https://www.science.smith.edu/~jcrouser/SDS293/lectures/10-linear-model-selection-pt1.pdf

LECTURE 10:

LINEAR MODEL SELECTION PT. 1

October 16, 2017

SDS 293: Machine Learning

Outline

- Model selection: alternatives to least-squares
- Subset selection
  - Best subset
  - Stepwise selection (forward and backward)
  - Estimating error
- Shrinkage methods
  - Ridge regression and the Lasso
  - Dimension reduction
- Labs for each part

Back to the safety of linear models…

$$\mathtt{Y} \approx \beta_0 + \beta_1 \mathtt{X}_1 + \ldots + \beta_p \mathtt{X}_p$$

Bias vs. variance

Discussion

How could we

reduce the variance?

Subset selection

- **Big idea:** if having too many predictors is the problem maybe we can get rid of some

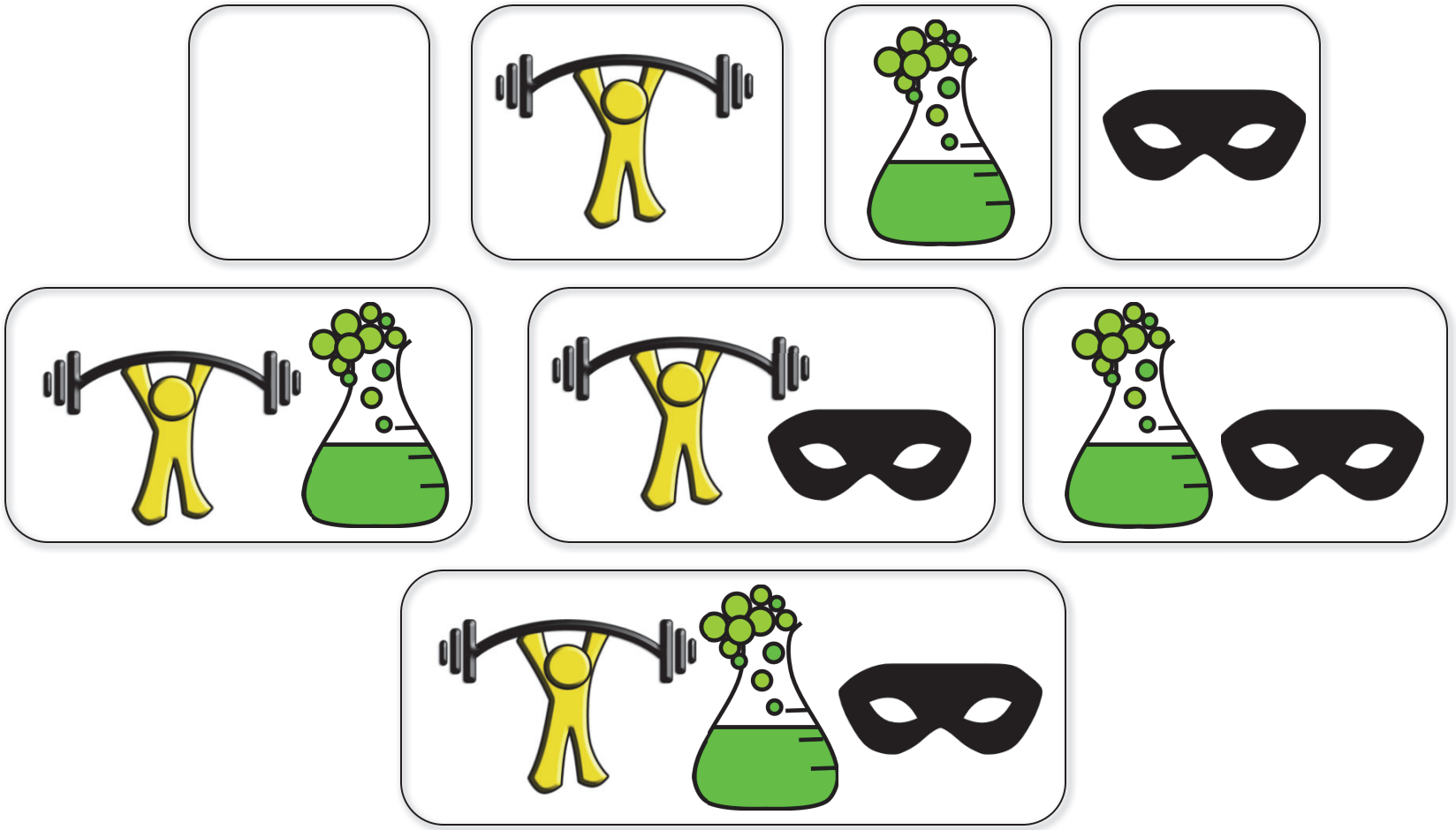
- **Problem:** how do we choose?

Flashback: superhero example

$$height = \beta_1 \left(\quad \right) + \beta_2 \left(\quad \right) + \beta_3 \left(\quad \right)$$

Image credit: Ming Malaykham

Best subset selection: try them all!



Finding the “best” subset

Start with the null model M_0 (containing no predictors)

1. For $k = 1,2, \ldots, p$:
   a. Fit all (p choose k) models that contain exactly p predictors.
   b. Keep only the one that has the smallest RSS (or equivalently the largest R^2). Call it M_k.
2. Select a single “best” model from among $M_0 \ldots M_p$ using cross-validated prediction error or something similar.

Discussion

Question 1: why not just use the one with the lowest RSS?

Answer: because you'll always wind up choosing the model with the highest number of predictors (why?)

Discussion

Question 2: why not just calculate the cross-validated prediction error on all of them?

Answer: so… many... models...

A sense of scale…

- We do a lot of work in groups in this class
- How many different possible groupings are there?
- Let's break it down:

47 individual people

1,081 different groups of two

16,215 different groups of three…

Model overload

- Number of possible models on a set of p predictors:

$$\sum_{k=1}^{p} \binom{p}{k} = 2^p$$

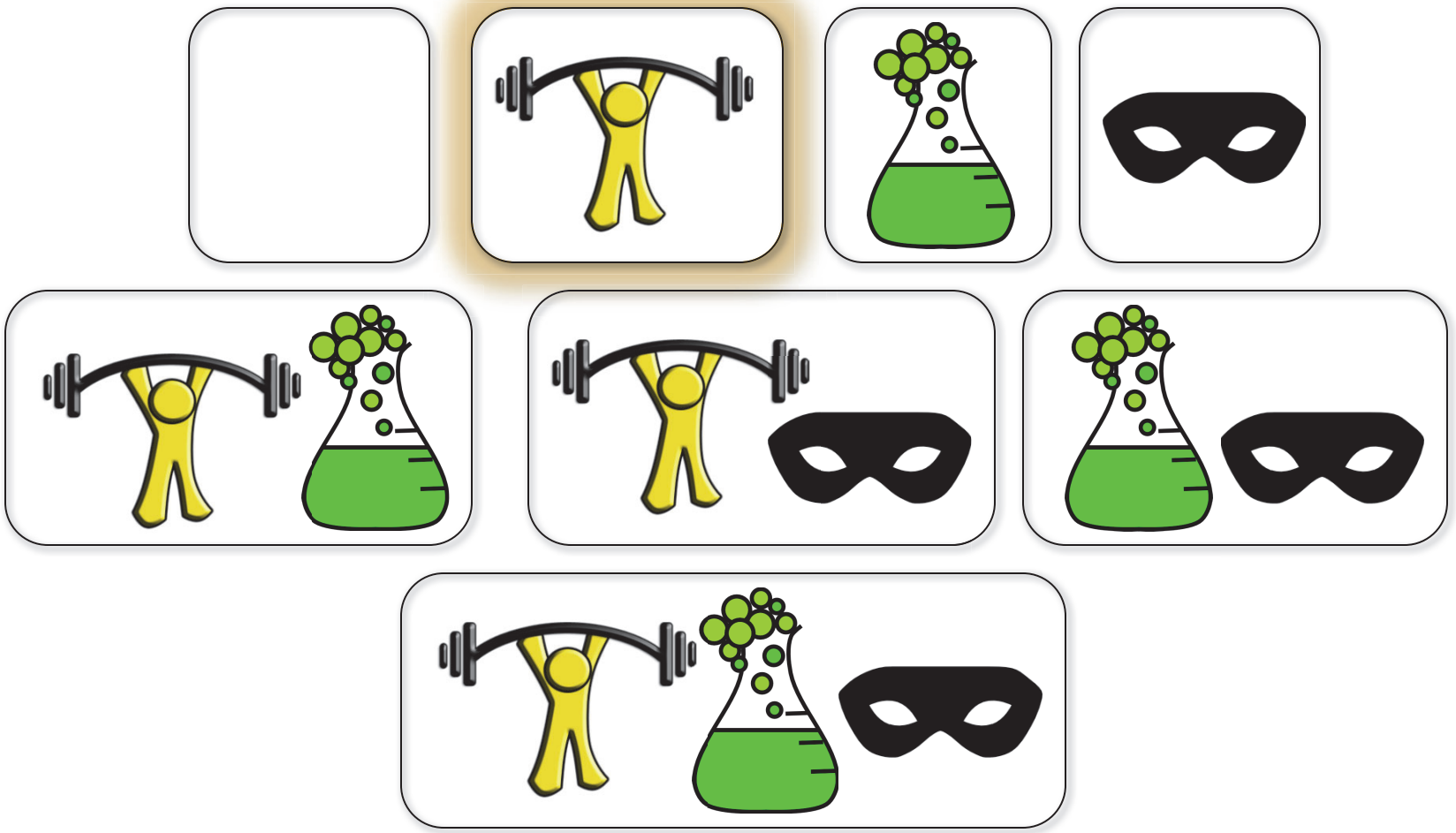
- On 10 predictors: **1,024** models
- On 20 predictors: **1,048,576** models

A bigger problem

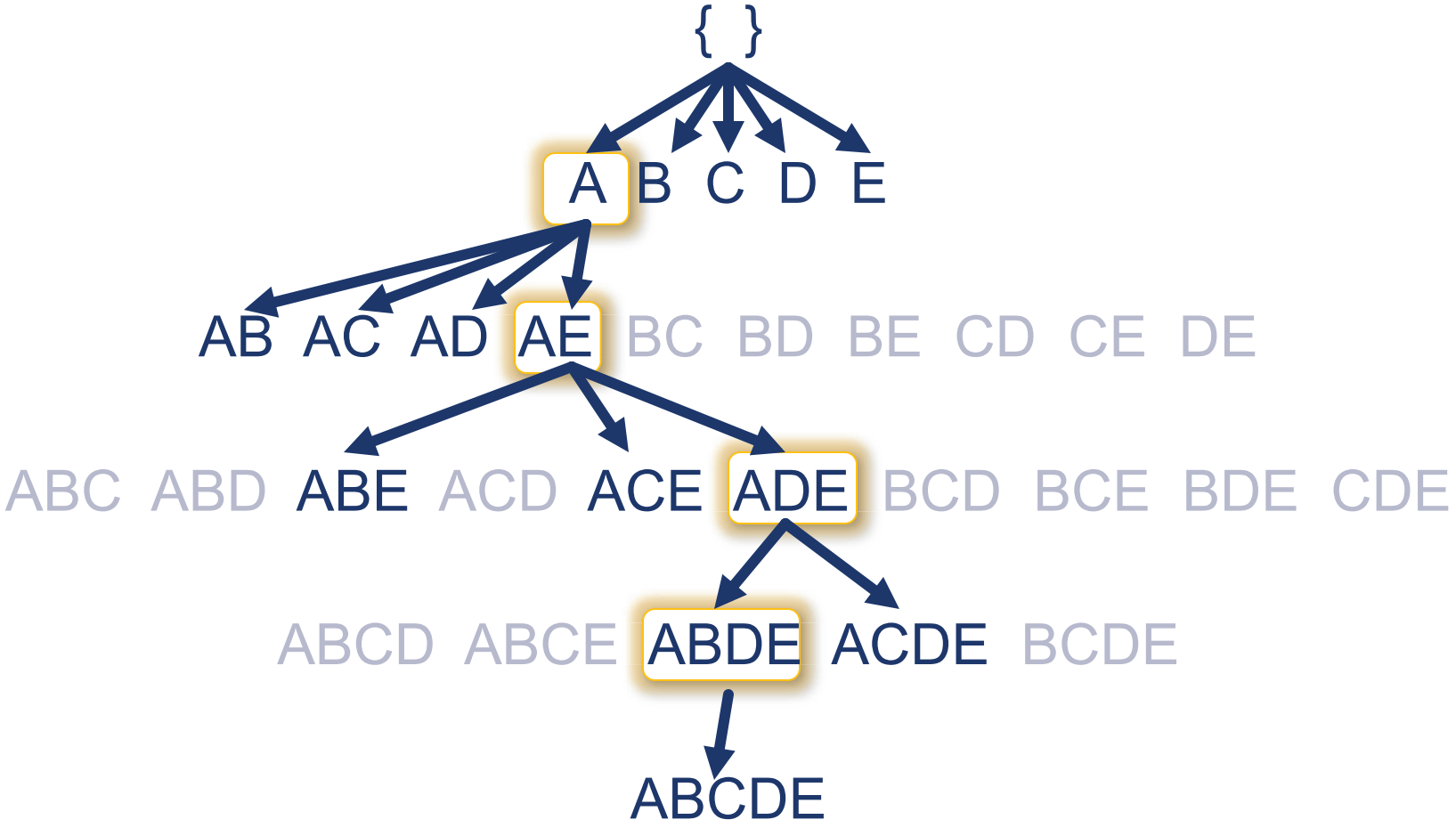
Question: what happens to our estimated coefficients as we fit more and more models?

Answer: the larger the search space, the larger the variance. We're overfitting!

What if we could eliminate some?



A slightly larger example ($p = 5$)



Best subset selection

Start with the null model M_0 (containing no predictors)

1. For $k = 1, 2, \ldots, p$:
   a. Fit all $(p \; choose \; k)$ models that contain exactly p predictors.
   b. Keep only the one that has the smallest RSS (or equivalently the largest R^2). Call it M_k.
2. Select a single "best" model from among $M_0 \ldots M_p$ using cross-validated prediction error or something similar.

Forward selection

Start with the null model M_0 (containing no predictors)

1. For $k = 1, 2, \ldots, p$:
   a. Fit all $(p - k)$ models that augment M_{k-1} with exactly 1 predictor.
   b. Keep only the one that has the smallest RSS (or equivalently the largest R^2). Call it M_k.
2. Select a single "best" model from among $M_0 \ldots M_p$ using cross-validated prediction error or something similar.

Stepwise selection: way fewer models

- Number of models we have to consider:

$$\sum_{k=1}^{p}\binom{p}{k} = 2^p \rightarrow \sum_{k=0}^{p-1}(p-k) = 1 + \frac{p(p+1)}{2}$$

- On 10 predictors: 1024 models → **51 models**
- On 20 predictors: over 1 million models → **211 models**

Forward selection

Question: what potential problems do you see?

Answer: there's a risk we might prune an important predictor too early. While this method usually does well in practice, it is not guaranteed to give the optimal solution.

Forward selection

Start with the null model M_0 (containing no predictors)

1. For $k = 1,2,\ldots,p$:
   a. Fit all $(p-k)$ models that augment M_{k-1} with exactly 1 predictor.
   b. Keep only the one that has the smallest RSS (or equivalently the largest R^2). Call it M_k.
2. Select a single "best" model from among $M_0 \ldots M_p$ using cross-validated prediction error or something similar.

Backward selection

Start with the full model M_p (containing all predictors)

1. For $k = p, (p-1), \ldots, 1$:
   a. Fit all k models that reduce M_{k+1} by exactly 1 predictor.
   b. Keep only the one that has the smallest RSS (or equivalently the largest R^2). Call it M_k.
2. Select a single "best" model from among $M_0 \ldots M_p$ using cross-validated prediction error or something similar.

Forward selection

Question: what potential problems do you see?

Answer: if we have more predictors than we have observations, this method won't work (why?)

Choosing the optimal model

- Flashback: measures of **training** error (RSS and R^2) aren't good predictors of **test** error (what we care about)
- Two options:
  1. We can **directly** estimate the test error, using either a validation set approach or cross-validation
  2. We can **indirectly** estimate test error by making an adjustment to the training error to account for the bias

Adjusted R^2

- **Intuition:** once all of the useful variables have been included in the model, adding additional junk variables will lead to only a small decrease in RSS

$$R^2 = 1 - \frac{RSS}{TSS} \rightarrow R^2_{Adj} = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$$

- Adjusted R^2 pays a penalty for unnecessary variables in the model by dividing RSS by $(n\text{-}d\text{-}1)$ in the numerator

AIC, BIC, and C_p

- Some other ways of penalizing RSS

Estimate of the variance of the error terms

$$C_p = \frac{1}{n}\left(RSS + 2d\hat{\sigma}^2\right)$$

$$AIC = \frac{1}{n\hat{\sigma}^2}\left(RSS + 2d\hat{\sigma}^2\right)$$

Proportional for least-squares models

$$BIC = \frac{1}{n}\left(RSS + \log(n)d\hat{\sigma}^2\right)$$

More severe penalty for large models

Adjust or validate?

Question: what are the benefits and drawbacks of each?

	Adjusted measures	Validation
Pros	Relatively **inexpensive** to compute	More **direct** estimate (makes fewer assumptions)
Cons	Makes more **assumptions** about the model – more opportunities to be wrong	More **expensive**: requires either cross validation or a test set

LECTURE 11:

LINEAR MODEL SELECTION PT. 2

October 18, 2017

SDS 293: Machine Learning

Flashback: subset selection

- **Big idea:** if having too many predictors is the problem maybe we can get rid of some

- Three methods:
  - **Best subset:** try all possible combinations of predictors
  - **Forward**: start with no predictors, greedily add one at a time
  - **Backward**: start with all predictors, greedily remove one at a time

"greedy" = Add/remove whichever predictor improves your model **right now**

Flashback: comparing methods

	Best Subset Selection	Forward Selection	Backward Selection
How many models get compared?	2^p	$1+\frac{p(p+1)}{2}$	$1+\frac{p(p+1)}{2}$
Benefits?	Provably optimal	Inexpensive	Inexpensive; doesn't ignore interaction
Drawbacks?	Exhaustive search is expensive	Not guaranteed to be optimal; ignores interaction	Not guaranteed to be optimal; breaks when $p>n$

Flashback: choosing the optimal model

- We know measures of training error (RSS and R^2) aren't good predictors of test error (what we actually care about)
- Two options:
  - We can **indirectly** estimate test error by making an adjustment to the training error to account for the bias:

$$R^2_{adj} \quad C_p \quad AIC \quad BIC$$

Pros: inexpensive to compute

Cons: makes additional assumptions about the model

  - We can **directly** estimate the test error, using either a validation set approach or a cross-validation approach

Discussion: potential problems?

Only training on a subset of the data means our model is **less accurate**

From the kitchen of: Grandma SDS

Recipe for: Best Subset Selection

First divide the data into training and test sets

Preheat the null model M_0 with no predictors.* on the **training set**

1. For $k = 1, 2, \ldots, p$:
   a. Fit all the models that contain exactly k predictors.
   b. Keep only the model with the smallest training error. Call it M_k.

2. ~~Estimate the error,~~ and select a single "best" model from among $M_0 \ldots M_p$
   ^ Calculate the error rate on **the test set**

Cross-validation: how would this work?

From the kitchen of: Grandma SDS

Recipe for: Best Subset Selection

Preheat the null model M_0 with no predictors.

1. For $k = 1, 2, \ldots, p$:
   a. Fit all the models that contain exactly k predictors.
   b. Keep only the model with the smallest training error. Call it M_k.

2. ~~Estimate the error,~~ and select a single "best" model from among $M_0 \ldots M_p$
   ^ Use k-fold cross-validation to calculate the CV error

Flashback: subset selection

- **Big idea:** if having too many predictors is the problem maybe we can get rid of some

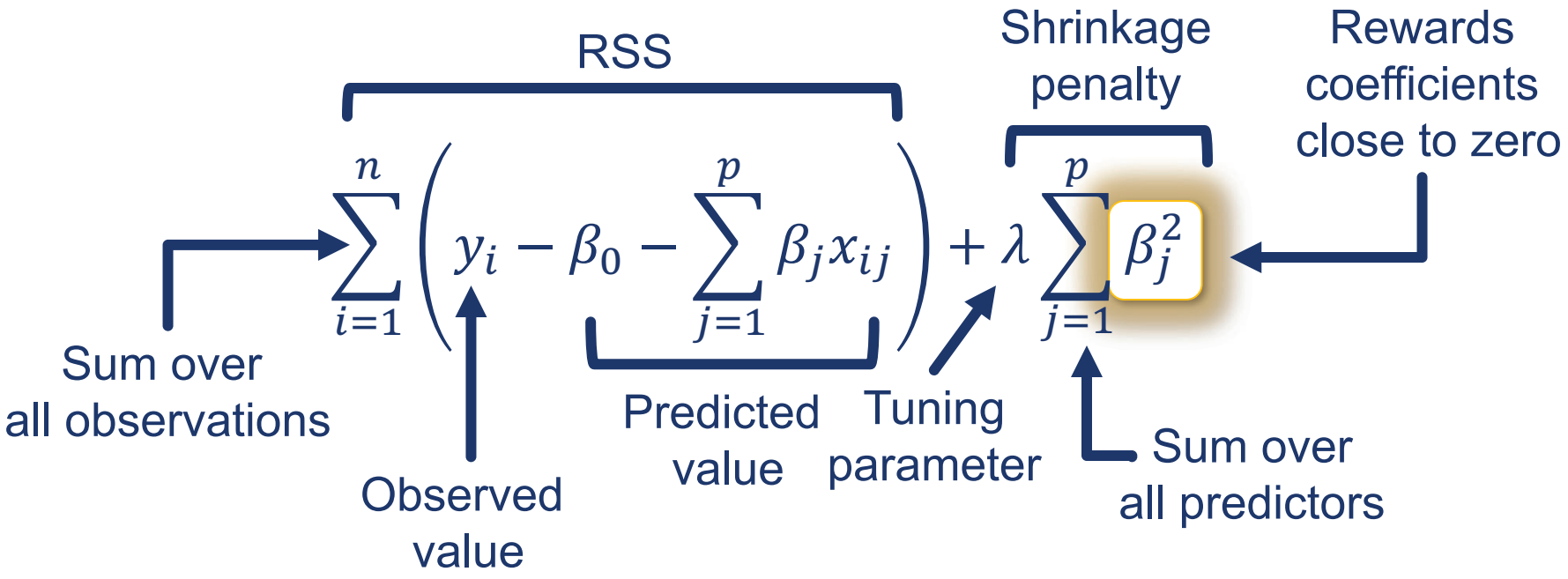
- Three methods:
  - **Best subset:** try all possible combinations of predictors
  - **Forward**: start with no predictors, greedily add one at a time
  - **Backward**: start with all predictors, greedily remove one at a time

Common theme of subset selection:

ultimately, individual predictors are either **IN** or **OUT**

Approach 1: ridge regression

- **Big idea**: minimize RSS plus an additional penalty that rewards small (sum of) coefficient values



* In statistical / linear algebraic parlance, this is an ℓ_2 penalty

Ridge regression: caveat

- RSS is scale-invariant*
- **Question**: is this true of the shrinkage penalty?

$$\overbrace{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)}^{\text{RSS}} + \overbrace{\lambda\sum_{j=1}^{p}\beta_j^2}^{\text{Shrinkage penalty}}$$

- **Answer:** no! This means having predictors at different scales would influence our estimate… need to first **standardize** the predictors by dividing by the standard deviation

* multiplying any predictor by a constant doesn't matter

Discussion

- **Question:** why would ridge regression improve the fit over least-squares regression?
- **Answer:** as usual, comes down to **bias-variance tradeoff**
  - As λ increases, flexibility decreases: $\downarrow$ variance, $\uparrow$ bias
  - As λ decreases, flexibility increases: $\uparrow$ variance, $\downarrow$ bias
  - **Takeaway:** ridge regression works best in situations where least squares estimates have high variance: trades a small increase in bias for a large reduction in variance

Comparing ridge regression and the lasso

- Efficient implementations for both (in R and python!)
- Both significantly reduce variance at the expense of a small increase in bias
- **Question**: when would one outperform the other?
- **Answer:**
  - When there are relatively many equally-important predictors, **ridge regression** will dominate
  - When there are small number of important predictors and many others that are not useful, **the lasso** will win

Lingering concern…

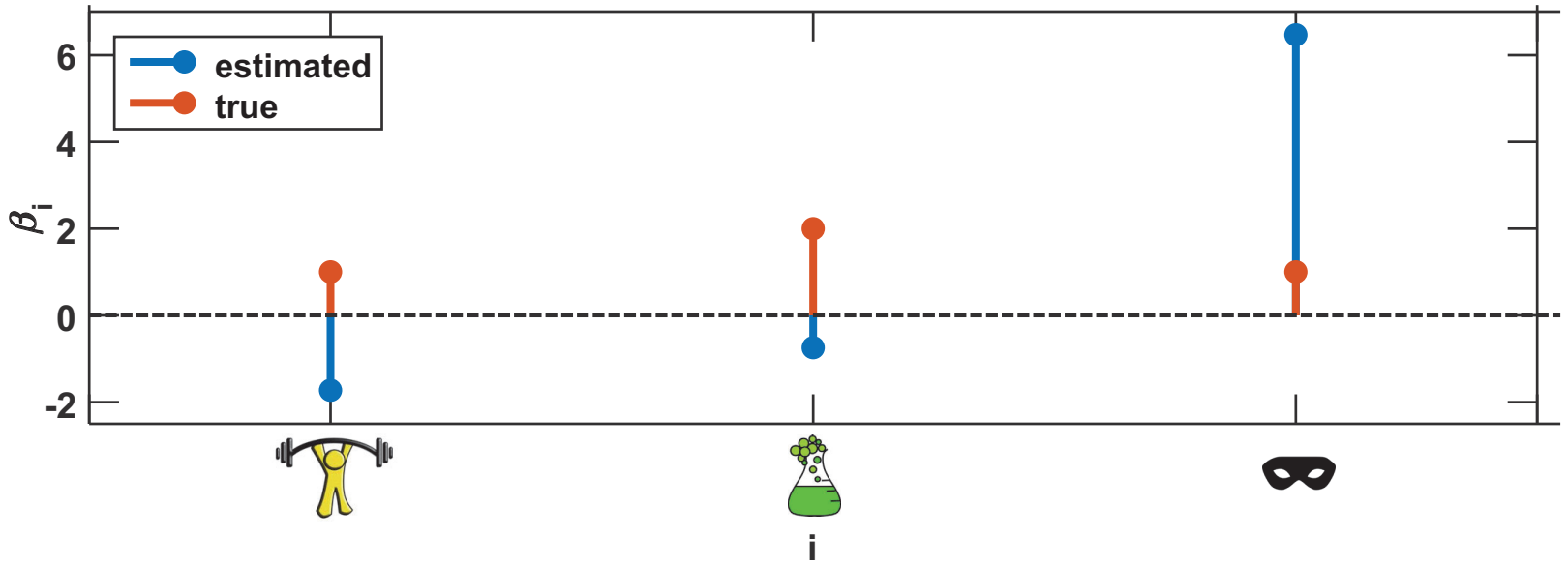
- **Question:** how do we choose the right value of λ?
- **Answer:** sweep and cross validate!
  - Because we are only fitting a single model for each λ, we can afford to **try lots of possible values** to find the best ("sweeping")
  - For each λ we test, we'll want to calculate the **cross-validation error** to make sure the performance is consistent

Recap: Ridge Regression and the Lasso

- Both are “shrinkage” methods
- Estimates for the coefficients are ***biased*** toward the origin
  - Biased = “prefers some estimates to others”
  - Does not yield the true value in expectation
- Question: why would we **want** a biased estimate?

Estimate for β

- When we try to estimate using OLS, we get the following:



(Relatively) huge difference between actual and estimated coefficients

What's going on here?

$$\begin{bmatrix} 232.03 \\ 156.29 \\ 113.82 \\ 229.07 \\ 287.72 \end{bmatrix} = 1 \begin{bmatrix} 63.9 \\ 28.9 \\ 54.3 \\ 69.8 \\ 50.4 \end{bmatrix} + 2 \begin{bmatrix} 54.0 \\ 45.1 \\ 13.3 \\ 49.5 \\ 85.4 \end{bmatrix} + 1 \begin{bmatrix} 59.1 \\ 36.9 \\ 33.7 \\ 59.7 \\ 67.9 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}$$

$$\approx avg\left(\text{  }, \text{  } \right)$$

- Some dimensions are redundant
  - Little information in 3rd dimension not captured by the first two
  - In linear regression, redundancy causes noise to be **amplified**