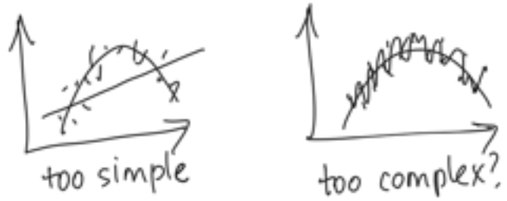
Model Selection

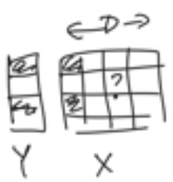
- Today: Non-Bayesian methods.



core challenge in ML: how to generalize?
→ want to predict well in the future!

mostly focus on how to identify if a model will generalize
(rest of course: ways to generalize)

- Examples:
  HW: sunspots vs Republicans in Senate
  Simple case:
  - suppose we have $N=8$ data points
  - data: x_n: 2000 dims, all binary
    all $x_{nd} \sim p$, $p = 0.5$
    $y_n = x_{n1}$ (perfect predictor of y given x)



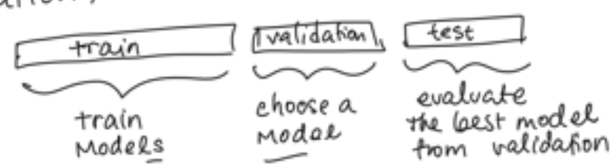
  - What is the probability that $y_n = x_{n2}$ for all n in $1..N$
    $(N=8)$: $(\frac{1}{2})^N = (\frac{1}{256})$
  - What is the probability that there is at least one spurious perfect correlation? $(d \neq 1)$

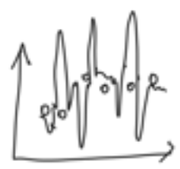
$$\Pr(\text{at least one spurious}) = 1 - \Pr(\text{no spurious at } d)^{D-1} = 1 - \underbrace{\left(\frac{255}{256}\right)}_{\to 0}^{1999} \approx 1$$

What can we do? Today: statistical techniques, but
1) always inspect for plausibility
2) this (stat methods) will not discover issues due to confounds

(assume test data is like train data)

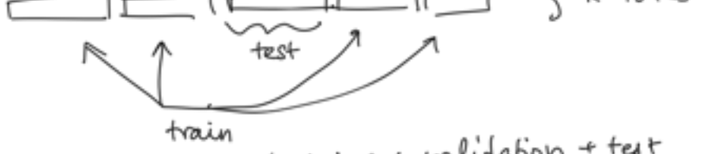
- Validation / Cross-Validation





What if we want & want confidence intervals?
[And not enough data]

↳ cross-validation:



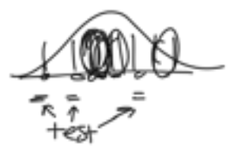
if you want: add training + validation + test
fold fold

(not indep, but can give sense of variance)

↳ Bootstrap: where we sample *with* replacement
N samples from our data; train; test on rest

stat story: $X, Y \sim \mathcal{D}$
if we train: $\hat{y} = f(x)$, err } this is the frequentist question
$E_{\mathcal{D}}[err]$ $E_{\mathcal{D}}[err^2]$

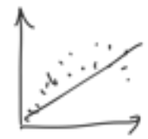
bootstrap: ~~X, Y~~ $X, Y \sim$ subset of $[X, Y]$



⇒ All of these help measure generalization (empirical approach)

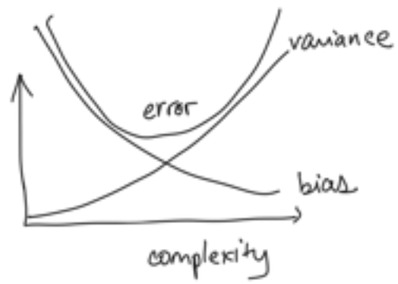
Deeper look at WHY models generalize

bias-variance trade-off



simple: a lot of bias, deviates from data BUT not thrown off by a few data ("underfitting")

complex: lot of variance, sensitive to data BUT don't deviate from data ("overfitting")



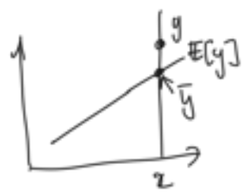
Let's formalize! We care about $\mathbb{E}_{\substack{all \\ y}}[(y - \hat{y})^2] = \mathbb{E}_{\substack{all \\ y}}[(y - f_{\mathcal{D}}(x))^2]$

for some x, assume some true process to create y's

squared error

prediction depends on data $\mathcal{D}$

real life [y is a random variable with mean $\bar{y} = \mathbb{E}[y]$



$$\mathbb{E}[(y - f_{\mathcal{D}}(x))^2] = \mathbb{E}[(y - \bar{y} + \bar{y} - f_{\mathcal{D}}(x))^2]$$

$$= \underbrace{\mathbb{E}[(y - \bar{y})^2]}_{\text{noise inherent in the real situation}} + \underbrace{\mathbb{E}[(\bar{y} - f_{\mathcal{D}}(x))^2]}_{\text{error from our modeling}} + 2\mathbb{E}[(y - \bar{y})(\bar{y} - f_{\mathcal{D}}(x))]$$

$(y - \bar{y})$: 0 mean; $(\bar{y} - f_{\mathcal{D}}(x))$: indep.

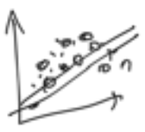
⇓ expand:

lets consider all the different $f_{\mathcal{D}}(x)$ we could

learn w/ diff datasets $\mathcal{D}$.

$$\mathbb{E}_{\mathcal{D}}\left[(\bar{y} - f_{\mathcal{D}}(x))^2\right] = \mathbb{E}\left[(\bar{y} - \bar{f}(x))^2\right] + \mathbb{E}_{\mathcal{D}}\left[(\bar{f}(x) - f_{\mathcal{D}}(x))^2\right] + 2\mathbb{E}\left[(y - \bar{f}(x))(\bar{f}(x) - f_{\mathcal{D}}(x))\right]$$

- $\mathbb{E}_{\mathcal{D}}[(\bar{y} - f_{\mathcal{D}}(x))^2]$: real world expectation, constant
- depends on $\mathcal{D}$, define $\bar{f}(x) = \mathbb{E}_{\mathcal{D}}[f_{\mathcal{D}}(x)]$
- $\mathbb{E}[(\bar{y} - \bar{f}(x))^2]$: bias^2 — mean of $f_{\mathcal{D}}(x)$ doesn't match real mean!
- $\mathbb{E}_{\mathcal{D}}[(\bar{f}(x) - f_{\mathcal{D}}(x))^2]$: variance, what you get depends on $\mathcal{D}$
- $2\mathbb{E}[(y - \bar{f}(x))(\bar{f}(x) - f_{\mathcal{D}}(x))] = 0$ (indep; mean 0)



error = noise^2 + bias^2 + variance

(bias^2 ↕ variance: control)

Note: $N \to \infty$ (or large data), var ↓

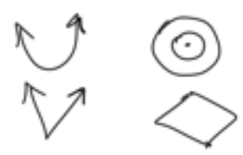
How can we manage bias, variance trade-off?

→ Regularization: penalize model complexity

e.g. $\mathcal{L}_{\mathcal{D}}(w) \to \min_w \mathcal{L}_{\mathcal{D}}(w) + \lambda R(w)$

$R(w)$: $\|w\|_2^2$ Ridge Reg. / L2 — convex ☺

$\|w\|_1$ L1/lasso — convex, non-diff.



choose λ via cross-validation.

⇒ Ensembles: use a committee to make a decision (vote/avg)

Idea: bias doesn't increase/change, variance goes down

→ (BAGGING) ⇒ sample datasets

→ RF / extra random forests

→ Adaboost / Boosting

sequentially learn predictors that are indep of prev predictors

Concept Check:

Test set & Train set for some data

	Model A	Model B	
test:	0.60	0.70	accuracy
train:	0.90	0.80	

Which model is better?

test. 0.60
train. 0.95

train: 0,72

1) Will adding more data help for model A? model B?

2) Should we have a validation split in our train?

3) Which model will have stable or similar results if we run on another dataset to train.