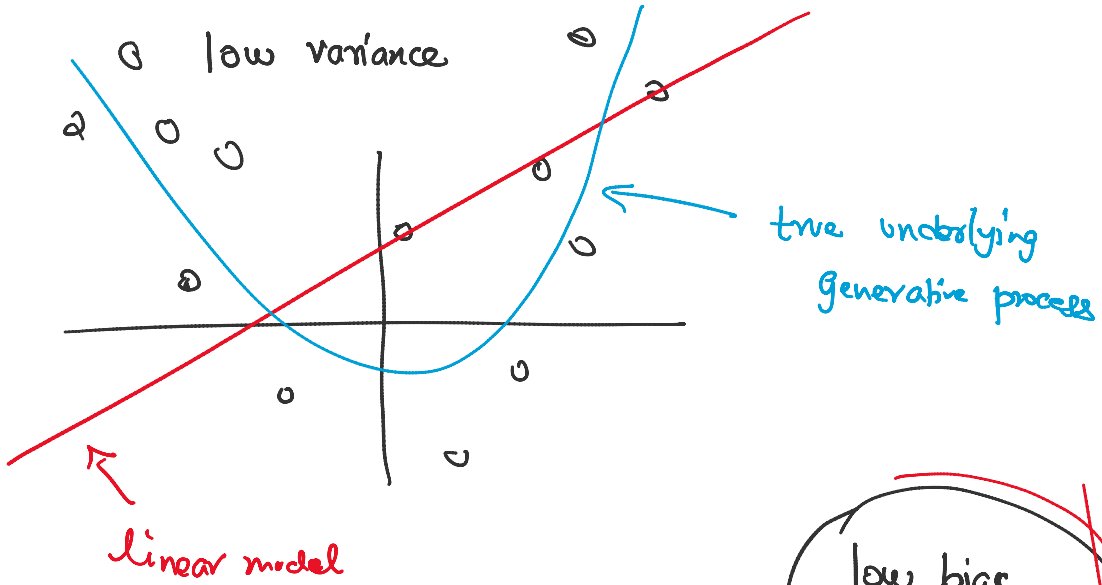
Lecture 18

Monday, November 6, 2023 10:02 AM

Simple models → high bias

e.g. just <u>linear</u>



low bias
low variance

<u>Complex Model</u>: 'fit anything'

low bias

high variance

$$L = (y - \hat{y})^2$$

↗ Squared error loss

y ← true target

$\hat{y}$ ← predicted value

goal: min expected loss (on test cases)

$$E_{\vec{x}}\left[(y-\hat{y})^2\right] = \boxed{\underbrace{E\left[(y-\mu_y)^2\right]}_{\text{noise term (variance of the target)}}}$$

expected loss

$(\vec{x}, y)$ ← given

$\hat{y} = M(\vec{x})$

(variance of the target)

$\mu_y = E[y]$

bias-variance tradeoff

$+ \; E\left[(\hat{y} - \mu_y)^2\right]$

average bias term

$+ \; E\left[(\hat{y} - \mu_{\hat{y}})^2\right]$

average variance term

$\mu_{\hat{y}} = E[\hat{y}]$

Ensemble methods

Bootstrap Aggregation (Bagging)

↳ average from multiple models

→ reducing variance

Bagging (D, Algo)

for $i = 1 \ldots k$

D_i = bootstrap sample (D)

M_i = Algo (D_i)

$\underbrace{M_1 \; M_2 \ldots M_k}_{k \text{ different models}}$

given any point $\vec{z}$

k # of samples

bootstrap sampling

Sample with replacement

$|D| = n$

2 classes
H L

$$\hat{y}_1 = M_1(\vec{z})$$
$$\hat{y}_2 = M_2(\vec{z})$$
$$\vdots$$
$$\hat{y}_k = M_k(\vec{z})$$

<u>classification:</u> majority vote $\{\hat{y}_1, \hat{y}_2, \ldots \hat{y}_k\}$

$$\underbrace{H \quad H \quad L \quad L \quad L}_{L}$$

regression : $E[\hat{y}] = \frac{1}{k}\left(\hat{y}_1 + \hat{y}_2 + \ldots + \hat{y}_k\right)$

<u>reduce variance</u> : better applied to a more complex model (low bias)

will help lower variance

Random Forests:

Ensemble of decision trees on bootstrap samples

+

sample the attributes to split on at every <u>node</u>

$(A_1, A_2 \ldots A_d)$

$\downarrow$

sample $m \ll d$ attributes

$A_5 \; A_{20} \ldots$

Boosting

$\hookrightarrow$ reduce bias via ensemble

↳ generally effective for simple model that have <u>high bias</u>

Adaptive Boosting (Adaboost)

↳ <u>biased</u> sampling

↳ pick points where previous round makes mistakes

Adaboost (D, Algo, K)

$\vec{\omega} = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)$ prob vector

ω_i = prob that point $\vec{x}_i$ will be sampled

for $i = 1 \ldots k$

D_i = sample D according to $\vec{\omega}$ of size n

$M_i = \text{Algo}(D_i)$

ε_i = error on D $\qquad \varepsilon_i \equiv$ error rate $= \frac{\#\text{ of mistakes}}{n}$

$\alpha_i = \ln\left(\frac{1}{\varepsilon_i} - 1\right) \equiv$ score of classifier M_i

<u>reweighting the points</u>

$\forall j = 1 \ldots n$

if M_i makes no mistake on $\vec{x}_j$

then ω_j stays the same

else: M_i makes a mistake

$$\omega_j = \omega_j \cdot \exp(\alpha_i)$$

$$= \omega_j \times \exp\left(\ln\left(\frac{1}{\varepsilon_i} - 1\right)\right)$$

$$= \omega_j \cdot \left(\frac{1}{\varepsilon} - 1\right)$$

$\varepsilon_i \leq 0.5$

$\ln\left(\frac{1}{0.5} - 1\right) = \ln(2-1) = \ln(1) = 0$

$\varepsilon_i \leq 0.25 = 1/4$

$\ln\left(\frac{1}{1/4} - 1\right) = \ln(4-1) = \ln(3)$

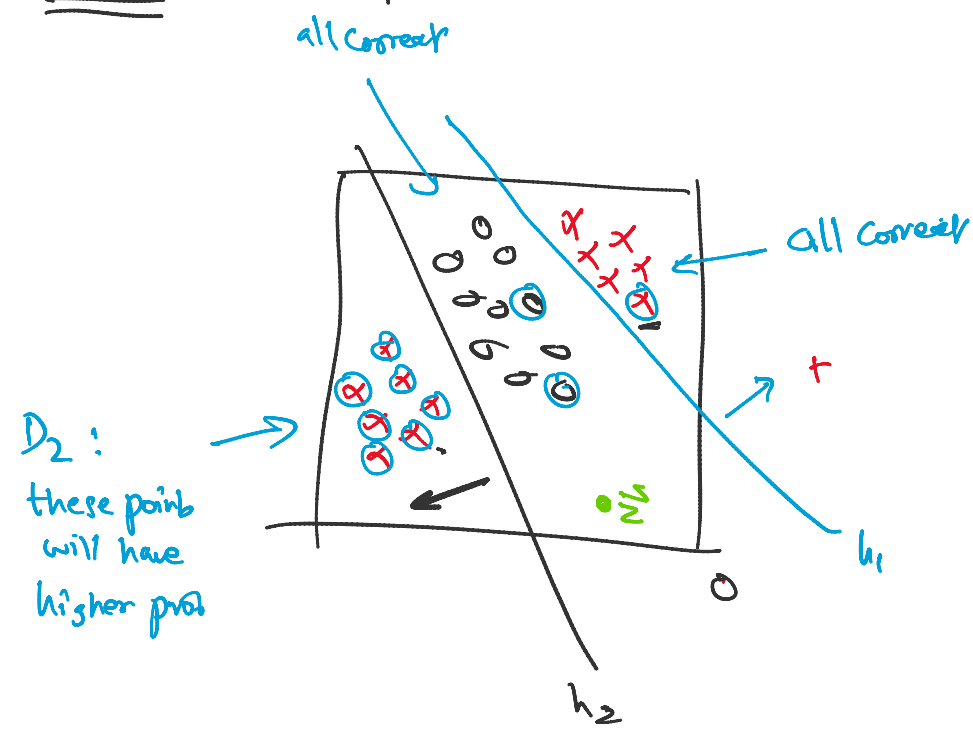
$\varepsilon_i \leq \frac{1}{100}$

$\alpha_i = \ln(99)$

$$= w_j \cdot \left(\frac{1}{\varepsilon} - 1\right)$$

normalize $\vec{w}$

$$w_j' = \frac{w_j'}{\sum w_j}$$



$h_1(\vec{z}) \rightarrow 0 \cdot \alpha_1$

$h_2(\vec{z}) \rightarrow 0 \cdot \alpha_2$

final prediction: <u>not</u> simple majority voting

weighted vote

Unsupervised Methods

Pattern mining

clustering

Pattern Mining

frequently occurring trends

↳ co-occurrence patterns

↳ Itemset Mining

transaction dataset

$I = \{$ set of items $\} = \{A, B, C, D, E\}$

$T = \{$ subsets of items $\} =$

1. $\{A, B, D, E\}$
2. $\{B, C, E\}$
3. $\{A\ B\ D\ E\}$
4. $\{A\ B\ C\ E\}$
5. $\{A\ B\ C\ D\ E\}$
6. $\{B\ C\ D\}$

D

$X = \{A, B\} \equiv AB$

$Sup(X) = 4$

$T_i \subseteq I$

$|I|$ is usually very large

↳ look

$|D| = n$ → very large

→ not so large

Q: what are the frequently occurring subsets?

$sup(x) \equiv$ absolute support$(x) = Count(x)$

$$\underbrace{\hat{P}(X)}_{\text{prob of } X} = \frac{Sup(X)}{n}$$

(joint prob of all the items in X)

relative support $= \hat{P}(X)$

$X \subseteq I$

X is some itemset / subset of I

minsup ≡ Minimum support threshold

absolute: 3

relative: 0.5

Q: given minsup threshold, find <u>all</u> frequent patterns

i.e. all $X \subseteq I$

such that $sup(X) \geq$ minsup

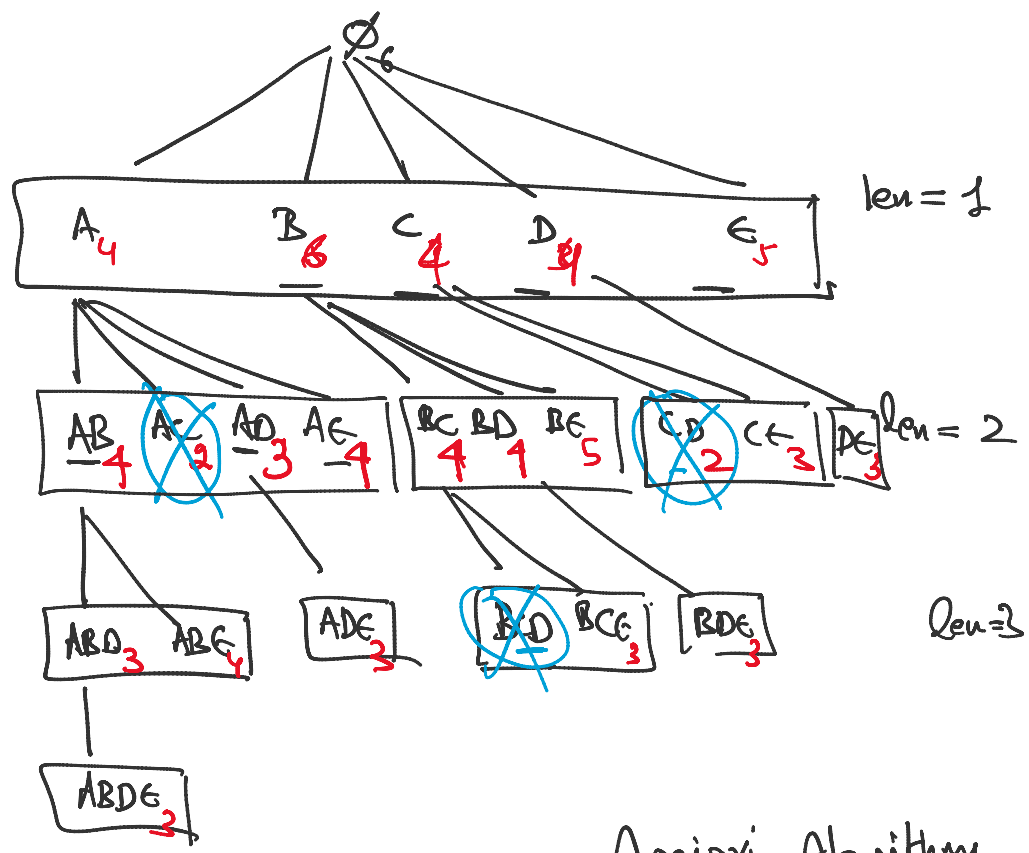
or $P(X) \geq$ minsup (fraction)

$I \equiv$ set of items

# of possible subsets $\underbrace{2^{|I|}}_{\text{Combinatorial explosion}}$

1	ABDE
D 2	BCE
3	ABDE
4	ABCE
5	ABCDE
6	BCD

search tree minsup = 3



ABDE

AB AD AE BD DE DE

BCE

BC BE CE

Apriori Algorithm

$F =$ set of all frequent itemsets!