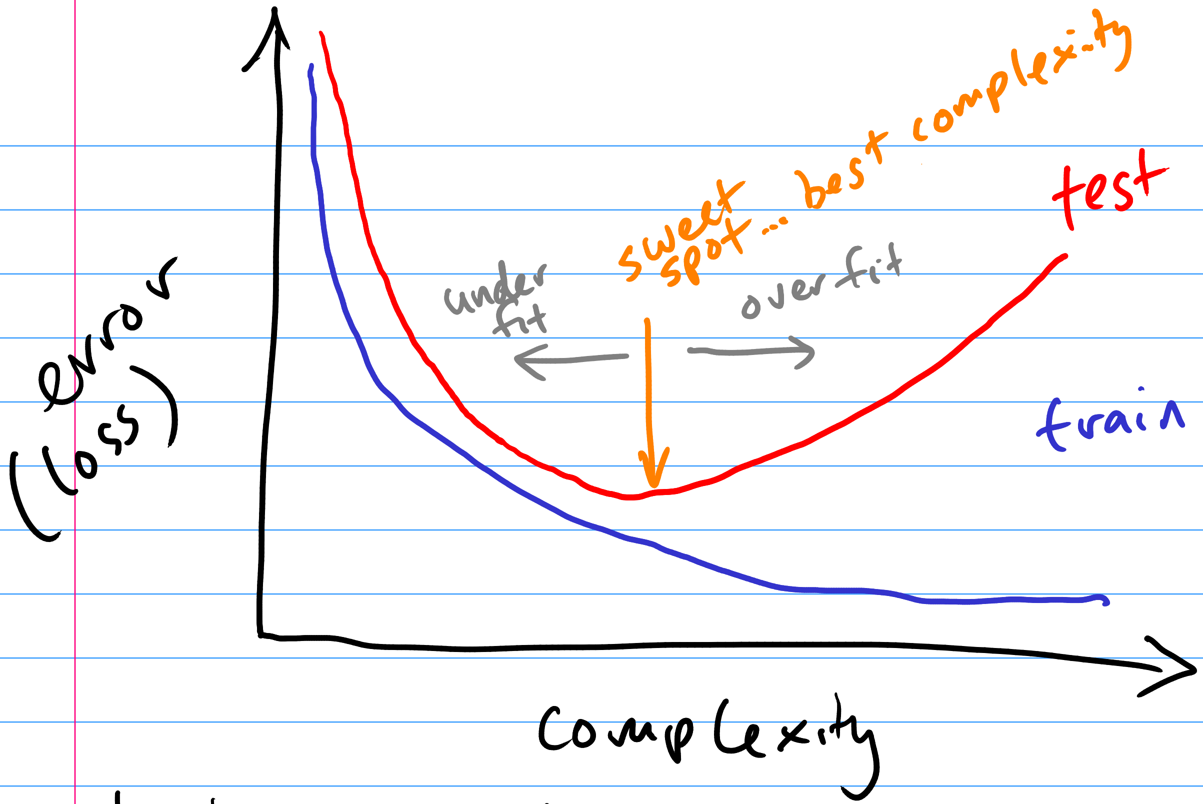
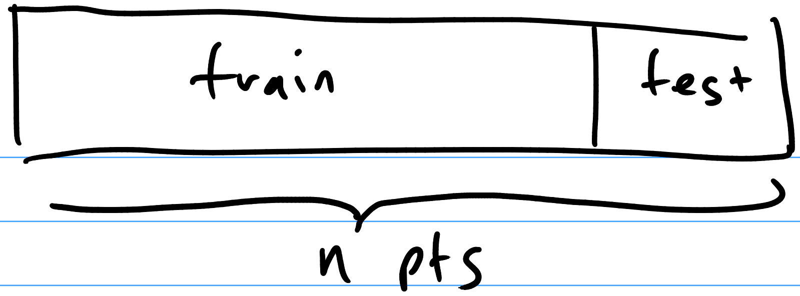
Resampling + parameter tuning

Goal: know how to tune models w/o overfitting

cross-validation (CV) $\begin{cases} \text{leave-one-out} \\ \text{bootstrap} \end{cases}$





Different ways to split

tuning parameters:

- Ridge λ, Lasso, any other penalization $\lambda R(\vec{w})$
- k in k-NN # of neighbors
- h bandwidth in kernel $K(\vec{x}, \vec{x}') = \exp\left(-\frac{\|\vec{x} - \vec{x}'\|^2}{2h^2}\right)$

Why we have a testing set

testing set must be new

1) We want to estimate performance on new data

Goal	Error rate	
Friend suggestion	10%	OK
Pedestrian detector	10%	not ok

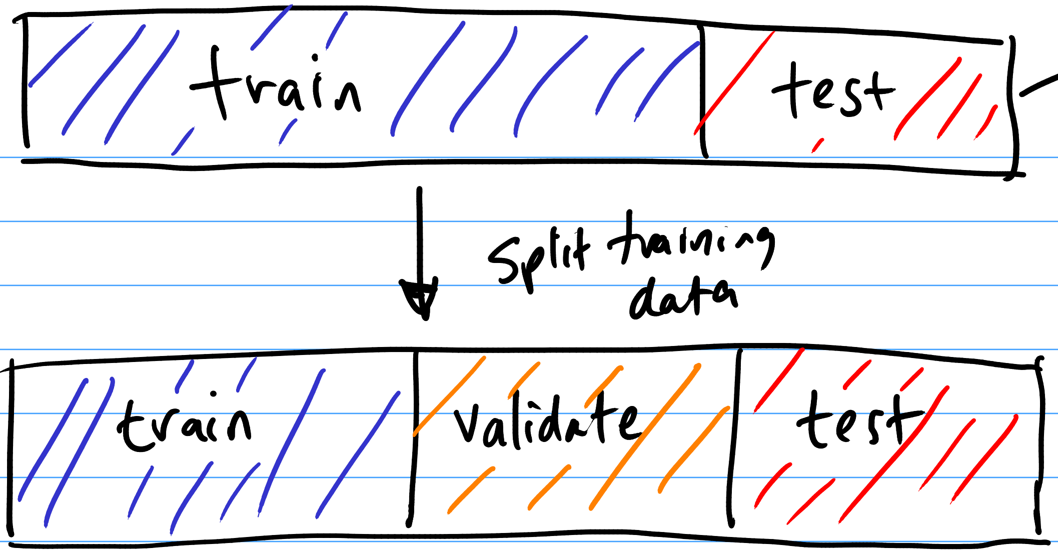
can't do both

2) tune params somehow

If we use test set to select λ we will underestimate the error for unseen data:

λ_{opt} depends on the test set, in this case.

TO ESTIMATE ERROR RATE ON UNSEEN DATA, MODEL CAN NEVER SEE TEST SET



always for evaluating test error
estimate of error on new data

Split data 3 ways once

Assumptions:
All n data pts are i.i.d.

Problem:
- shrunk training set could be hard to fit a big model
- high variance since you pick one validation, test set

Alg. for selecting λ

model selection

for $\lambda \in \Lambda$ ← set of possible λ's

  Fit model with param λ to $\boxed{\text{training data}}$

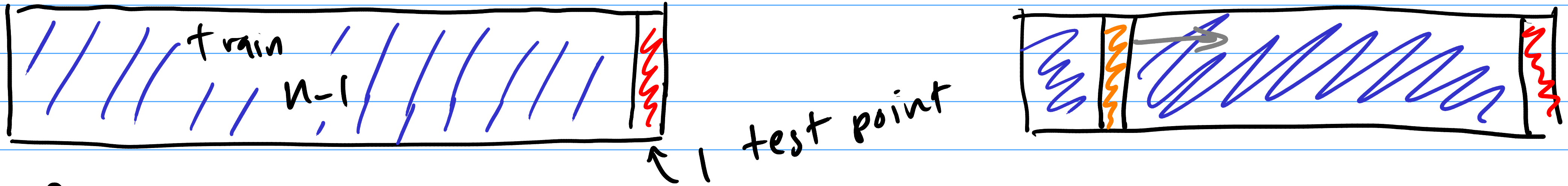
  Evaluate error(λ) with $\boxed{\text{validation data}}$

Pick model $\lambda_{opt} = \arg\min_{\lambda} \text{error}(\lambda)$

Fit model with param λ_{opt} to $\boxed{\text{train}}$ + $\boxed{\text{validation}}$ data

Evaluate error with $\boxed{\text{testing data}}$

leave-one-out CV (LOOCV)



for $i = 1, \dots, n$

select model (λ) ← Fit model to $\boxed{X^{(-i)}, \vec{y}^{(-i)}}$ all data except i

Test model on $\vec{x}_i, y_i \longrightarrow$ error(i)

report avg. test error $\frac{1}{n} \sum_{i=1}^{n}$ error(i)

advantage:
- less variance in estimate of error
- training set size $n-1$
- tricks for Ridge that fit all n models at once

disadvantage:
- have to fit n models ($n \cdot (n-1)$ fits for nested CV)

k-fold CV

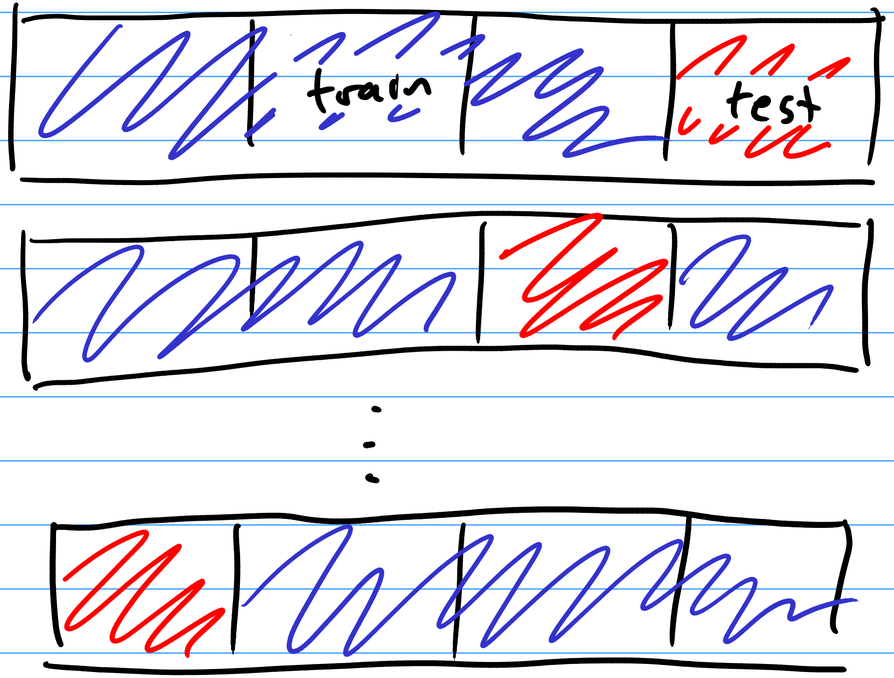
- most common technique

divide dataset into k blocks

$k \approx 10, 100$

$\frac{n}{k}$ points

k=4



error(1)

error(2)

⋮

error(k)

$$\frac{1}{k}\sum_{i=1}^{k} \text{error}(i)$$

↓

estimate of performance of avg. model

advantage : • fit k models

• test set n/k, lower variance than LOOCV

disadvantage : • small k shrinks training data, might not be reliable

$k=n \Rightarrow$ LOOCV