ANNOUNCEMENTS

1. HW5 late due ~~tomorrow, 5pm~~ ; HW6 due next Mon, 5pm
2. Grading policies updated on course website

Turn your non-note-taking devices off now

Last week: ERM, bias–variance tradeoff

find $\hat{h} \in \mathcal{H}$ s.t. $\arg\min_{h \in \mathcal{H}} \hat{\varepsilon}(h)$ → training error

$$\varepsilon(\hat{h}) \leq \varepsilon(h^*) + 2\sqrt{\frac{1}{2n}\log\frac{2k}{\dots}}$$

- $\varepsilon(\hat{h})$: gen error of $\hat{h}$
- $\varepsilon(h^*)$: gen err of h^*
- n: num of samples
- $k = |\mathcal{H}|$: complexity of $\mathcal{H}$

if $\mathcal{H}$ becomes more cmplx — {linear, poly, quad},
then $\varepsilon(h^*) \downarrow$

TRADEOFF b/w $\varepsilon(h^*)$, $2\sqrt{\dots}$ when $\mathcal{H} \uparrow$ (complexity of)

Complexity – regularized ERM –

$$\hat{h} = \underset{h \in \mathcal{H}}{\arg\min}\ \hat{\varepsilon}(h) + \lambda \Omega(h)$$

measures complexity of "h"

Bayesian Viewpoint of ERM!

linear regr. $\min \frac{1}{n} \sum (y - \hat{y})^2 + \lambda^2 \|\theta\|^2 \longrightarrow$ MAP estimate

$\frac{1}{n} \sum (y - \hat{y})^2$: cost fn

Practical advice on training ML algorithms

ChatGPT detector, 100 such words / "please"

Two-way cls

LR + regularization — 70/30 → Val set: 30% error

tr / val

WHAT TO DO?

Jay — 1. increase feature set — getting more complex, fixes high bias

David — 2. kernels / use a different algorithm — change obj

Tushaar — 3. Get more samples — fixes variance!

Tushaar — 4. GD did not converge — change opt

Tushaar — 5. decrease feature set — fixes high variance!

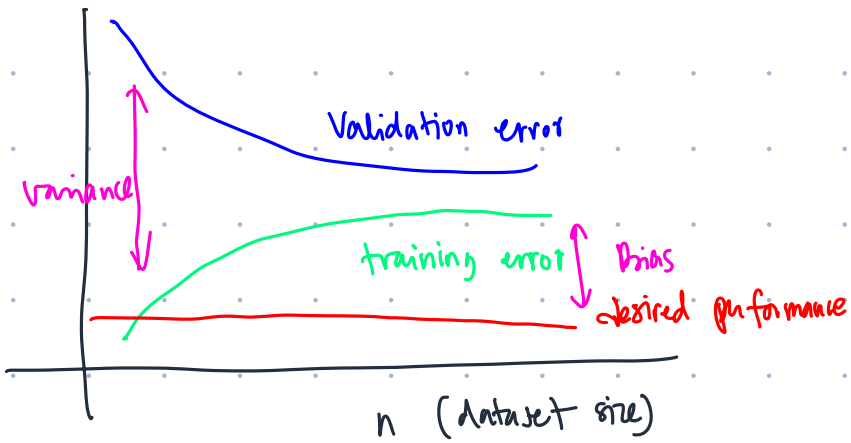
Jay — 6. may be GD diverged — change opt

Beatriz — 7 use diverse feature set, emojis — fixes high bias

Tushaar | class
4 | ✗ 4

Overfitting vs. underfitting

overfitting ⇐ 



Bad optimization vs. Bad objective

LR — $\ell(\theta)$ (objective)

Validation set — $acc(LR) < acc(SVM)$

$\ell(\theta_{LR}) < \ell(\theta_{SVM})$

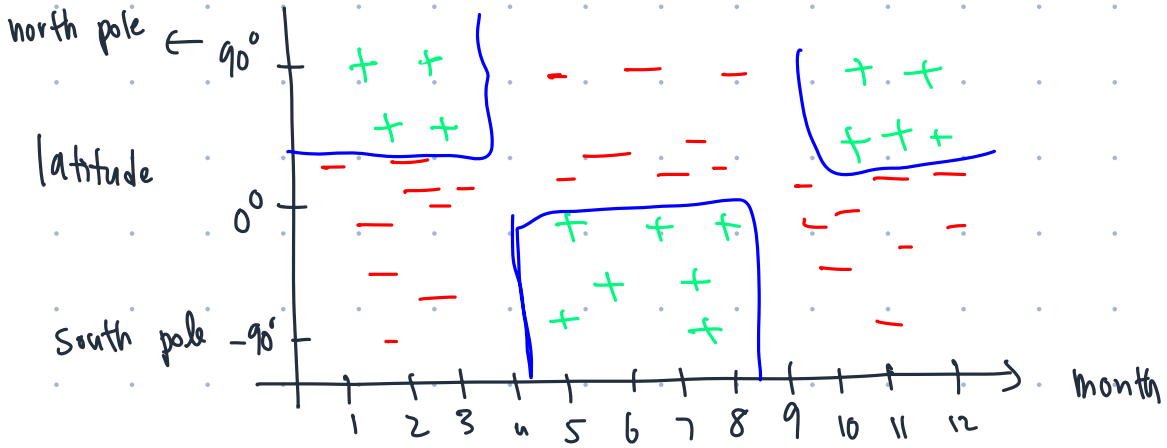
} OPT was the problem!

$acc(LR) < acc(SVM)$

$\ell(\theta_{LR}) \geq \ell(\theta_{SVM})$

} OPT worked, but objective didn't

Today: DECISION TREES — sledding on the slope!



No "line", but kernels could help!

But, Region-based segregation !!!
Seems like a good idea

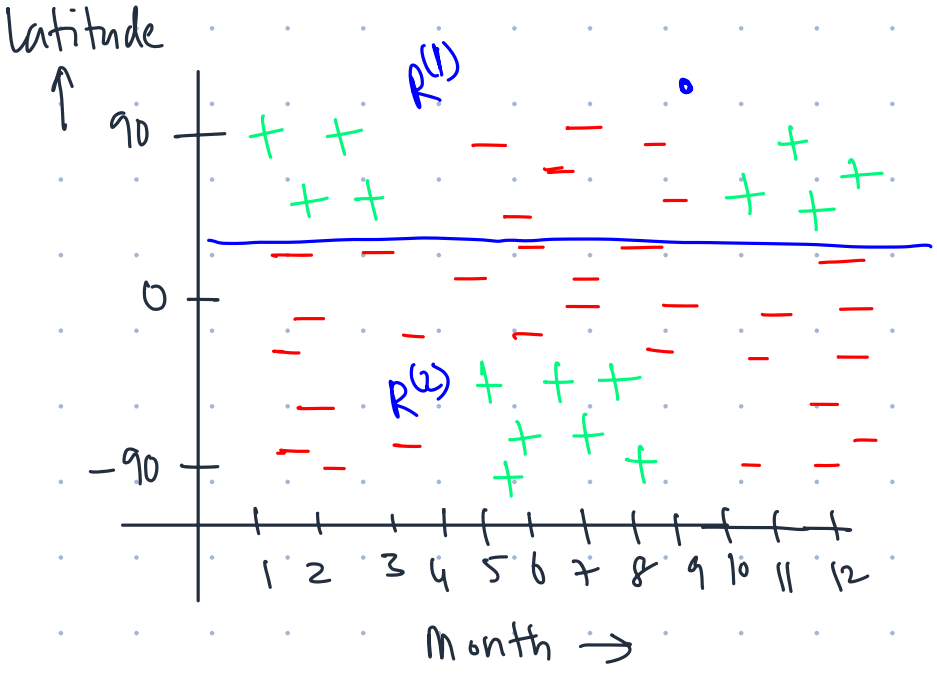
$$\mathcal{X} = R^{(1)} \cup R^{(2)} \cdots \cup R^{(r)}$$

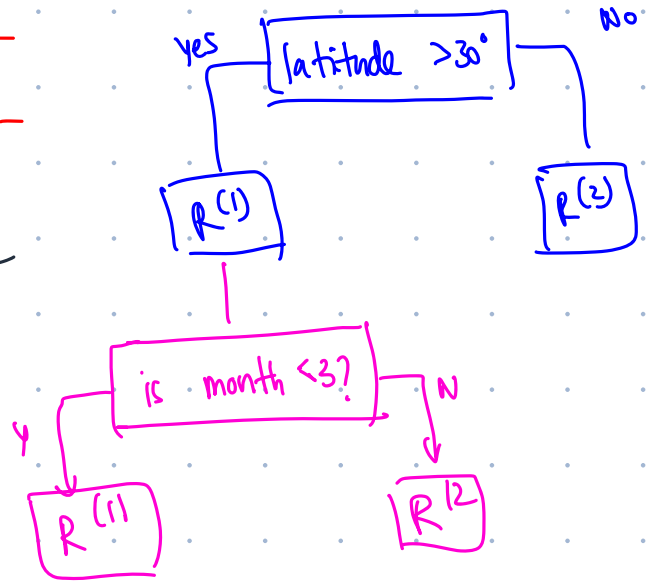
Splitting $\mathcal{X}$ into maximally-compact regions is NP-hard!

GREEDY way!

Idea: Top-down, greedy, recursive partitioning

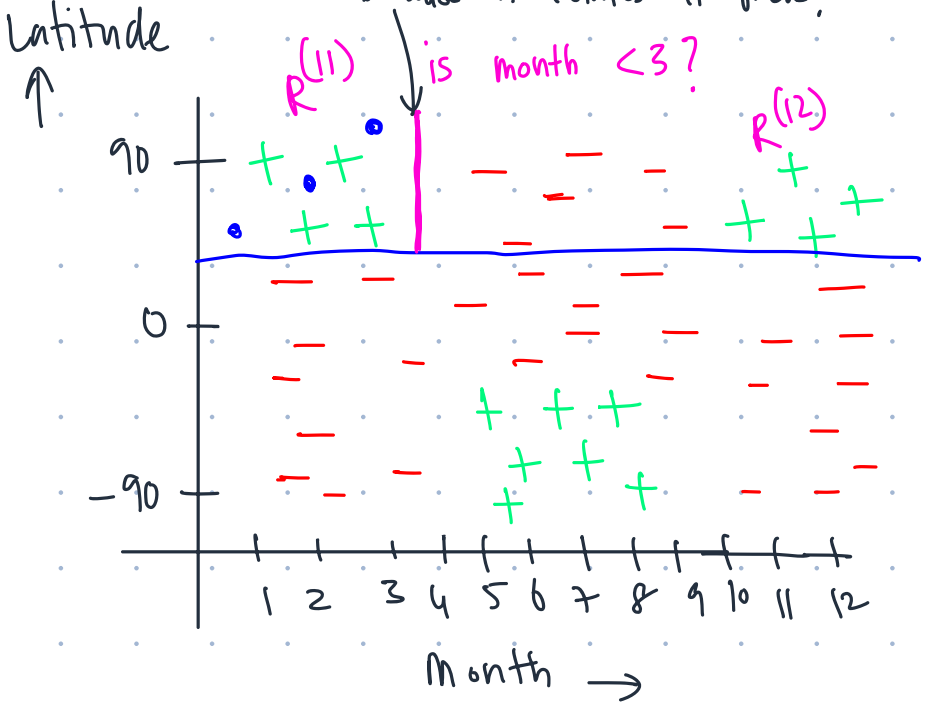
We're gonna play 20 Q with this dataset





Why this line?

Alex – because it isolates all greens!



Boon: FREE INTERPRETABILITY

Two main "points of interest"

(1) what does it mean to ask the "right" questions?

(2) when do I stop asking questions?

↓

when we have a "pure" region, further splitting is useless!

if not possible to get a pure node, just take majority vote!

Asking the "right" questions

↗ Is latitude $> 30°$?
vs.
Is month > 6?
vs.
Is latitude $> 60°$?

L = greedy best

"what is a question?"
- feature (eg. latitude, month)
- threshold (eg. $30°$, March)

What do we want from parent → children?

Some, $J(R)$ = impure a region is

$$\max \quad J(R^{(p)}) - \left(\frac{J(R^{(p1)}) + J(R^{(p2)})}{2}\right)$$

→ maximize the decrease in impurity!

more generally, we weighted avg of children:

$$\max \quad J(R^{(p)}) - \left[\frac{|R^{(p1)}|\, J(R^{(p1)}) + |R^{(p2)}|\, J(R^{(p2)})}{|R^{(p1)}| + |R^{(p2)}|}\right]$$

$|R^{(p1)}|$ = #samples in $R^{(p1)}$

Measuring impurity
$J(t)$

Idea-1: Misclassification error

what if impurity of a node = misclassification error of that region

R — 500+, 400− → misclassification rate = $\frac{400}{1000} = 0.4$

R = not pure
prediction = +ve

$R^{(1)}$ — 400+, 0− → misclassification rate = 0

$J = 0.1$

900+, 100−

(f, t)

C_1: 700+, 100− $J = \frac{100}{800}$

C_2: 200+, 0− $J = 0$

$J = 0.1$

900+, 100−

(f', t')

C_1: 500+, 100− $J = \frac{100}{600}$

C_2: 400+, 0− $J = 0$

OBJ − $0.1 - \left(\frac{800\left(\frac{100}{800}\right) + 200(0)}{1000} \right)$

$= 0.1 - \frac{100}{1000} = 0$

$0.1 - \frac{100}{1000} = 0$

① Neither splits are better
② Neither splits gave any improvement over parent

Idea-2 : Measure of randomness

Entropy measure of randomness

$$J = -\sum_{y \in \{+1,-1\}} p_y \log_2 p_y$$

p_y = proportion of samples in R with class label = y.

600+, 400−

$$p_+ = \frac{600}{1000}, \quad p_- = \frac{400}{1000}$$

$$J = -\left[0.6 \log_2 0.6 + 0.4 \log_2 0.4\right]$$

Entropy vs. misclassification error

Binary classification

$$J_{ent} = -p_+ \log p_+ - (1-p_+) \log (1-p_+)$$

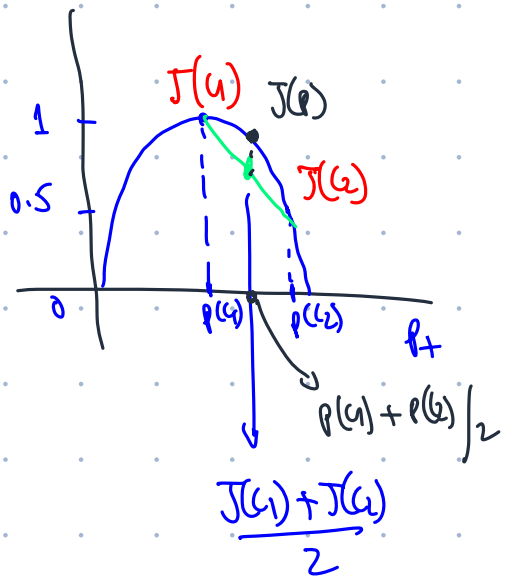
$$J_{mis} = 1 - \max(p_+, 1-p_+)$$

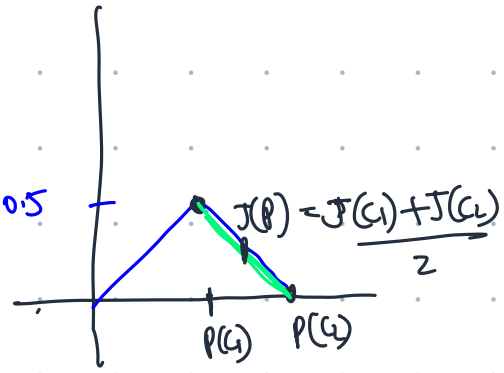
$\}$ $p_- = 1 - p_+$ in two-class

Assume: whatever splits we have, they have equal cardinalities

$$|R^{(1)}| = |R^{(2)}|$$

$$p_+^{(Parent)} = \frac{p_+^{(1)} + p_+^{(2)}}{2}$$





not strictly concave, children need not have lower error

strictly concave, children will always have lower entropy than parent

Aside: Categorical attributes?

Splitting on NetID?