Cross Validation and the Bias Variance Trade-off

(morning)

Goals

- Be able to…
  - explain how to get an honest sense of how well a model is doing
    - explain train/test splits
    - state the purposes of Cross Validation
    - explain k in “k-fold cross validation”
  - describe the sources of model error
    - describe overfitting and underfitting
  - explain how to compare models and select the best one

We want to <u>**Do The Best**</u>™

- How can we *evaluate* how well we're doing?
  - ideally with real numbers
  - ideally in a comparable way
- How can we *change* how well we're doing?
  - ideally in a directed, intentional way

How can we evaluate how well we're doing?

- Story:
  - Little Sophie™ learns multiplication

How can we evaluate how well we're doing?

- Memorization vs Generalization
- We usually care about generalization
  - (Otherwise use a database)
  - Eg: predicting stock market data, predicting heart disease
    - we care about "unseen" data points

How can we evaluate how well we're doing?

- Story:
  - Sophie does well on a test she hasn't seen

How can we evaluate how well we're doing?

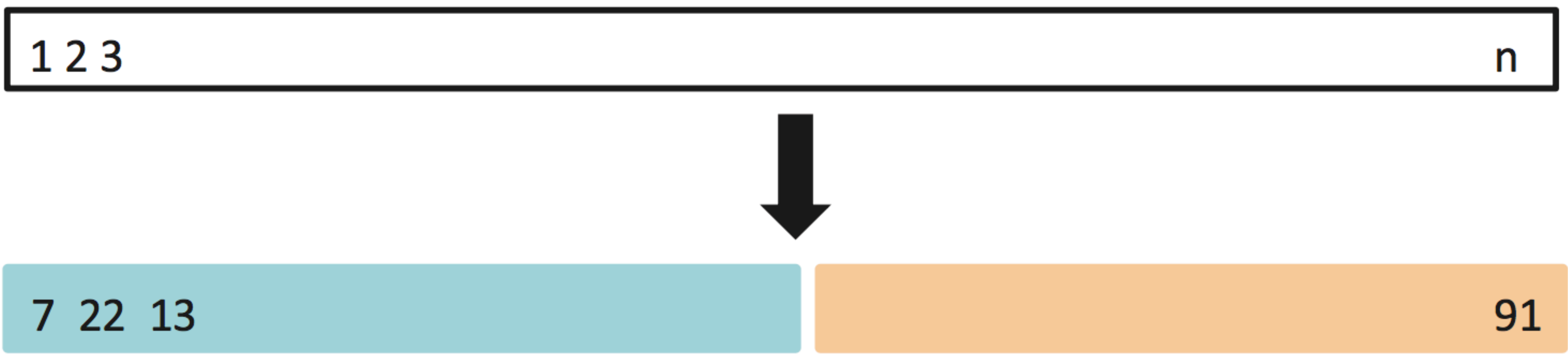
- We can set aside some data points to be "unseen" so that, using them, we can evaluate our ability to generalize (rather than memorize)
- Train/Test split
  - How much in each?
    - More in Train: better model, less able to evaluate it well
    - More in Test: worse model, better able to evaluate it well

How can we evaluate how well we're doing?

- We could keep the data all together (no!)
- We could split the data evenly in two equal-sized groups
- We could split the data into n groups (each data point in its own group)
- We could split the data into k << n groups

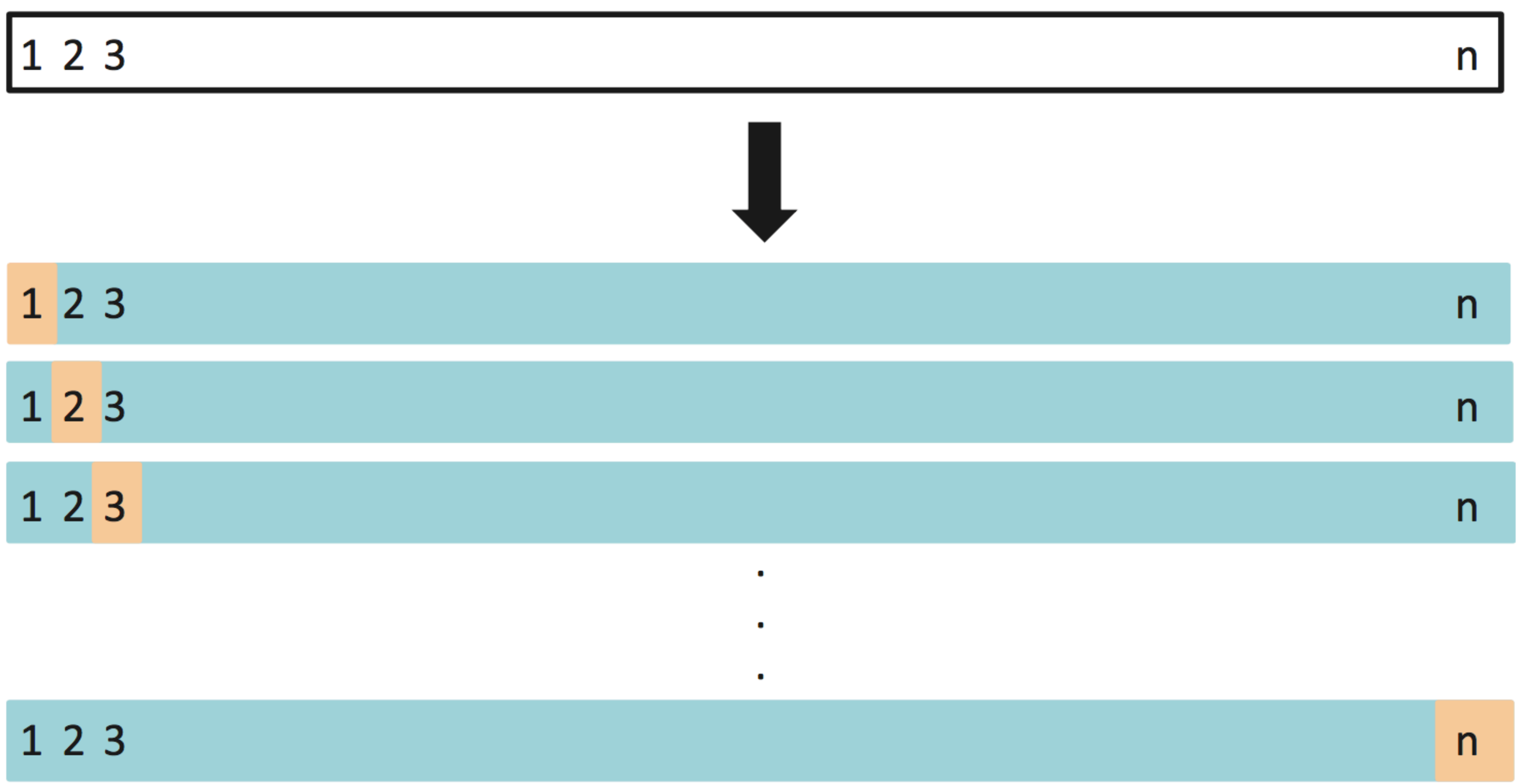
How can we evaluate how well we’re doing?

- We could split the data evenly in two equal-sized groups



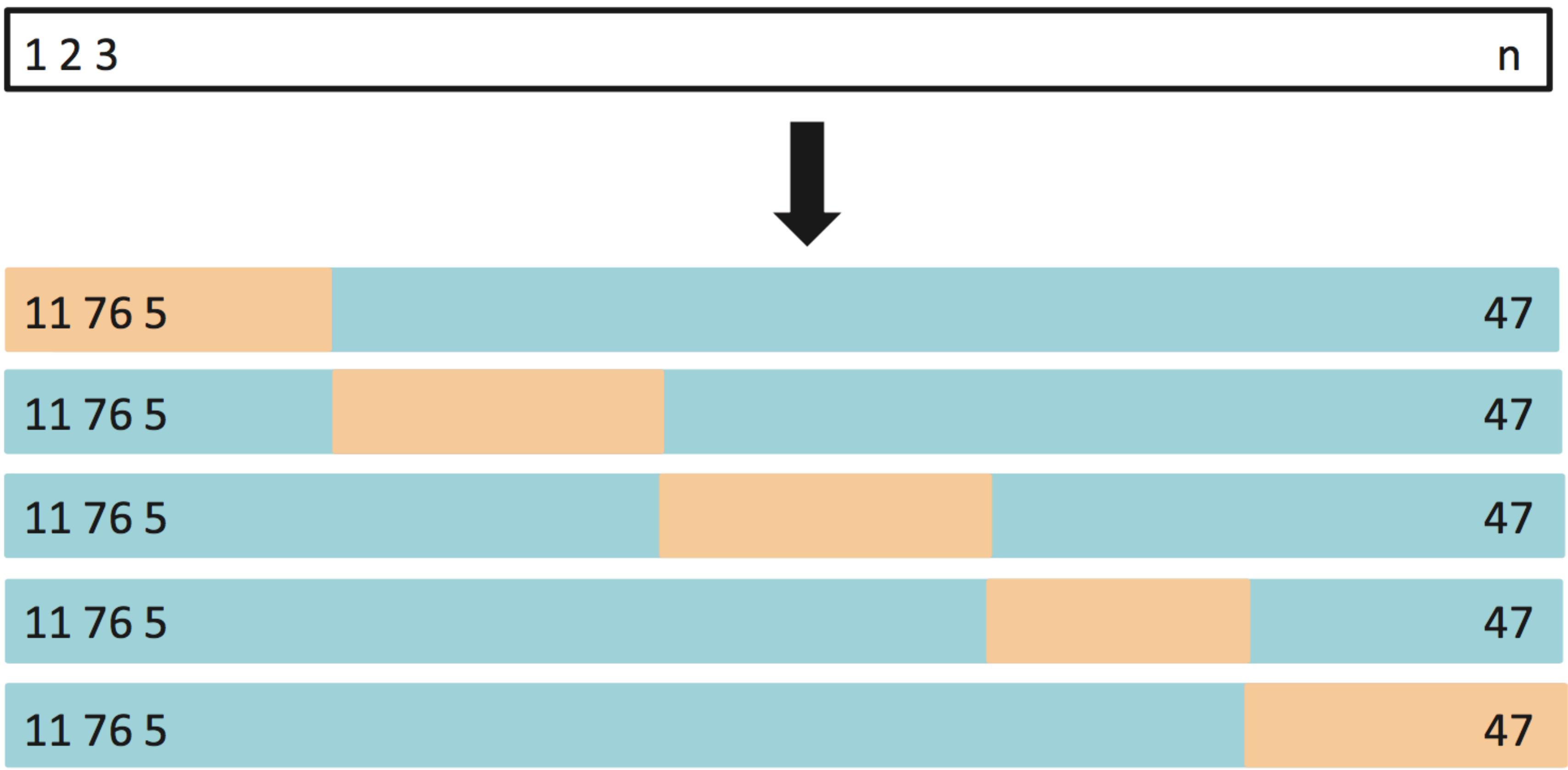
How can we evaluate how well we're doing?

- We could split the data into n groups (each data point in its own group)



How can we evaluate how well we're doing?

- We could split the data into k << n groups



How can we evaluate how well we're doing?

- K Fold Cross Validation:
  - Split dataset D into subsets, D_1, D_2, D_3, … D_k
  - errors_list = []
  - For each subset D_i
    - Mark D_i as the test set
    - Train a model on data in all subsets != D_i
    - errors_list.append(Evaluate model on D_i)
  - Report error = mean(errors_list)

How can we evaluate how well we're doing?

- Story:
  - Sophie consistently do well on unseen data

How can we change how well we're doing?

- Story:
  - Sophie notices she does better with hot chocolate

How can we change how well we're doing?

- Try different parameters (broadly speaking)
  - feature presence
    - what do we capture? do we have an 'is_married' feature?
  - feature encoding
    - do we encode age as int? float? categorical?
  - feature degree
    - should a feature interact with itself? x_0^2
    - should a feature interact with another? $x_0 * x_{15}$
    - how many features should we have?

How can we change how well we're doing?

- Let's focus on degree

- $y = \beta_0 + \beta_1 x_1$
- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2$
- $y = \beta_0 + \beta_1 x_1 + \beta_1 x_1^2 + \beta_2 x_1^3 + \ldots$

- How do we choose?

How can we change how well we’re doing?

- What’s the best degree?
- It’s not a parameter, exactly, it’s a… Hyperparameter!
  - Models usually optimize parameters internally
    - To, say, minimize error on the training set
  - We usually optimize hyperparameters with CV

How can we change how well we're doing?

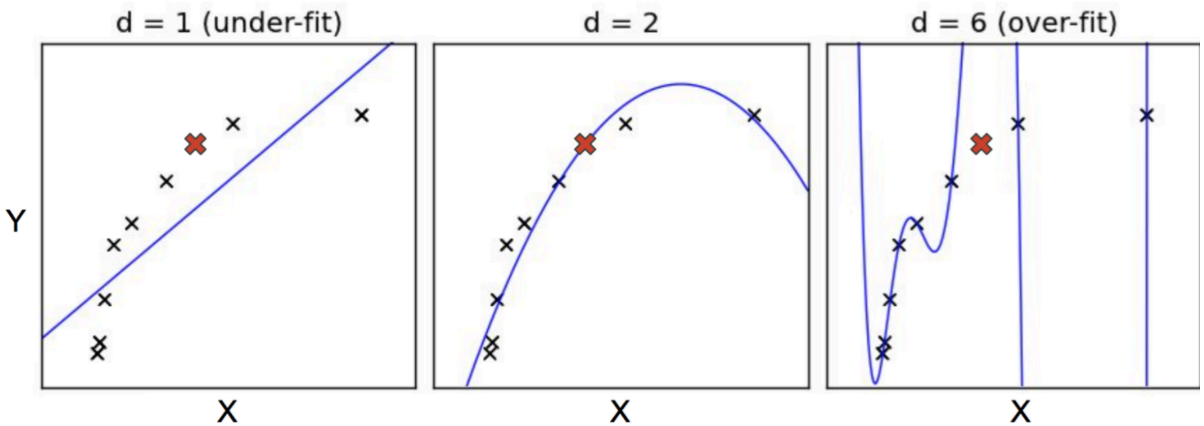
- errors = {}
- For each value *v* of the hyperparameter *h* we want to test…
  - Run a k-fold CV where $h=h_i$ (for every model in every fold). Let the mean of the errors be *e*
  - errors[v] = *e*
- Pick the dictionary key *v* w the lowest value *e*
- How well does our model work?
  - How does it perform on unseen data?
  - What data is unseen?

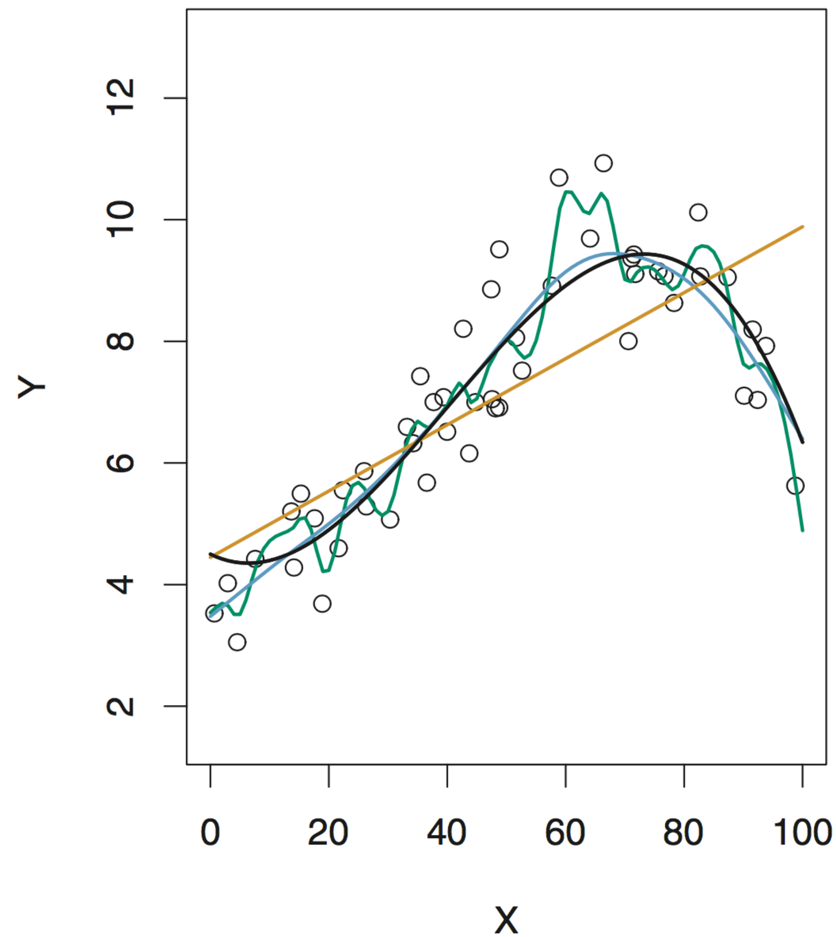
How can we change how well we're doing?

- If making decisions like choosing hyperparameters based on the results of CV runs…
  - Those hyperparameters are chosen to minimize error over all the data they can see (data used within the CV folds)
  - Just as regular-parameters are chosen to minimize error over all the data they can see (data used within a single model)
- Therefore we need a *new*, *completely unseen* set of data on which to evaluate our model's performance
  - We train a model with the optimally-chosen hyperparameter, using all data from our CV folds as training data
  - We use a *new*, *only-just-now-for-the-very-first-time-seen* set of held-out data as our test data
- This means that, before we begin any model creation, if we anticipate having hyperparameters (almost always), and want to ultimately be able to get an honest sense of how we're doing (almost always), we need to set aside a set of data that we don't use until the very end.
  - What we previously referred to as merely Train/Test becomes in the hyperparameter world:
  - Train, Validation, Test

How can we change how well we're doing?

Hyperparameter: degree of polynomial



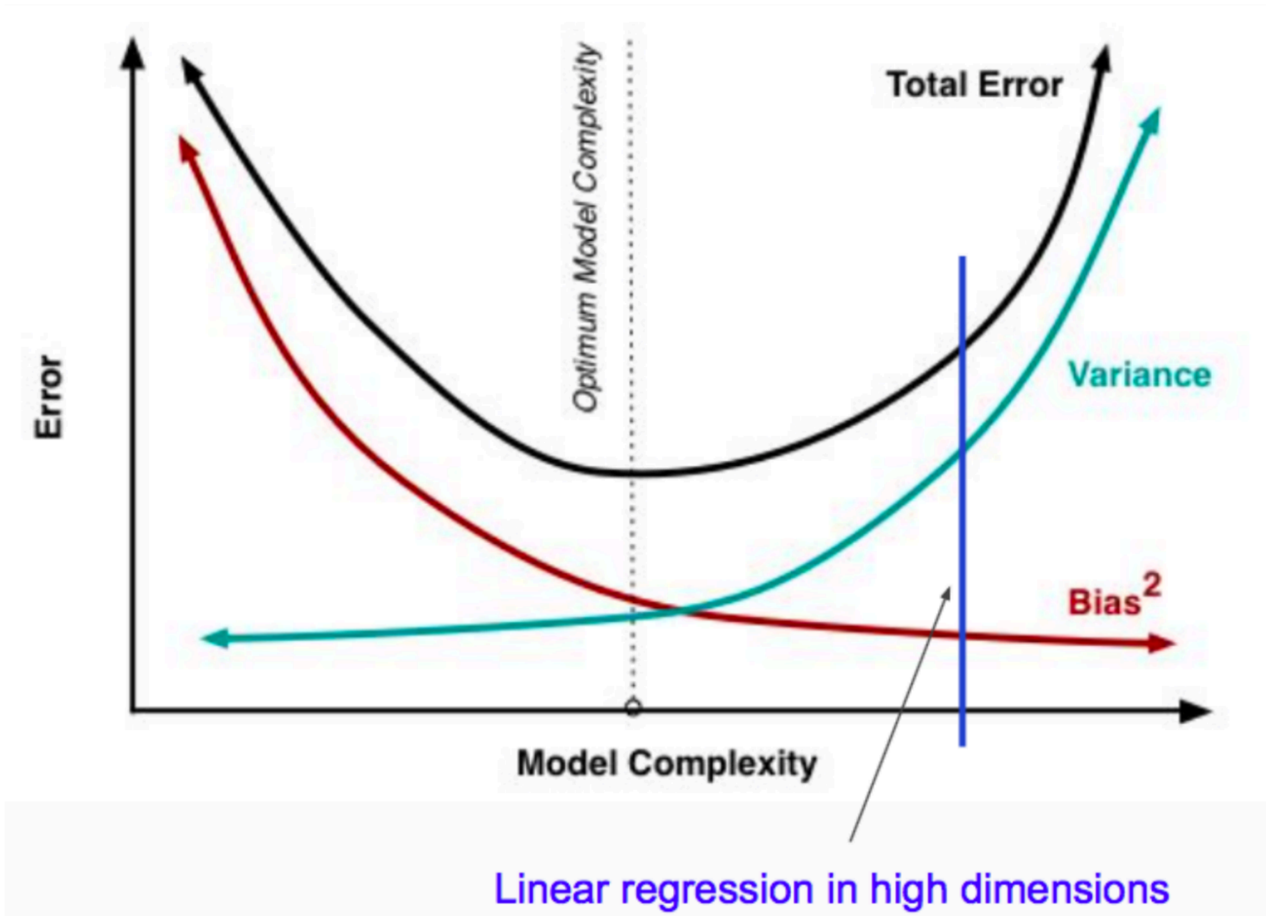


$y = \beta_0 + \beta_1 x_1$

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2$

$y = \beta_0 + \beta_1 x_1 + \beta_1 x_1^2 + \beta_2 x_1^3 + \ldots$

Bias Variance Tradeoff



$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon)$$

expected test MSE

variance: amount f() would change if trained on a different dataset

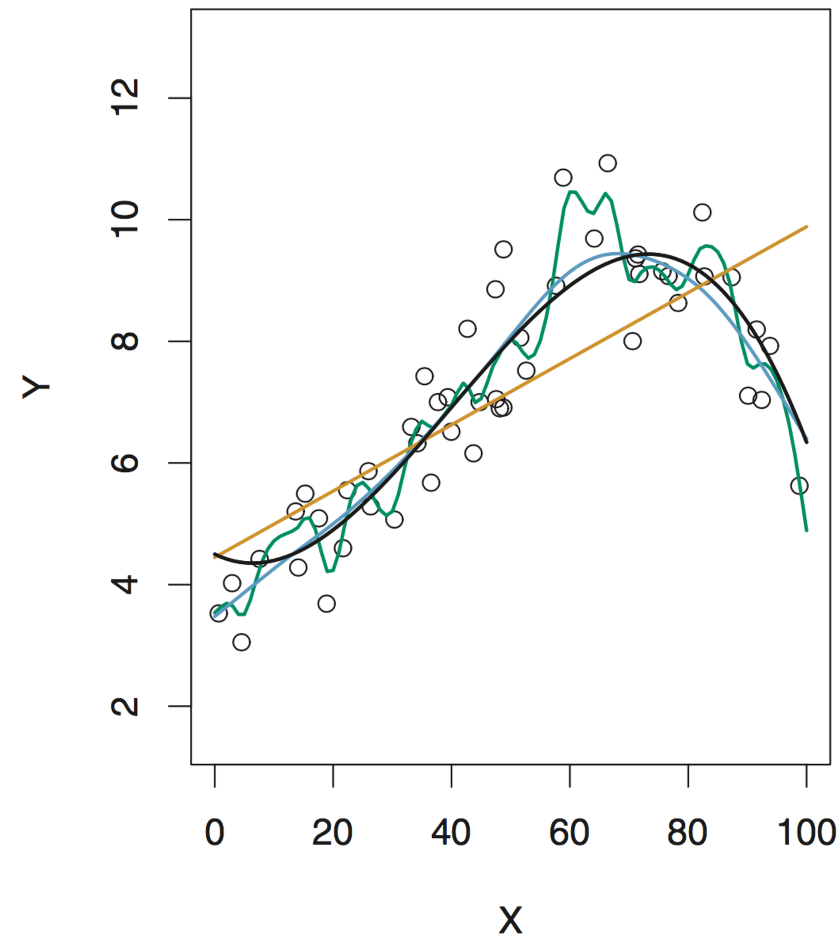
bias: error due to simplified approximation

irreducible error

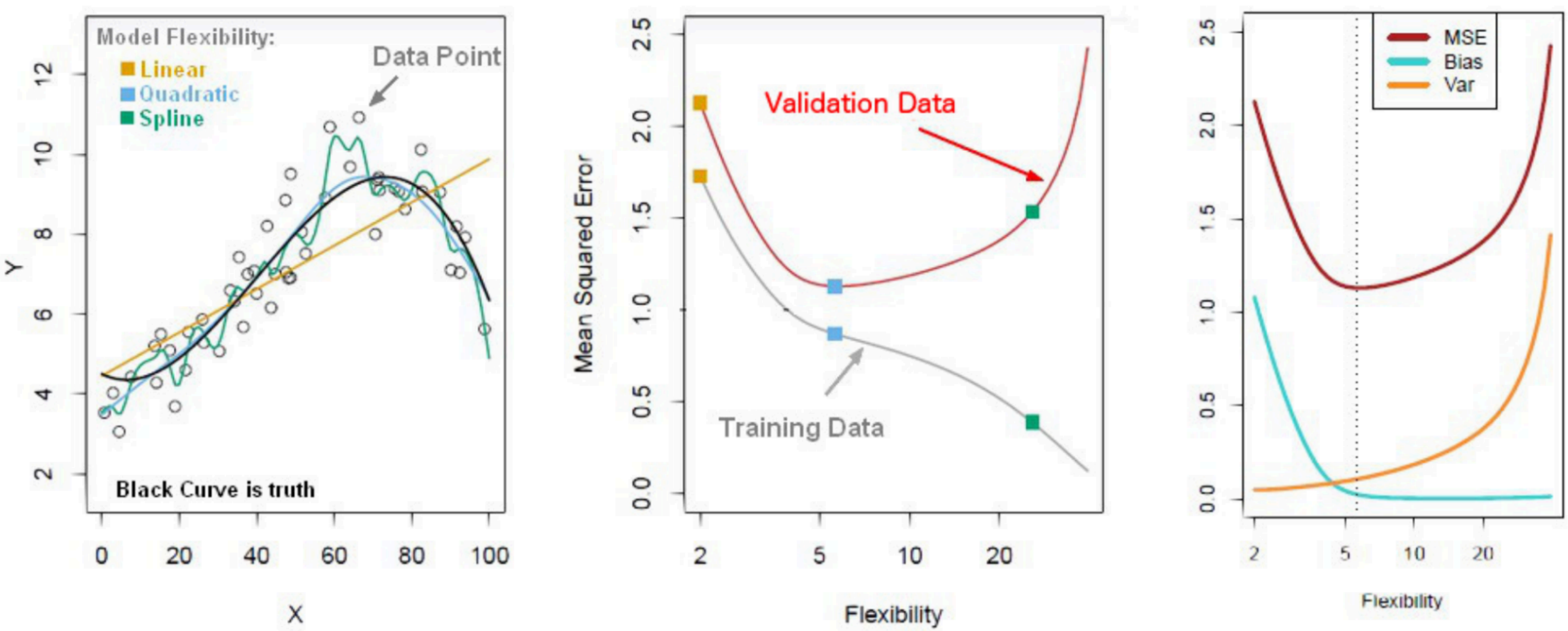
How can we change how well we're doing?

Hyperparameter: degree of polynomial

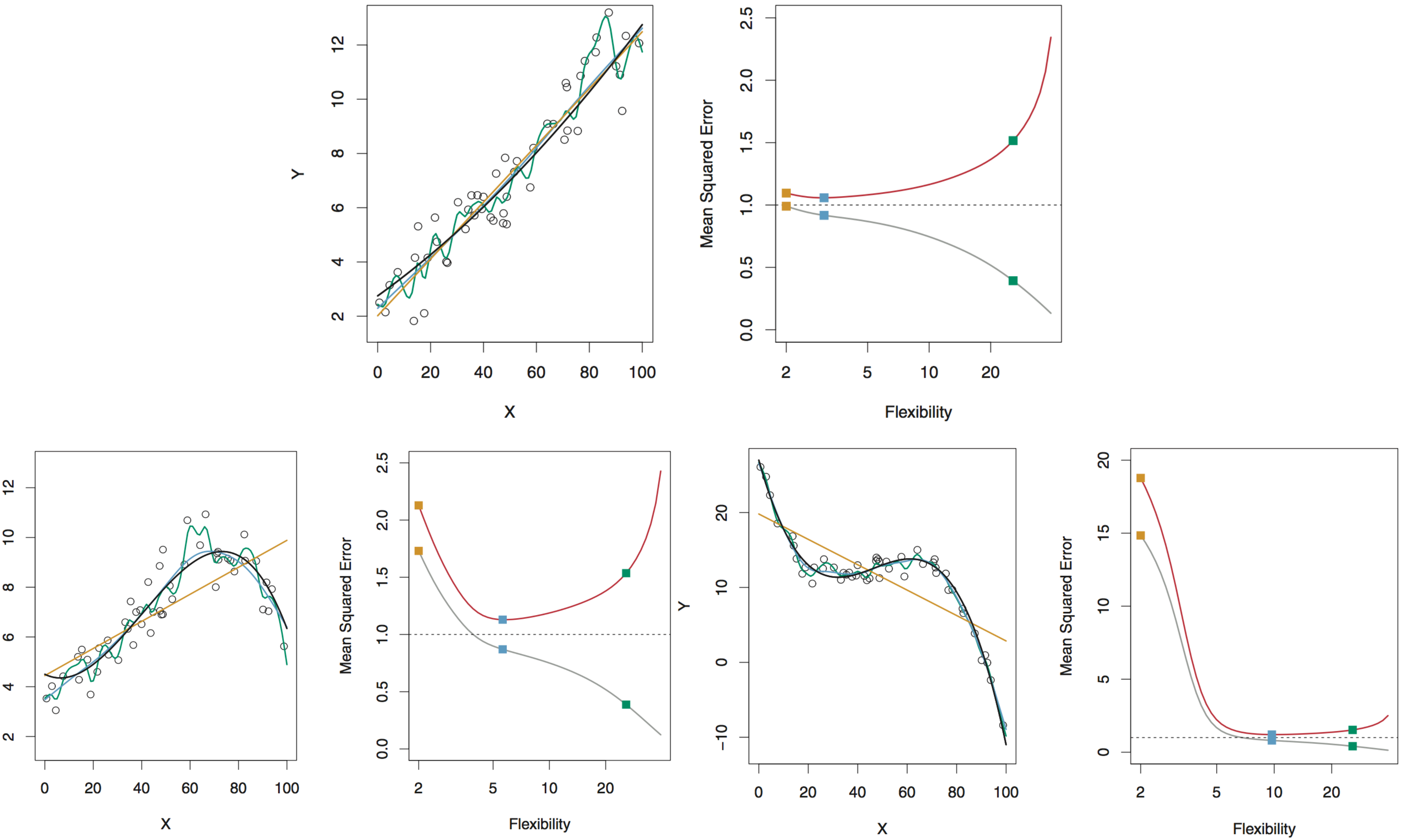
- Underfitting
  - Model does not fully capture the signal in X
  - Insufficiently flexible model
- Overfitting
  - Model erroneously interprets noise as signal
  - Overly flexible model



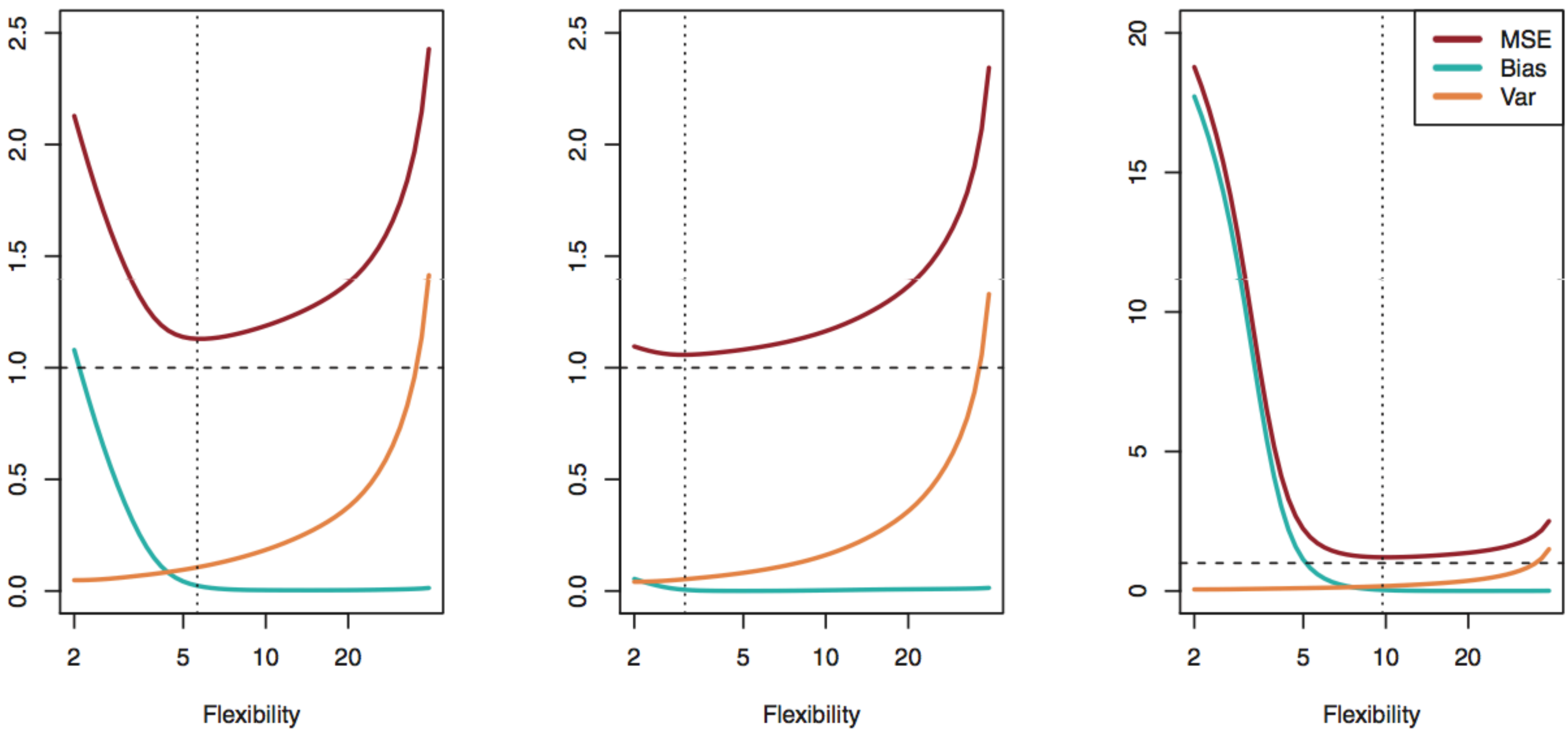
Bias Variance Tradeoff



Bias Variance Tradeoff



Bias Variance Tradeoff



Caveat: this is pretty contentious since it looks a lot like precision and accuracy, which it's not intended to be

Bias Variance Tradeoff