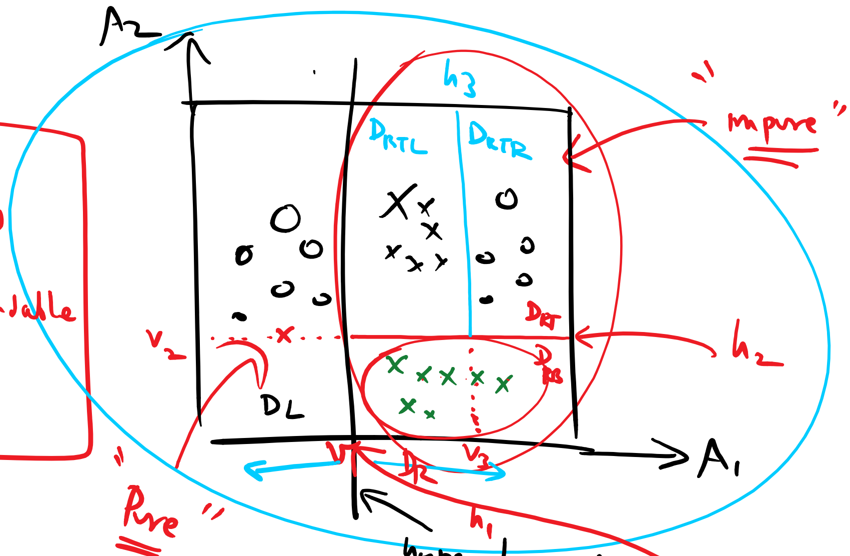
Bias: deviation from the true decision boundary

Decision Trees

↳ recursive linear decisions axis-parallel

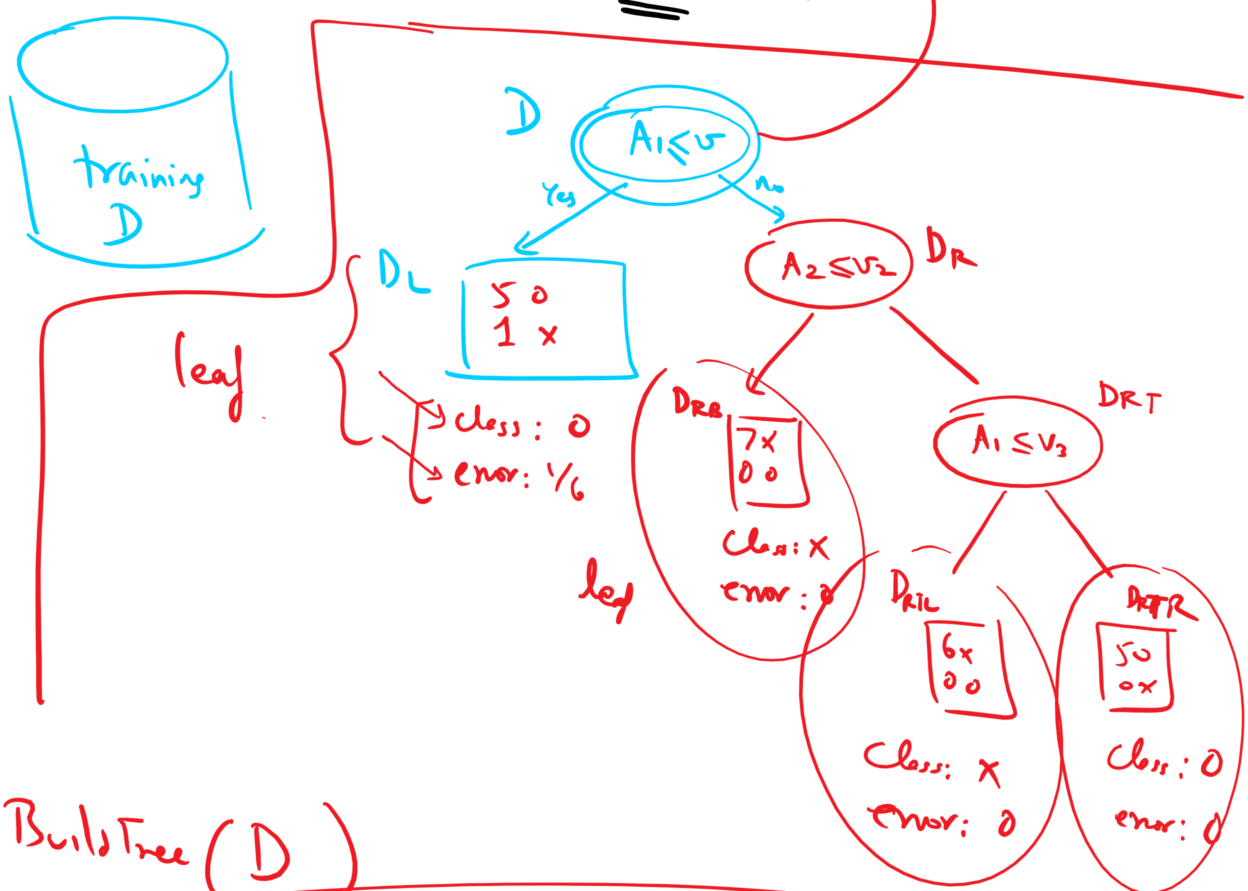
1) Simplicity

Understandable models



hyperplane 1.

"best" choice



BuildTree(D)

① try "all" possible axis parallel hyperplanes / splits
pick the best $A_i \le v_j$

② split Data into 2 groups

$D(\vec{x} \mid x_i \le v_j)$, $D(\vec{x} \mid x_i > v_j)$

② split Data into 2 groups

$D(\vec{x} \mid x_i \le v_j)$ $\qquad$ $D(\vec{x} \mid x_i > v_j)$

D_L $\qquad\qquad\qquad\qquad$ D_R

③ BuildTree (D_L)

BuildTree (D_R)

Stopping criteria

→ $|D| \le n_0$: Stop based on size

general models

→ purity $(D) \ge p_0$
$= 0.9$

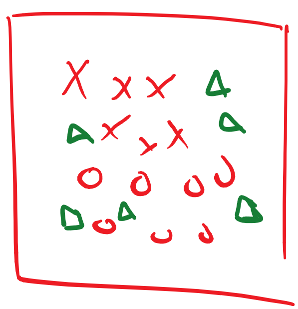
majority class

---

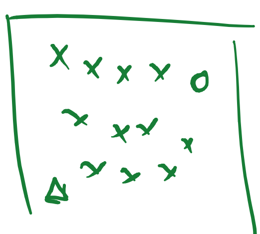
how to score a split?

→ Information Theory

→ Entropy of a region



high Entropy



low entropy

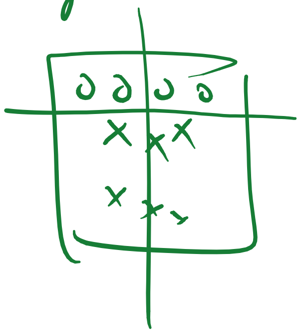
$$\boxed{H(D) = -\sum_{i=1}^{k} p_i \log(p_i)}$$

lowest value = 0 when there's only 1 class

$\boxed{\begin{array}{l} P_o : 1 \\ P_x : 0 \end{array}}$

highest value for a mixed case

before



Weighted Entropy of split

$$H(D_L, D_R) = \frac{n_L}{n} H(D_L) + \frac{n_R}{n} H(D_R)$$

Info Gain: $\underbrace{H(D) - H(D_L, D_R)}_{\text{maximize}}$