What *is* overfitting?

You've probably heard about it

I've delayed it intentionally

I want to do justice to this extremely important, central concept of machine learning

And I'm dissatisfied with the usual definitions!

Machine learning

is about algorithms

that allow us to increase model complexity

and optimize

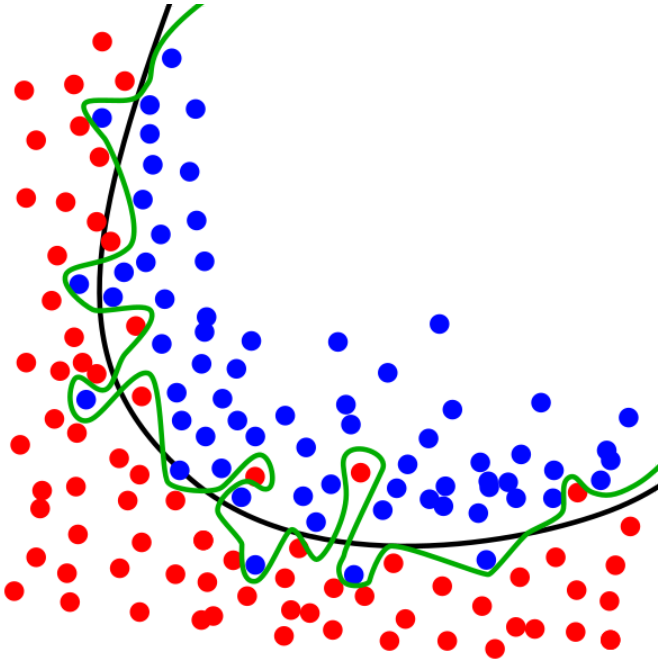
on larger datasets

over larger sets of parameters

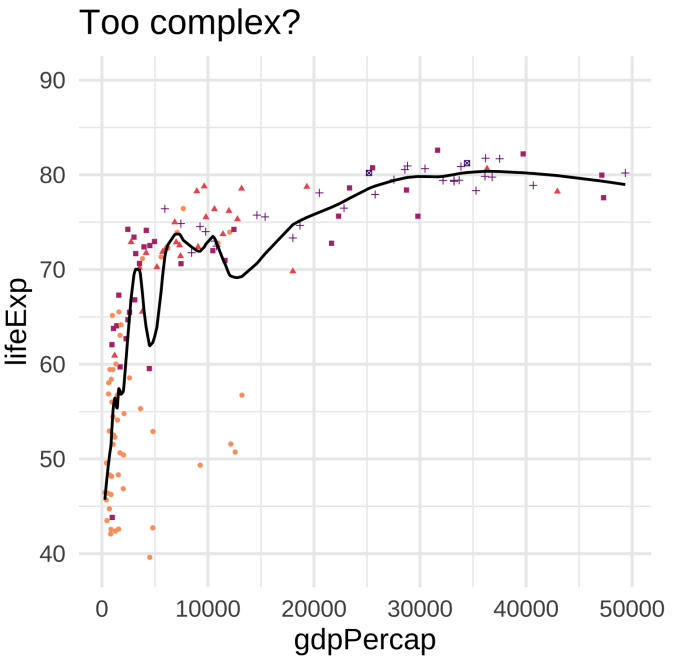
and even infinite-dimensional function spaces

With great fitting comes great responsibility

ML increases danger of this specific kind of modeling problem



Source: Wikipedia



Example in week 1

Outline

- **Complexity**: typical machine learning definition
- **Generalization**: simple probabilistic definition
- **Bias-variance**: statistical distinctions
- **Causality**: scientific/philosophical applications
- **Anthropology**: human learning and overfitting IRL (*supplemental but brief and useful for life in general?*)
- **Validation**: standard method to prevent overfitting (*to variation*, cannot help us with bias)

Typical ML definition of overfitting

Motivating idea: assume the model will be "deployed"

I.e. Some time after fitting the model will be used on new data

> "It is difficult to make predictions, especially about the future" - Danish saying

Overfitting the "training data"

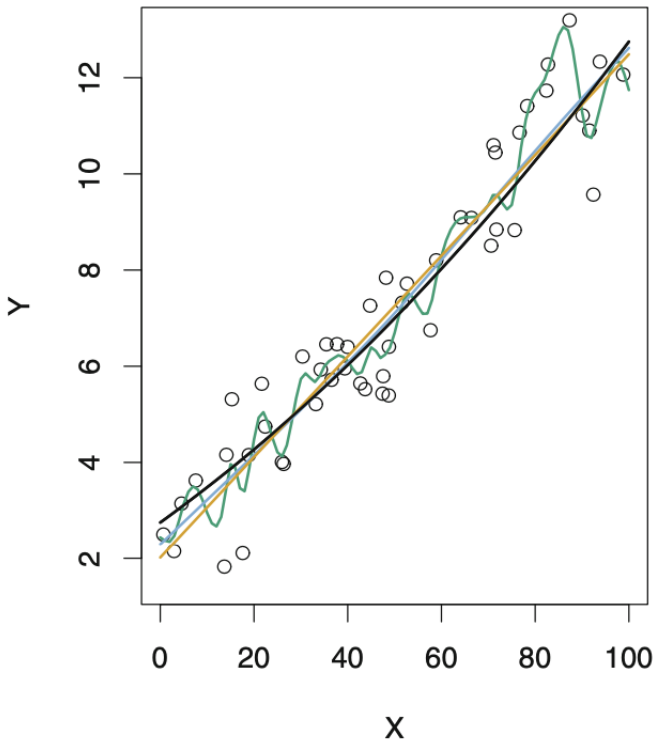
Using a model that is *too complex*

Specifically, one where the complexity is larger than the optimal complexity for predicting on a *new observation* or *new sample* of **test/validation data**

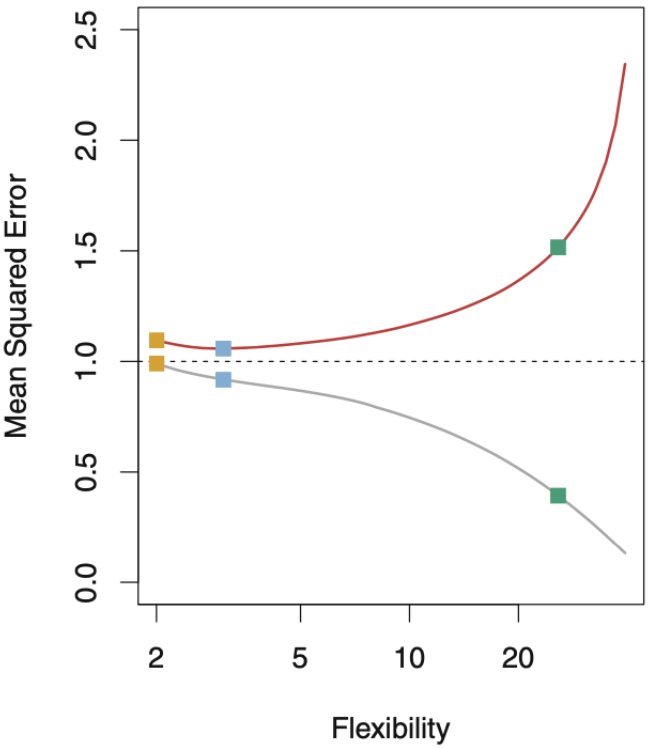
- $\hat{f}_\lambda$ model fitted/estimated on **training data**
- λ tuning parameter that *penalizes* complexity
  - larger λ, simpler model
- λ^* optimal param. value for predicting/classifying *new data*
- Overfitting: using $\hat{f}_\lambda$ for some $\lambda < \lambda^*$

ISLR Figure 2.10

True model is simple

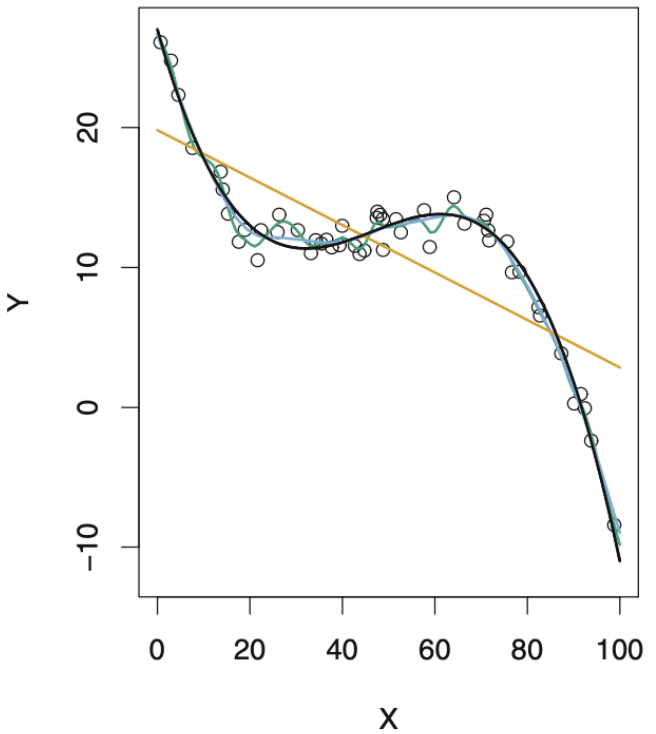


High complexity overfits

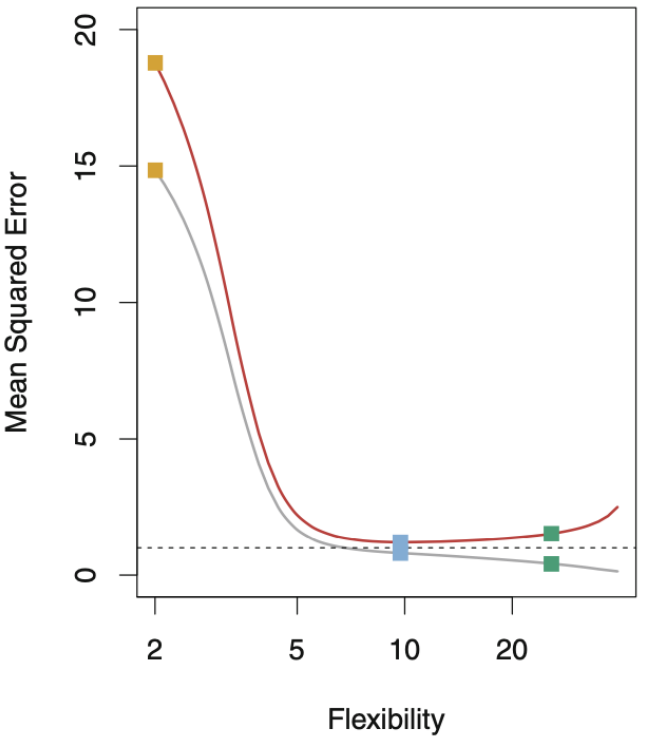


ISLR Figure 2.11

True model is complex



Low complexity underfits



The end

Most discussions of overfitting end there

Some go on a little more, relating it to **bias-variance** trade-off

Overfitting: *low bias but overwhelmingly high-variance*

(we'll do that soon)

Generalization

and a

probabilistic definition

Motivation: what is the *probability distribution* of the test data?

Two kinds of generalization

ML/AI books/courses talk about "generalization error"

Over-used term, same word / *importantly different* meanings

Generalization to a new observation from...

- the same distribution or DGP
- a different (but related) distribution

and *corresponding reasons for doing poorly*

- variance ("random/unstructured error", high entropy)
- bias ("systematic/structured error", low entropy)

Think about distributions

Suppose the training data is sampled i.i.d. from
$(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_n, y_n) \sim F$

and the test data is sampled i.i.id from
$(\mathbf{X}'_1, y'_1), (\mathbf{X}'_2, y'_2), \ldots, (\mathbf{X}'_{n'}, y'_{n'}) \sim F'$

In-distribution (ID) generalization: $F = F'$

Under/overfitting, **variability problem**, larger n allows more complex models to be fit

Out-of-distribution (OOD) generalization: $F \neq F'$

"Covariate/distribution/dataset shift", **bias problem**, larger n may not help. Need **modeling assumptions** like $F' \approx F$

Optimism and ID generalization

Observation: training error generally appears lower than test/validation error. Why?

Risk vs *empirical* risk minimization

$$R(g) = \mathbb{E}_F[L(\mathbf{X}, Y, g)]$$

$$\hat{f} = \arg\min_g \hat{R}(g) = \arg\min_g \frac{1}{n} \sum_{i=1}^{n} L(\mathbf{x}_i, y_i, g)$$

Fact: for some $\mathrm{df}(\hat{f}) > 0$ (depends on problem/fun. class)

$$\mathbb{E}_{Y|\mathbf{x}_1,\ldots,\mathbf{x}_n}[R(\hat{f}) - \hat{R}(\hat{f})] = \frac{2\sigma_\varepsilon^2}{n}\mathrm{df}(\hat{f}) > 0$$

Optimism, ID gen., and degrees of freedom

Linear case

If $\hat{f}$ is linear with p predictors (or p basis function transformations of original predictors) then

$$\mathrm{df}(\hat{f}) = p$$

Fairly general case

For many ML tasks and fitting procedures

$$\mathrm{df}(\hat{f}) \text{ increases as } \frac{1}{n\sigma_\varepsilon^2} \sum_{i=1}^{n} \mathrm{Cov}(\hat{f}(\mathbf{x}_i), y_i) \text{ increases}$$

Take-aways about optimism and ID gen.

- Empirical risk *underestimates* actual risk (ID generalization error)
- The magnitude of this bias is called **optimism**
- Optimism generally increases with function class complexity
  - e.g. for linear functions, increases linearly in p
- For a fixed function class, optimism decreases linearly in n
- Too much optimization $\rightarrow$ overfitting $\rightarrow$ more optimism

Two kinds of overfitting?

Many sources identify overfitting as a threat to generalization

Typically only apply this to **ID generalization**, and have solution strategies to avoid the **variability problems** due to overfitting

But overfitting is also a threat to OOD generalization!

This kind of generalization is often what we practically want

There are serious **bias problems** due to overfitting

Let's start using new terminology

Overfitting to variation and *overfitting to bias*

Now let's jump from probability to statistics

And talk about why we always need to care about *both kinds of generalization*

Statistical aspects of overfitting

Motivation: all models are wrong, including F and F'

or

Motivation: overfitting to noise... what's noise?

What *is* noise?

The effect of all (causal?) factors not captured by the model

Could be different reasons for failing to capture

- Measurement issues
- Wrong functional relationship
- Variables excluded (maybe not even measured or defined)

Does not require physical randomness (which maybe doesn't exist...)

Something considered **noise** in one setting, or by one modeler, could be **signal** to a different observer

Noise and residuals in regression

(*One of the most "agnostic" or minimal-theory ways of defining regression is as estimation of a conditional expectation function, without assuming any specific functional form like linearity*). The "noise" in regression is defined as

$$\varepsilon = y - \mathbb{E}_F[y|\mathbf{x}]$$

But this is math, not applied data analysis! Requires assuming a probability distribution F / random variable model

Otherwise, how do we define expectation?

We never observe ε, only residuals $r_i = y_i - \hat{f}(\mathbf{x}_i)$ of some model $\hat{f}$ *fit with specific assumptions/algorithms*

What is bias/variability?

Two analysts start with different assumptions

e.g. linearity vs flexible non-parametric methods

Fit different regression functions

Compute different residuals

See different patterns (or lack thereof) in residuals

Something considered **variation** in one setting, or by one modeler, could be **bias** to a different observer

Data science

In the "real world" there is a data generation process (DGP)

We *assume* this can be modeled as an i.i.d. sample from a probability distribution F

Probability model / mathematical justification for our methods

All models are wrong

Could model DGP as a mixture of distributions F and F' (heterogeneity), or time-varying F^t

Training/test data randomly shuffled?

Generalization in/out of distribution?

Two data scientists diverged...

Starting with different assumptions about DGP

Use different strategies to avoid overfitting

e.g. different ways of splitting into training and test data

Something considered **ID generalization** in one setting, or by one modeler, could be **OOD generalization** to a different observer

Statistical take-aways

Mathematical distinctions between ID and OOD generalization rely on assumptions (as do statistical distinctions between bias and variability)

ML methods for avoiding overfitting are motivated by ID generalization, guard against **overfitting to variability**

In applications, ID/OOD distinctions break down. If we probe them a bit we find it's more gray area / ambiguous

Most scientists and decision-makers care about external validity, conceptually related to OOD generalization

Overfitting to bias is a serious, widely neglected problem!

Considerations

of the

scientific and philosophic variety

with respect to overfitting

Motivation: does science overfit? Can philosophy of science help us understand how to prevent it? What about causality?

Stability, invariance, and causality

Idea: causal relationships should persist despite (some) changes in "background conditions"

Bradford Hill criteria for causation

> **Consistency**: Has [the observed association] been repeatedly observed by different persons, in different places, circumstances and times?

Apparently people think about causality this way

Can use the idea to motivate statistical methods for causal inference

Overfitting as a threat to causal inference

Bradford Hill criteria for causation

> "the larger the association, the more likely that it is causal." - Wikipedia, not Hill

Hill:

> the death rate from cancer of the lung in cigarette smokers is nine to ten times the rate in non-smokers

Problem: overfitting can make associations appear stronger

e.g. proportion of variation in `lifeExp` explained by `gdpPercap`

Increase flexibility, explain higher proportion... stronger evidence of causality? 🤔

Generalization, novelty, and severity

Philosophy of science: prediction vs "accommodation"

Prediction: happens in time before observation/measurement

Accommodation: theory built to explain past observation/data

Accurate prediction is better evidence in favor of a scientific theory than mere accommodation

ML: What's better evidence in favor of the model?

Popper and Lakatos: **temporal novelty**

Zahar, Gardner, Worrall: **use-novelty** (or problem novelty)

Mayo: novelty is not necessary. **Severity** is necessary

Anthropology?

ie. overfitting IRL

(in real life)

Motivation: do *we* overfit? ("Are we the baddies?")

Disclaimer: I am not an anthropologist *or* self-help author

How/why are humans different?

We seem to better at *learning* than other animals

Human eyes are different, allowing us to see where others are looking

Social learning

"Monkey see, monkey do"

Lots of animals learn by imitation, but humans specifically take imitation to a whole different level

Over-imitation, causal opacity, cultural evolution...

Validation

Estimate test error directly

using "validation data" / "test data"

i.e. a new set of data, "unseen" by $\hat{f}$

Indep. samples $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ and $D' = \{(\mathbf{x}'_i, y'_i)\}_{i=1}^{n'}$

Estimate $\hat{f}$ on D, evaluate $\hat{f}$ on D'

Motives

- Debiasing risk estimate. Since $\hat{f}$ does not depend on D', it is not **overfit to the variability** in D'
- If $\hat{f}$ is overfit to D then its test error on D' will be large (complexity too high, variability too high)
- Actual practice: analogous to "deploying an ML model in production"
- Philosophy of science: use novelty, actual prediction (not accommodation)
- Tukey: Exploratory Data Analysis vs Confirmatory
- Use test error to choose **model complexity** / **amount of regularization**

Choosing model complexity

Using test/validation data

Indep. samples $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and $D' = \{(\mathbf{x}_i', y_i')\}_{i=1}^{n'}$

- Estimate $\hat{f}_\lambda$ on D for a "path" or grid of λ values
- Evaluate $\hat{f}_\lambda$ on D' and choose $\hat{\lambda}$ accordingly (e.g. with minimum loss)
- Refit $\hat{f}_{\hat{\lambda}}$ on full data $D \cup D'$, this is our final model

Common when computational cost of fitting one model is high

Cross-validation

When computational cost of fitting one model is not too high

Idea: swap D and D' in previous process and get two estimates, $\hat{R}(\hat{f}_\lambda)$ and $\hat{R}(\hat{f}'_\lambda)$

Average these and choose $\hat{\lambda}$ using the average (e.g. minimizer)

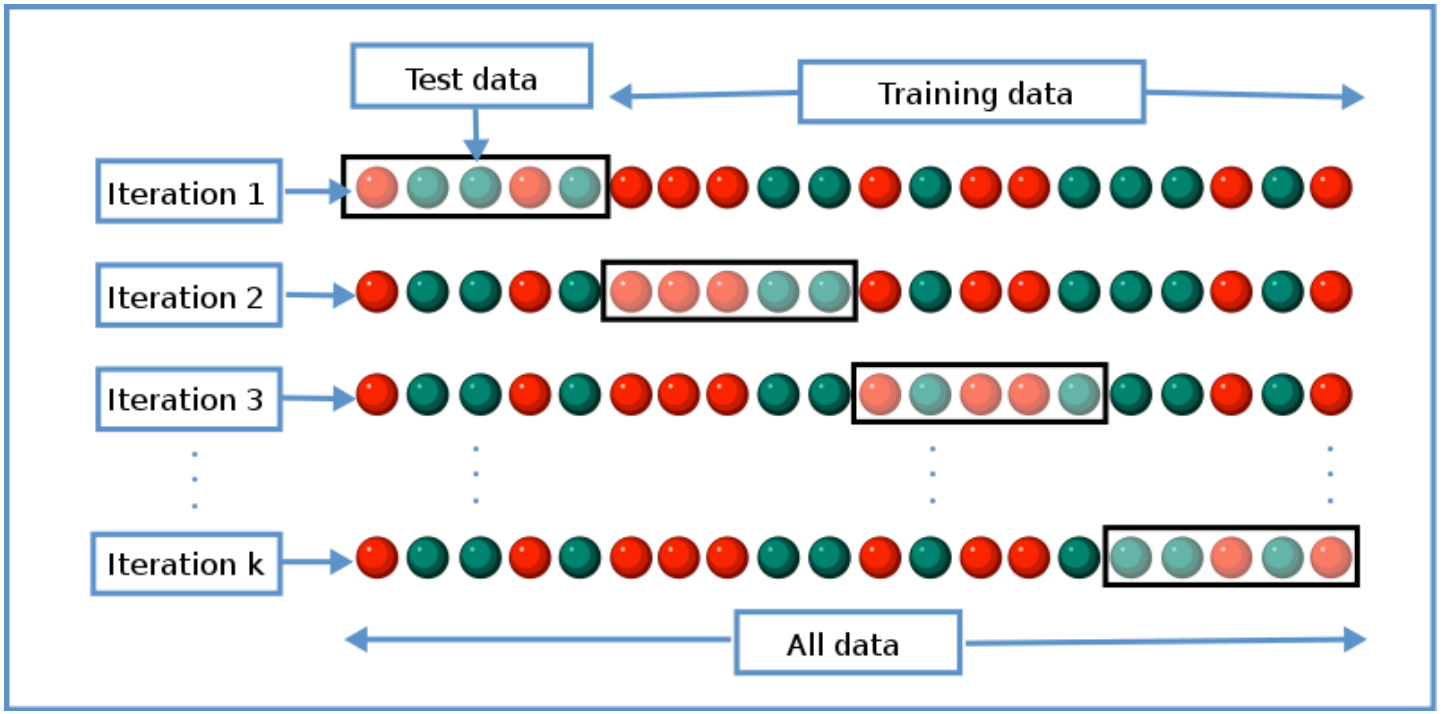
Idea: apply the same process with multiple independent "folds" of data

K-fold cross-validation

Each subset used once as test set, and $K-1$ times for training

Minimize $\hat{R}_{K\text{-cv}}(\lambda) = \frac{1}{K}\sum_{k=1}^{K}\hat{R}_k(\hat{f}_\lambda^{(k)})$

Cross-validation cartoon



Gives K estimates of test error (risk) at each λ

Credit: Wikipedia

K-fold cross-validation

Each subset used once as test set, and $K-1$ times for training

Choose $\hat{\lambda}$ to minimize

$$\hat{R}_{K\text{-cv}}(\lambda) = \frac{1}{K}\sum_{k=1}^{K} \hat{R}_k(\hat{f}_\lambda^{(k)})$$

where $\hat{f}_\lambda^{(k)}$ is fit on the dataset that "holds out" the kth fold

Then refit model $\hat{f}_{\hat{\lambda}}$ at that value of $\hat{\lambda}$ on the entire dataset

Lessons about cross-validation

- Think of it as a way to **choose model complexity**
- **Beware** common cross-validation errors! From Duda and Hart quoted in MLstory

> ... the same data were often used for designing and testing the classifier. This mistake is frequently referred to as "testing on the training data." A related but less obvious problem arises *when a classifier undergoes a long series of refinements guided by the results of repeated testing on the same data. This form of "**training on the testing data**" often escapes attention until new test samples are obtained*.

Lessons about cross-validation

- **Beware** common cross-validation errors! From ESL:

> Ideally, the test set should be kept in a "vault," and be brought out only at the end of the data analysis. *Suppose instead that we use the test-set repeatedly, choosing the model with smallest test-set error. Then the test set error of the final chosen model will underestimate the true test error*, sometimes substantially.

- Cross-validate entire model building pipeline (not just one step), and only do it once -- or at *least* not many times

- Choosing K: larger $\to \hat{R}_{K\text{-cv}}$ has lower bias, more variance. Often use $K = 5$ or 10

Regularization

- Fancy sounding word for "simplification," simpler models
- Increases bias to reduce variance

Cross-validation

- Fit and evaluate models on different subsets of data
- Choose amount of regularization/complexity
- Re-using data *more than once* $\longrightarrow$ overfitting again