CS/ENGR M148 L6:
Regularization

Sandra Batista

Administrative News

This week in discussion section:

Lab on regression and cross validation

Project Data Check-in: Your team will need to demonstrate a regression model on your project data.

Join our slido for the week…

https://app.sli.do/event/fCYNaz1LPznfsUF8YhGeqo

Today's Learning Objectives

Students will be able to:

- Review: Identify **overfitting** in terms of **bias** and **variance**
- Apply **model selection** for **overfitting**
- Understand the need for **cross validation** to address overfitting
- Apply **cross validation** and **regularization** to address overfitting

R-squared

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y} - y_i)^2}$$

- *This will be 0 if model is as good as mean*
- *This will be 1 if model is perfect.* ***Should we be concerned?***
- *This can be negative if the model is worst than the average. This can happen when we evaluate the model on the test set.*

Solving the least squares problem

- Method: Linear algebra.

$$y = X\beta + \epsilon.$$

- Find the value β that minimizes $(\|X\beta - Y\|_2)^2$; the minimal β is denoted $\hat{\beta}$.

- Using orthogonality, the solution emerges naturally as the normal equation: $(X^TX)^{-1}X^TY$.

^

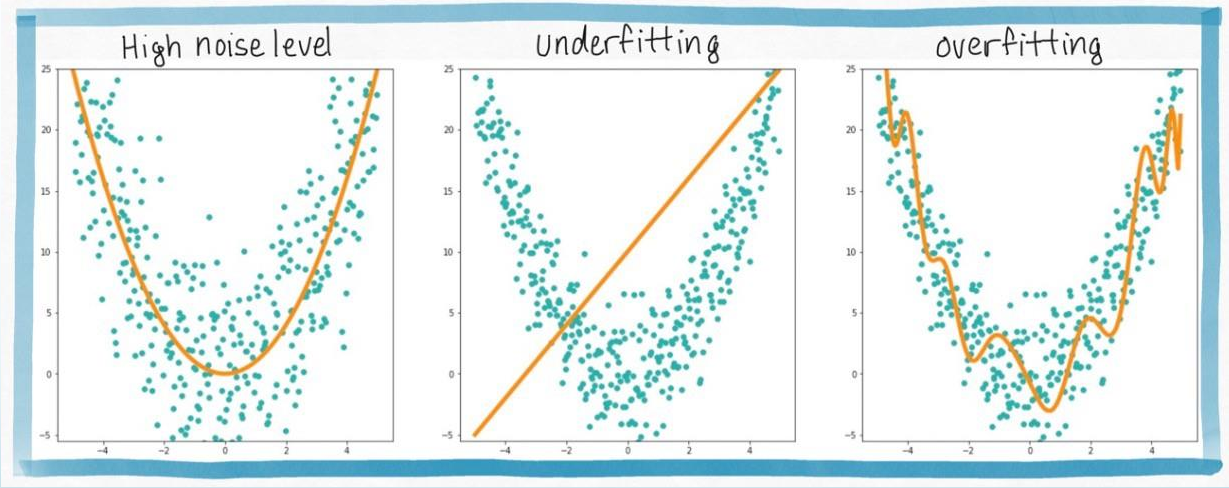
- Overdetermined: more rows than columns – common for OLS
- Underdetermined: more columns than rows – infinitely many or no solutions

Test Error and Generalization

We know to evaluate models on both train and test data because models can do well on training data but do poorly on new data.

When models do well on new data is called generalization.

There are at least three ways a model can have a high test error.



Irreducible and Reducible Errors

We distinguished the contributions of noise to the generalization error:

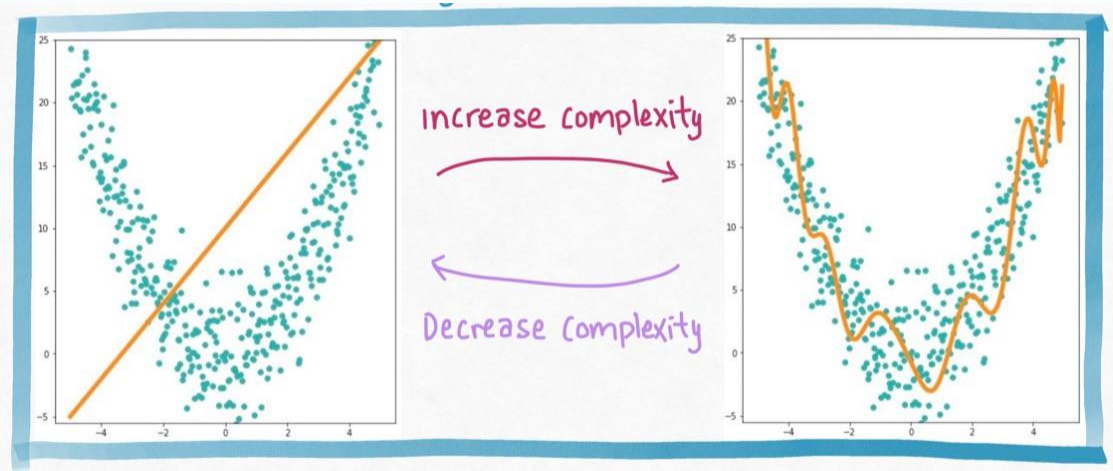
Irreducible error: we can't do anything to decrease error due to noise.

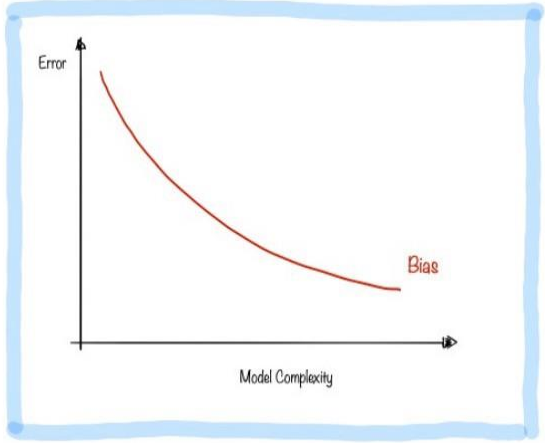
$$y = X\beta + \epsilon.$$

Reducible error: we can decrease error due to overfitting and underfitting by improving the model.

Underfitting and Overfitting

*Reducible error comes from either **underfitting** or **overfitting**. There is a trade-off between the two sources of errors:*

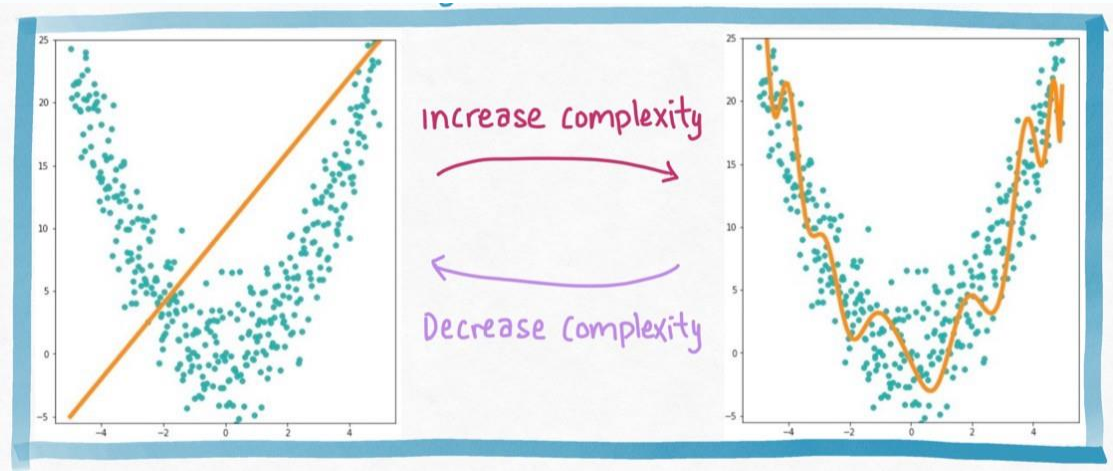


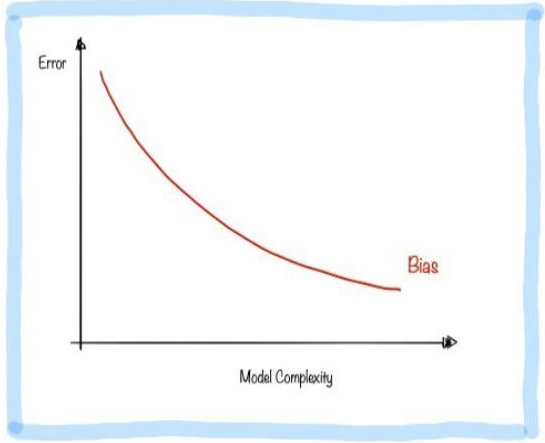


Underfitting and Overfitting

Underfitting *is when a model performs poorly on the training and testing data sets*

Overfitting *is when model performs well on training data set, but poorly on the testing data set.*

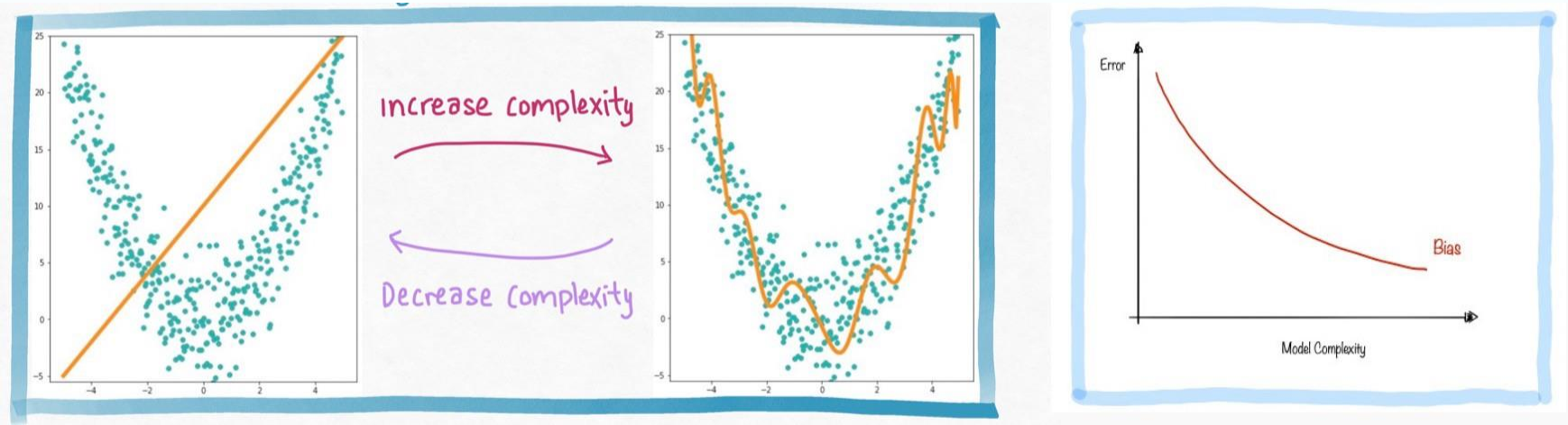




Bias Variance Tradeoff

Underfitting *occurs when there is* ***high bias***

Overfitting *occurs when there is* ***high variance***



Bias

Bias is the distance between expected value of estimator and parameter we want to estimate.

$Bias(\widehat{f(x)}) = E[\widehat{f(x)}]$ - *f(x)*

How do we actually calculate this?
Honestly, we can't. Why?

Ideally, we'd want many training data sets and a separate testing data set.

In practice, ***bootstrap training data sets*** *and use testing data set.*

Bootstrap

Bootstrapping *is the practice of sampling from the observed data* (X,Y) *in estimating statistical properties.*

Bootstrap

We pick a ball and replicate it and move it to the other bucket. This is ***sampling with replacement.***

Bootstrap

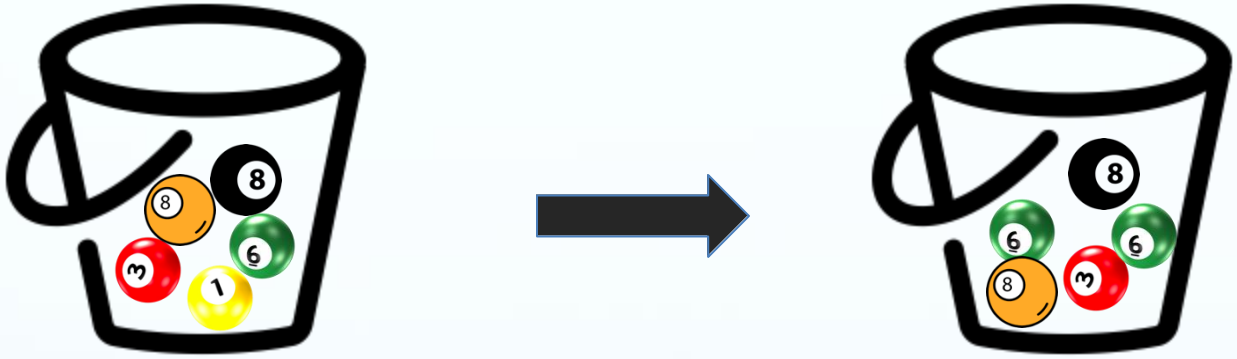
We then randomly pick another ball and again we replicate it.
As before, we move the replicated ball to the other bucket.

Bootstrap

We repeat this process.

Bootstrap

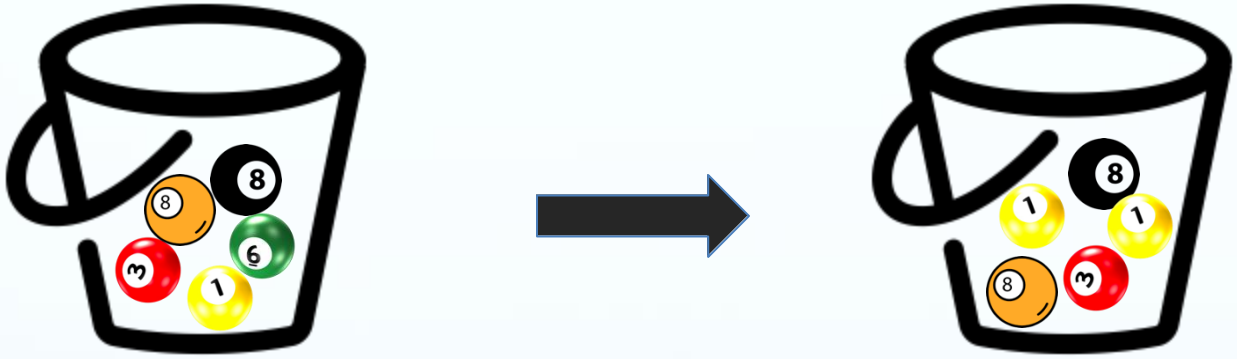
We continue until the "other" bucket has ***the same number of balls*** *as the original one.*



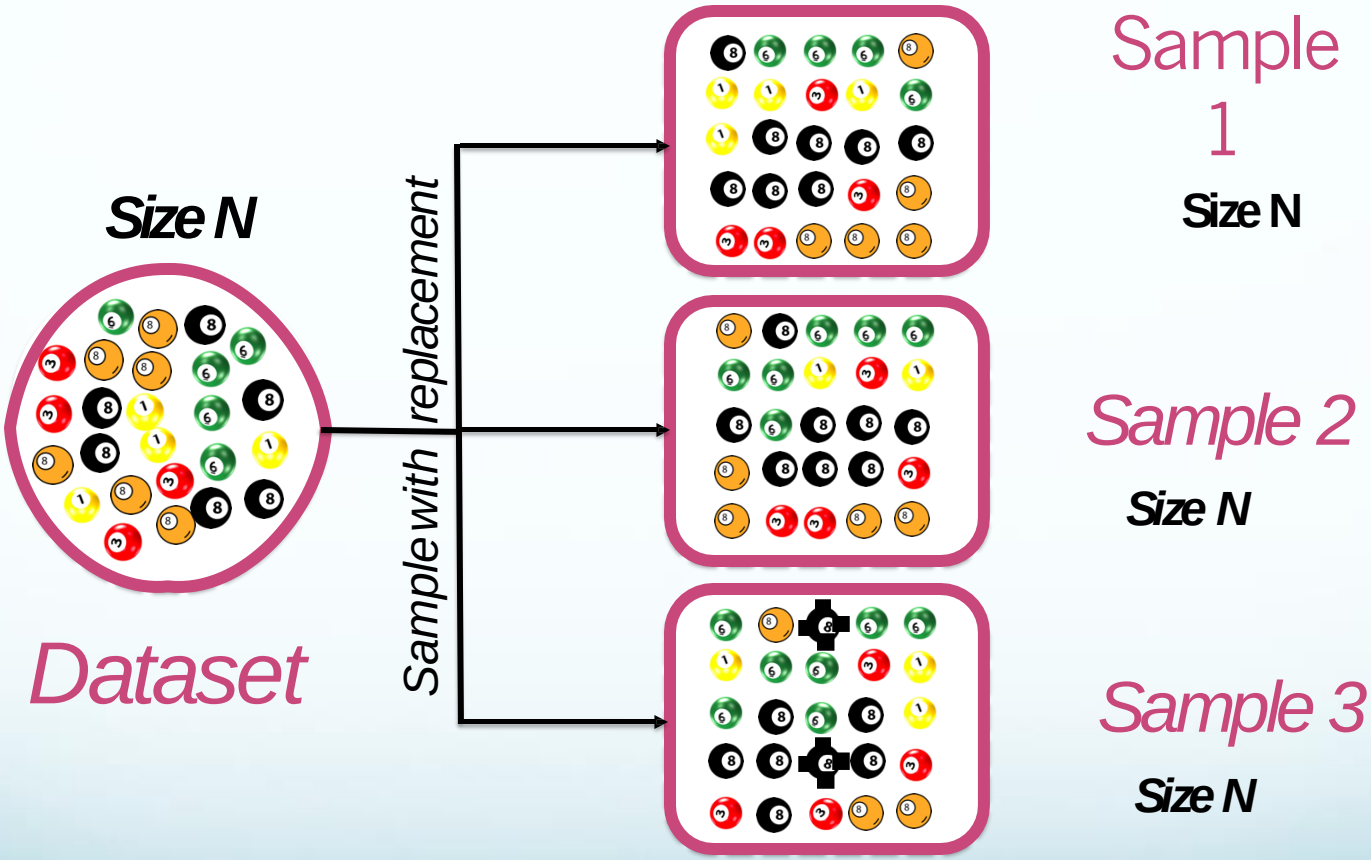
This new bucket represents a new sample

Bootstrap

We repeat the same process and acquire another sample.



Bootstrap


Size N
Dataset
Sample with replacement
Sample 1
Size N
Sample 2
Size N
Sample 3
Size N

Bias

Bias is the distance between expected value of estimator and parameter we want to estimate.

$Bias(\widehat{f(x)}) = E[\widehat{f(x)}]$ - *f(x)*

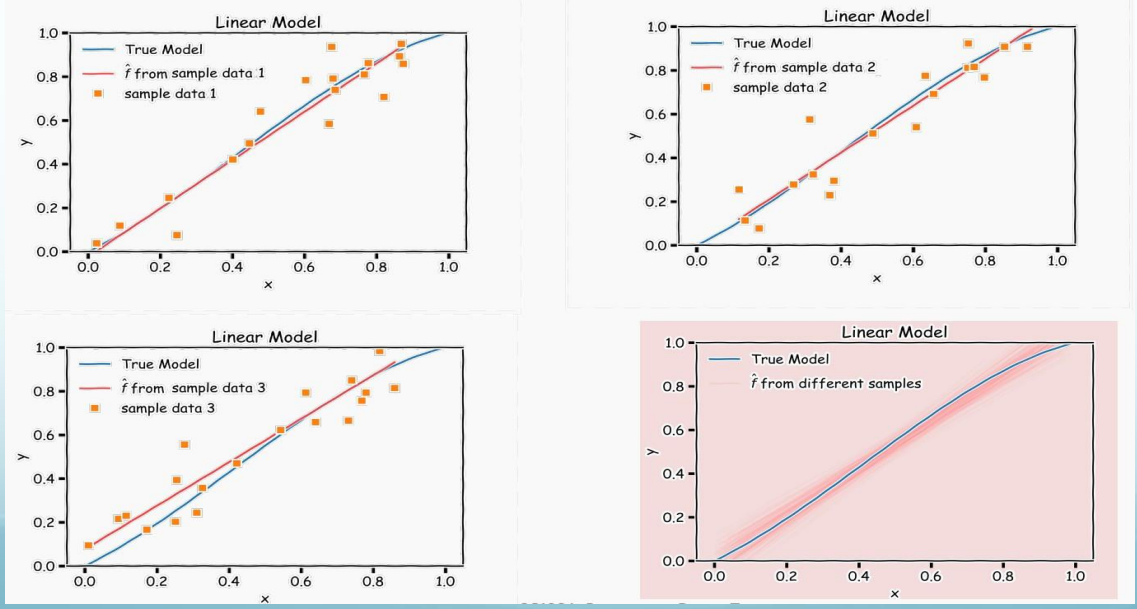
How do we estimate bias in practice?

.

Variance

Variance measures how the predictions will vary over training sets. ***Notice expectation is over training sets…***

$$Var(\widehat{f(x)}) = E[(\widehat{f(x)} - E[\widehat{f(x)}])^2]$$

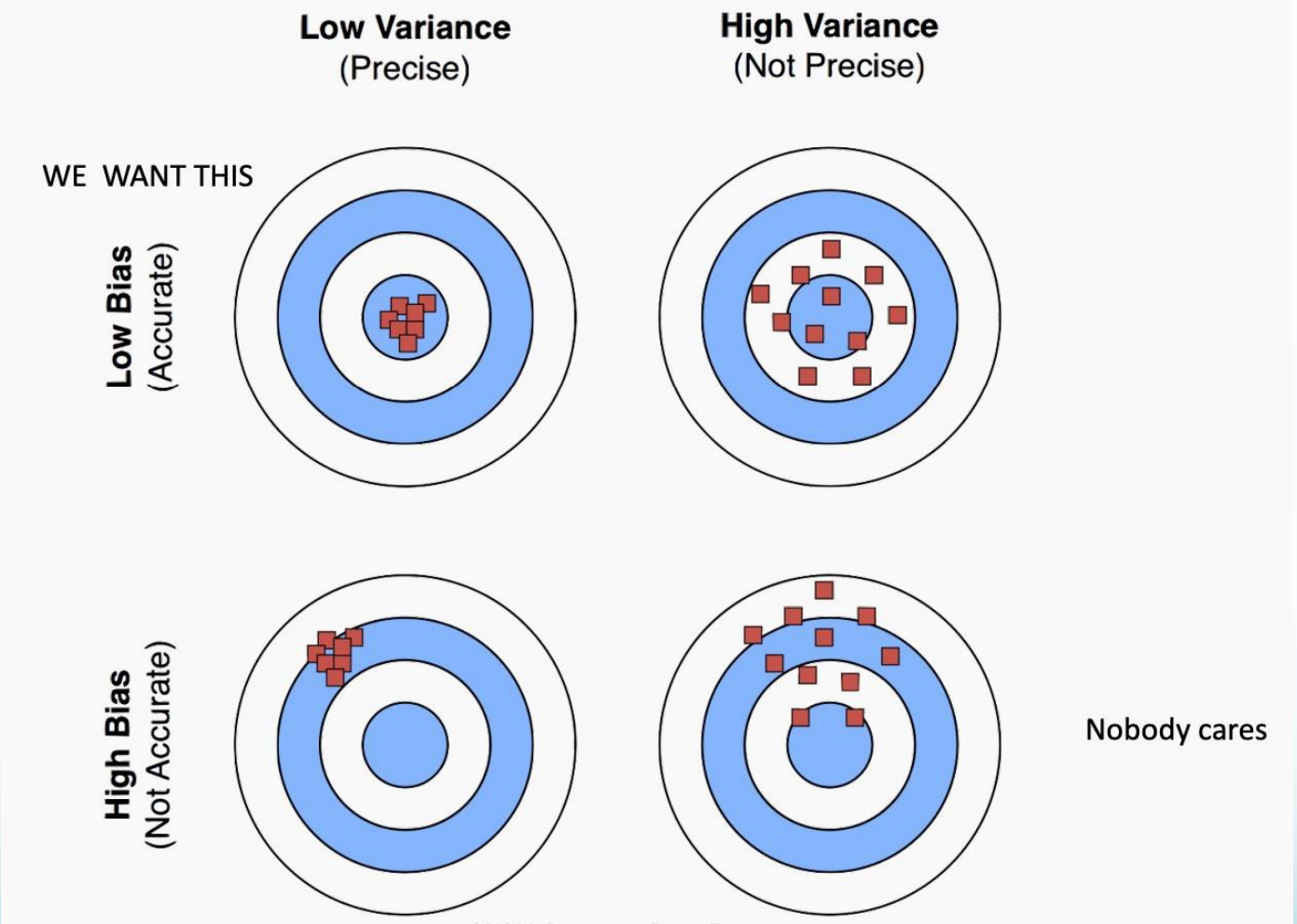


Variance

Variance measures how the predictions will vary over training sets. ***Notice expectation is over training sets…***

$$Var(\widehat{f(x)}) = E[(\widehat{f(x)} - E[\widehat{f(x)}])^2]$$

How do we estimate this in practice?


Low Variance
(Precise)
High Variance
(Not Precise)
WE WANT THIS
Low Bias
(Accurate)
High Bias
(Not Accurate)
Nobody cares

Your turn: Examining Bias and Variance

Please get the Jupyter notebook

Go to:
https://colab.research.google.com/drive/12z5-caCx9wWoGzRJEuDBkHwEdGSfq_mA?usp=sharing

[Yu, Barter 2024]

Today's Learning Objectives

Students will be able to:

- ✓ Review: Identify **overfitting** in terms of **bias** and **variance**
- ✗ Apply **model selection** for **overfitting**
- ✗ Understand the need for **cross validation** to address overfitting
- ✗ Apply **cross validation** and **regularization** to address overfitting

Overfitting

Overfitting occurs when a model corresponds too closely to the training set, and as a result, the model fails to fit additional data.

So far, we have seen that overfitting can happen when:

- *Too many parameters*
- *Degree of the polynomial is too large*
- *Too many interaction terms*
- *Number of samples used in training or validating*

Overfitting

Overfitting occurs when a model corresponds too closely to the training set, and as a result, the model fails to fit additional data.

Ways to address:

1. *Model selection: Limiting the number of parameters in model*
2. *Using more validation data sets*

Next, we will see other evidence of overfitting, which will point to a way of avoiding overfitting: Ridge and Lasso regressions.

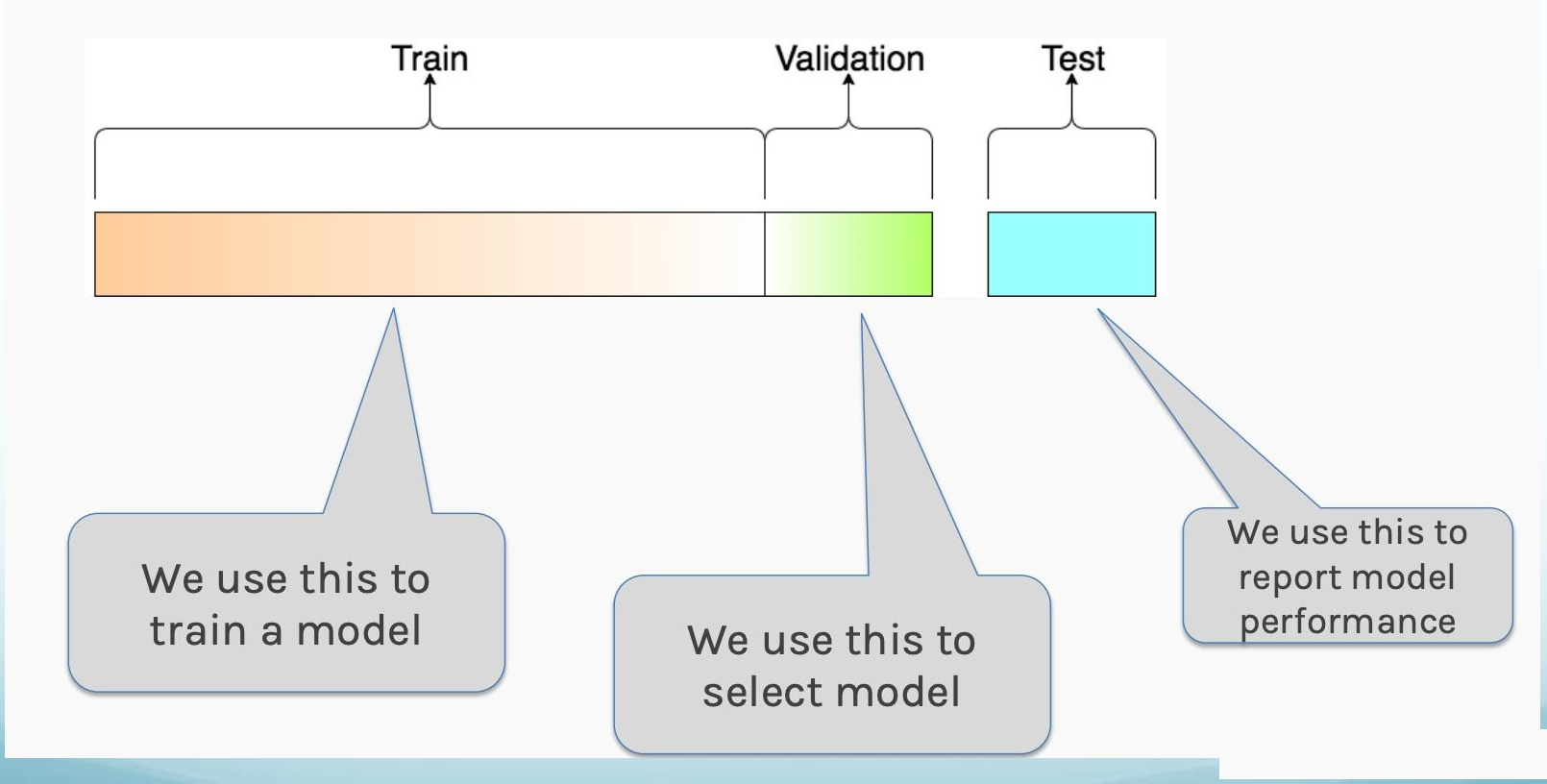
Generalization Error

We know to evaluate the model on both train and test data, because models that do well on training data may do poorly on new data (overfitting).

The ability of models to do well on new data is called generalization.

The goal of model selection is to choose the model that generalizes the best.

Train-Validation-Test



Model Selection

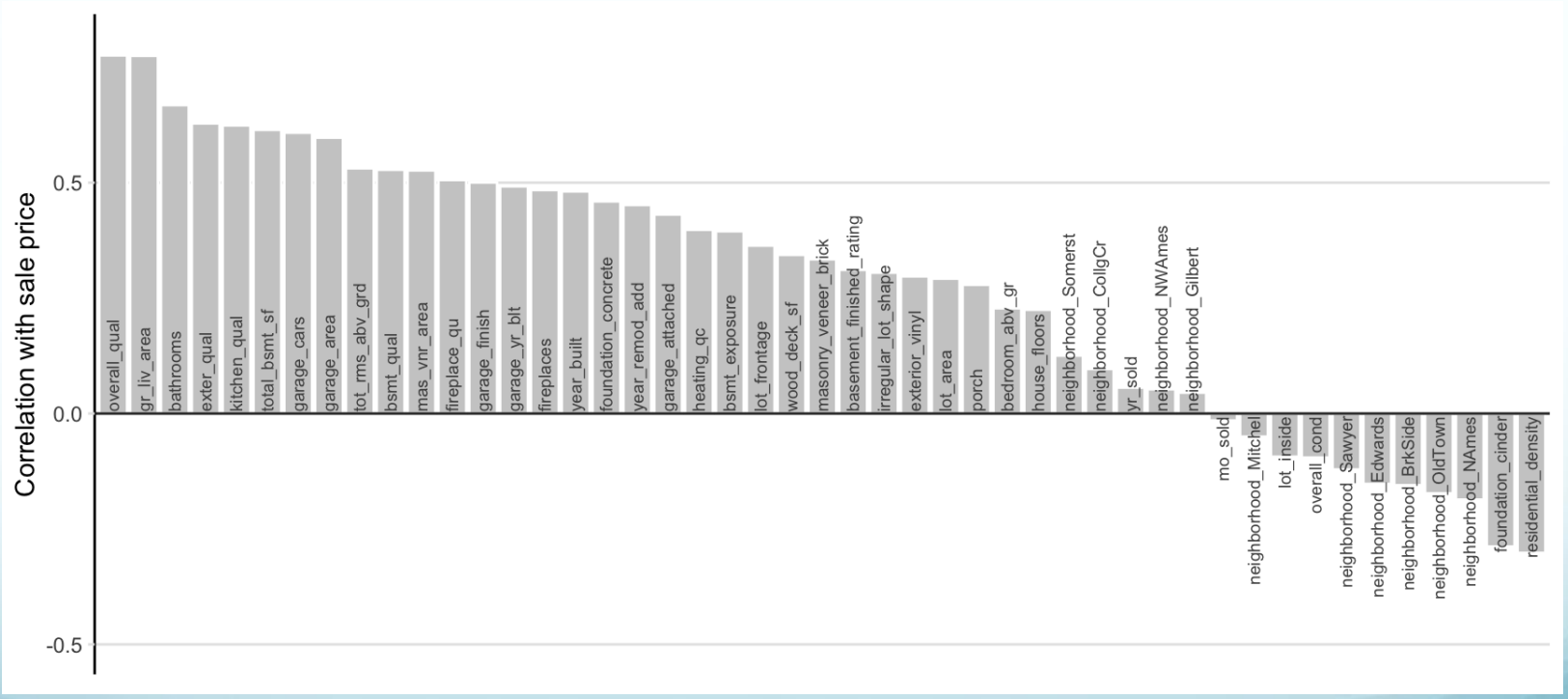
Question: How many different models when considering J predictors (only linear terms) do we have?

The combinatorial problem space is exponential in the number of predictors.

However, we have a few approaches that are quadratic in the number of predictors.

EDA for house prices

Correlation screening: keep only the predictive features that are most correlated with the response



[Yu, Barter 2024]

Stepwise Variable Selection and Cross Validation

Selecting optimal subsets of predictors (including choosing the degree of polynomial models) through:

- *stepwise variable selection - iteratively building an optimal subset of predictors by optimizing a fixed model evaluation metric each time.*
- *Stepwise variable selection is a **greedy** algorithm that can be done* **forwards** or **backwards**.
- *validation - selecting an optimal model by evaluating each model on validation set.*

Stepwise Variable Selection: Forward method

- *Start with the empty set of selected predictors, P, and empty model, M. We have J predictors in the model*
- *Let best_metric such as MSE be set*
- *Let k be our desired number of predictors*
- *For i = 1, …, k:*
- *best_feature_found = None.*
  - *For j = 1 to J:*
  - *if feature j is not already selected,*
  - *create C = list of features already in the M and feature j*
  - *Check metric for model with features in C*
  - *If metric for model with feature J is better than best_metric:*
  - *update best metric and best feature found*

Stepwise Variable Selection: Forward method

- *If no best_feature is found in the current round,*
- *break from the algorithm*
- *Otherwise:*
  - *Add the best feature to selected predictors P and M is model containing all selected predictors*

After k rounds, k variables will be selected

Stepwise Variable Selection: Backward method

- *The backwards method starts with all the predictors.*

Then on each iteration it removes the predictor that ***causes the least performance loss.***

Your turn:
Let's Look at Forward Selection

Please get the Jupyter notebook

Go to:
https://colab.research.google.com/drive/12z5-caCx9wWoGzRJEuDBkHwEdGSfq_mA?usp=sharing

Save a copy to your Google Drive and keep notes there...

[Yu, Barter 2024]

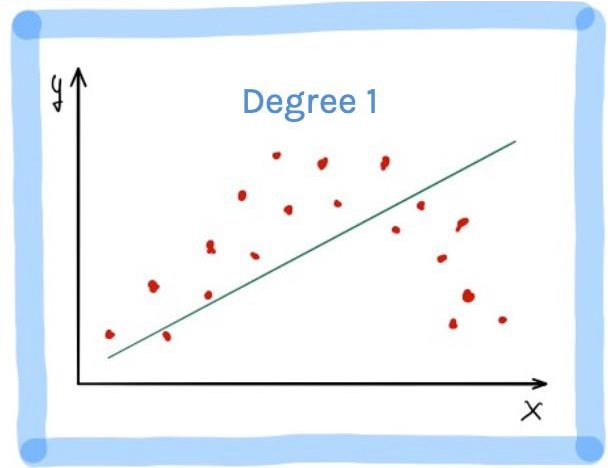
Today's Learning Objectives

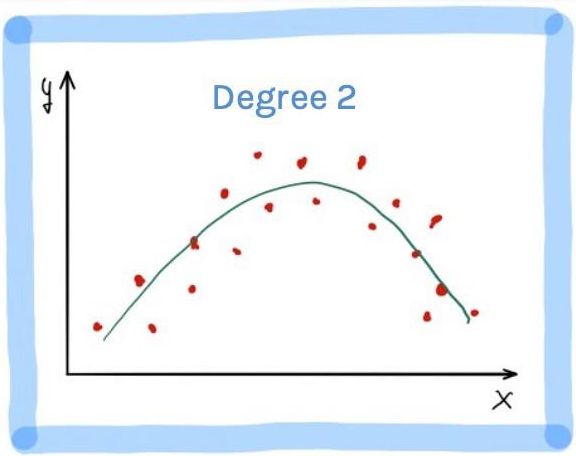
Students will be able to:

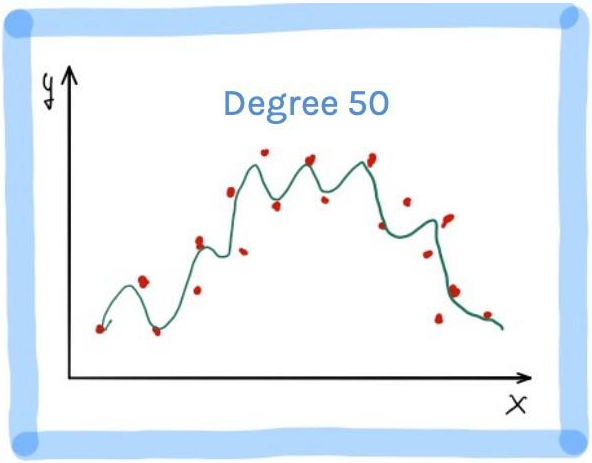
- ✓ Review: Identify **overfitting** in terms of **bias** and **variance**
- ✓ Apply **model selection** for **overfitting**
- ✗ Understand the need for **cross validation** to address overfitting
- ✗ Apply **cross validation** and **regularization** to address overfitting

Choosing the degree of the polynomial model

Fitting a polynomial model requires choosing a degree.





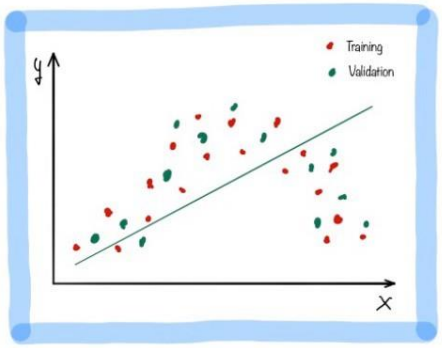


Underfitting: when the degree is too low, the model cannot fit the trend.

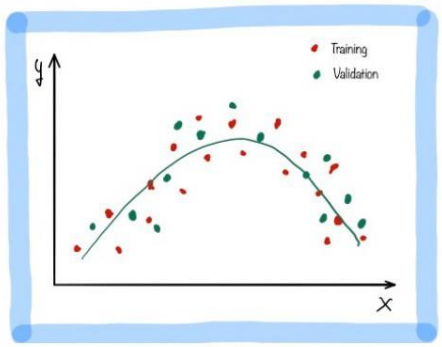
We want a model that fits the trend and ignores the noise.

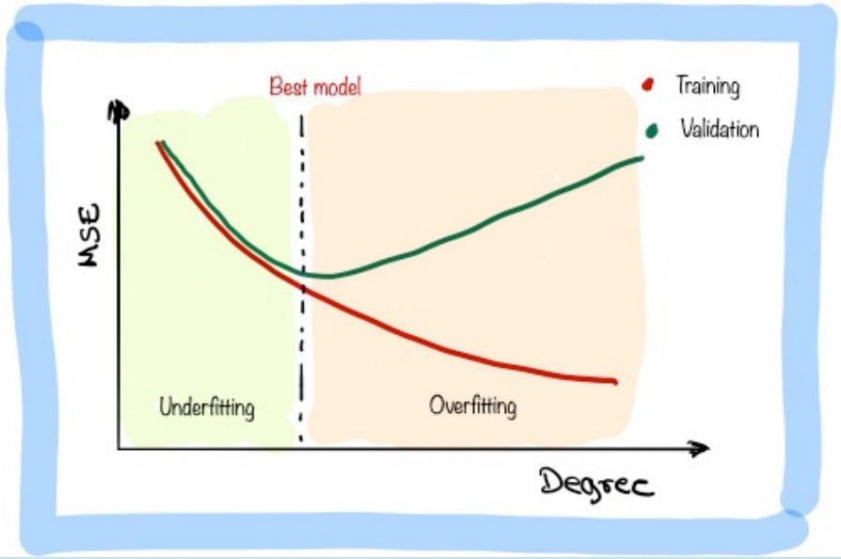
Overfitting: when the degree is too high, the model fits all the noisy data points.

Underfitting: train and validation error is high.

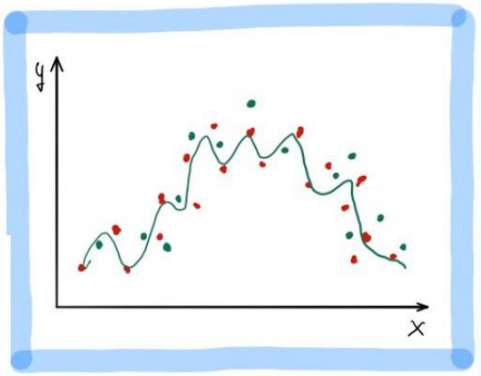


Best model: validation error is minimum.



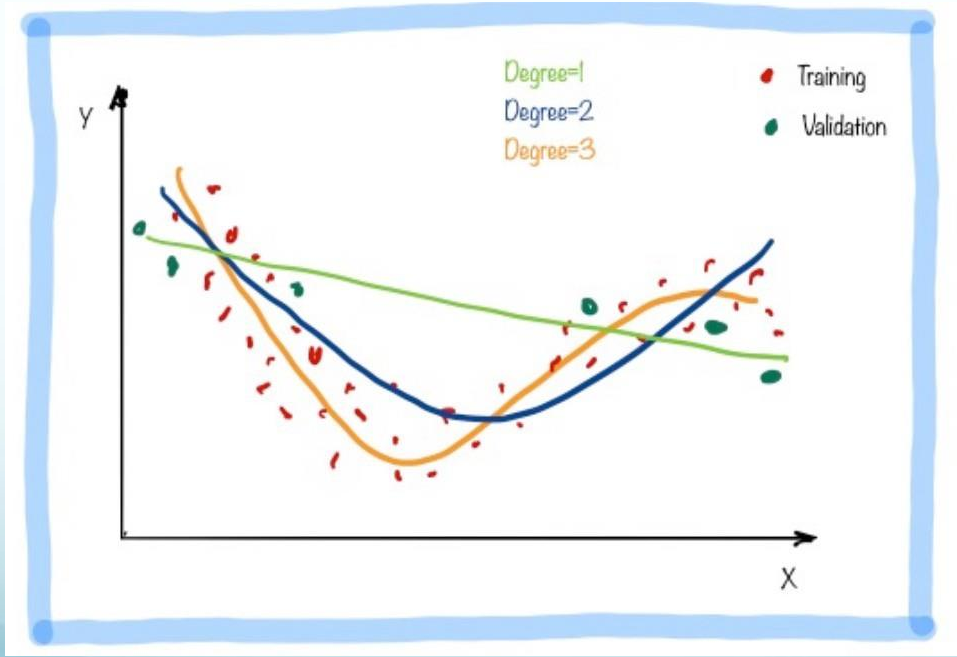


Overfitting: train error is low, validation error is high.



Cross Validation: Motivation

Using a single validation for comparing models- ***there is the possibility of overfitting to the validation set***



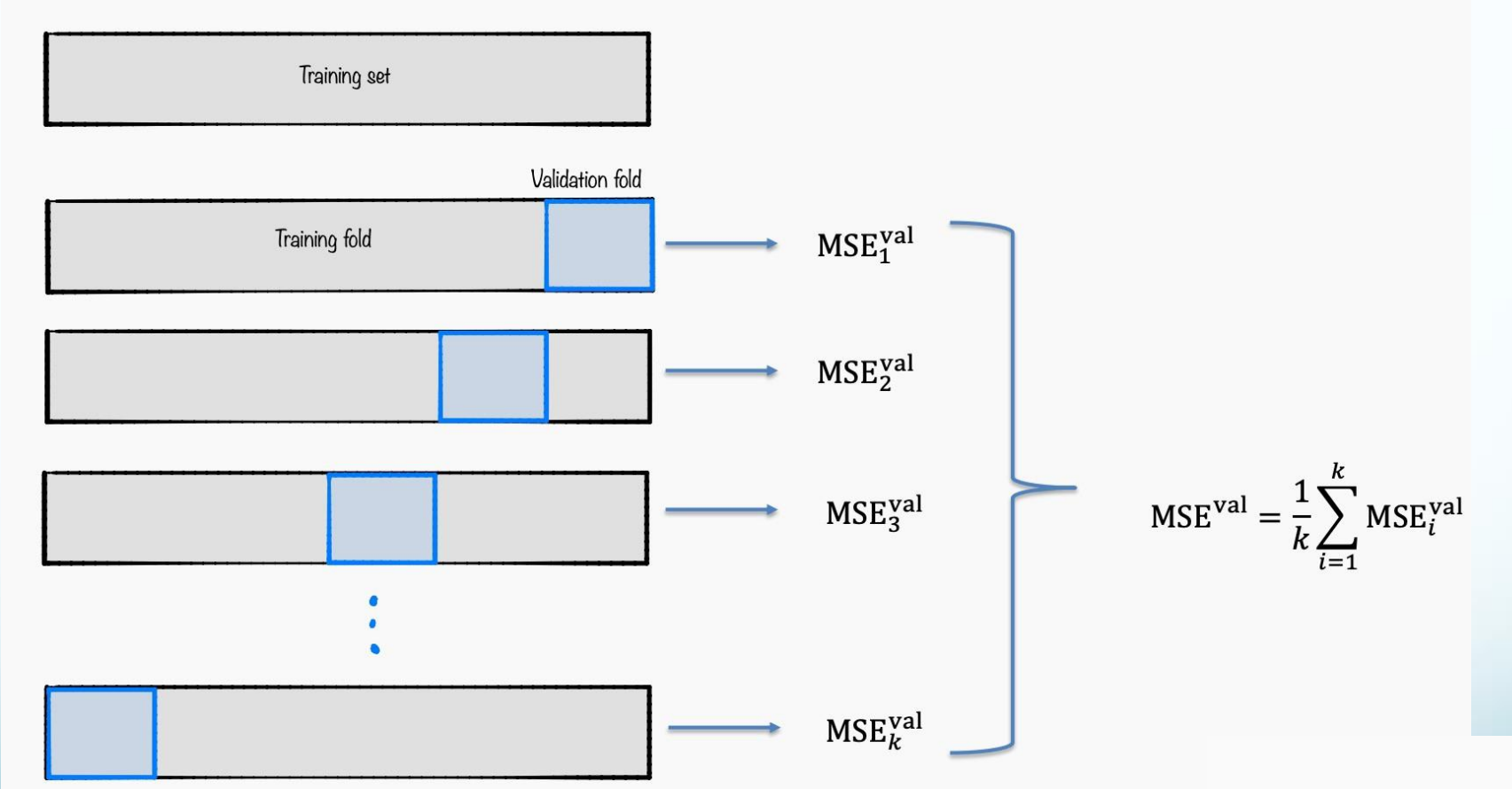
Cross Validation: Motivation

*Let's use **multiple** validation sets and average the validation performance.*

***One approach:** randomly split the training set into training and validation multiple time*

***Problem: R**andomly creating these sets can create the scenario where important features of the data may never appear in random draws.*

Cross Validation



V-Fold Cross Validation

V-fold CV aims to emulate the training/validation split for evaluating results on "pseudo"-validation data. The general process is as follows:

1. Split the data into V equal-sized non-overlapping subsets, called *folds*.
2. Remove the first fold (this fold will play the role of the pseudo-validation set), and use the remaining $V-1$ folds (the pseudo-training set) to train the algorithm.
3. Use the withheld first fold (the pseudo-validation set) to evaluate the algorithm that you just trained on the other $V-1$ folds using a relevant performance measure.

V-Fold Cross Validation

4. Replace the first fold, and remove the second fold (the second fold is now the pseudo-validation set). Train the algorithm using the other $V - 1$ folds (including the previously withheld first fold). Evaluate the algorithm on the withheld second fold.

5. Repeat step 4 until each fold has been used as the pseudo-validation set, and you have V values of the performance measure for your algorithm.

6. Combine (e.g., compute the mean of) the V performance measure values, each computed on a withheld fold.

Leave-One-Out

In practice:
5-fold CV (i.e., V=5) is common for small datasets up to a few thousand data points),

10-fold (V=10) CV is common for larger datasets.

Leave-one-out cross-validation: Each data point is a fold, so V = n where n is the number of

Today's Learning Objectives

Students will be able to:

- ✓ Review: Identify **overfitting** in terms of **bias** and **variance**
- ✓ Apply **model selection** for **overfitting**
- ✓ Understand the need for **cross validation** to address overfitting
- ✗ Apply **cross validation** and **regularization** to address overfitting

Regularization

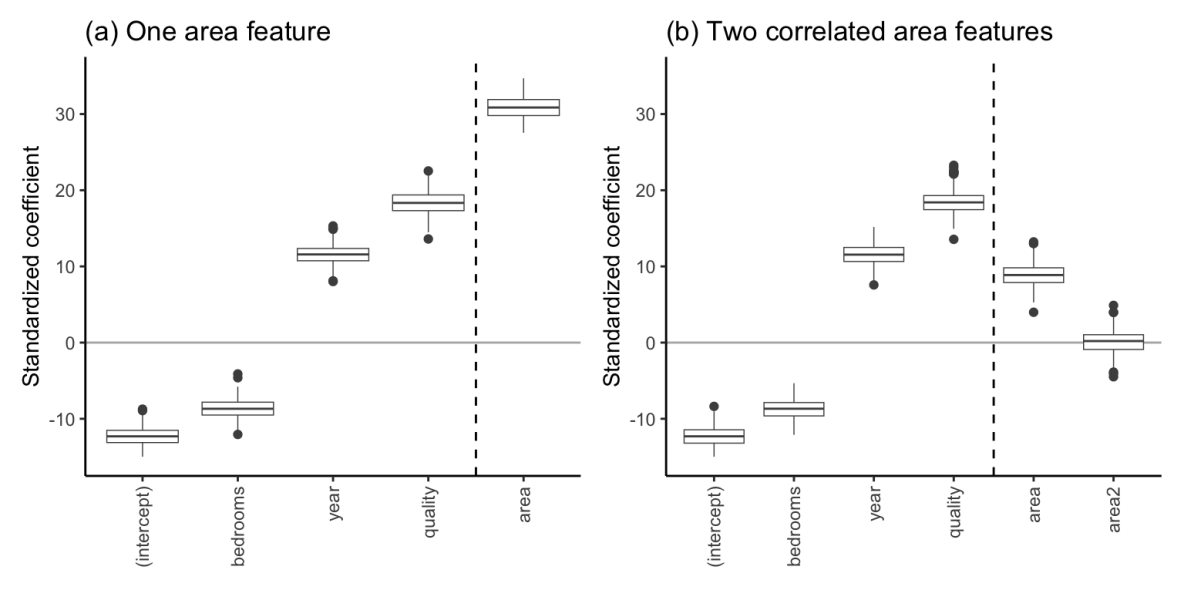
Regularization is a technique that forces predictive algorithms to simpler solutions by adding constraints to the minimization/optimization problem.

- Adds penalty based on weights to the model.
- Automated feature selection technique
- Addresses overfitting (too many features)
- Collinearity of features

- Best practice: **Standardize variables before regularization**

Collinearity Example

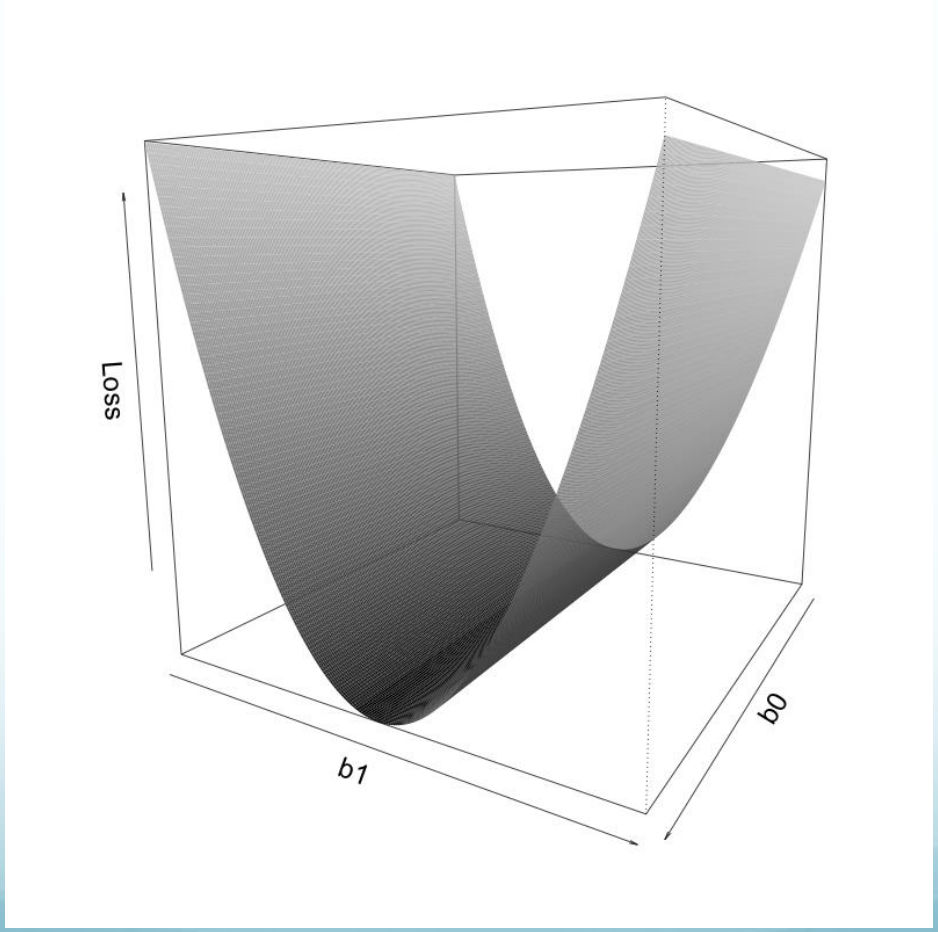
$$\text{predicted price} = b_0 + b_1\text{year} + b_2\text{bedrooms} + b_3\text{quality} + b_4\textbf{area}.$$



$$\text{predicted price} = b_0 + b_1\text{year} + b_2\text{bedrooms} + b_3\text{quality} + b_4\textbf{area} + b_5\textbf{area2}.$$

How do we optimize this?

$$\boldsymbol{predicted\ price} = \boldsymbol{b\_0} + \boldsymbol{b\_1} \times \boldsymbol{area}.$$



Ridge Regression

- Add a **regularization penalty** to the loss function
- Loss artificially increases as the values of b0 and b1 move farther from a particular point such as b0=0 and b1=0.
- Since LS algorithm minimizes loss, this forces b0 and b1 to stay relatively close to b0=0 and b1=0.
- Quadratic (squared) L2 penalty term is called **L2 regularization**
- **Ridge regression algorithm** is L2 penalty term with Least Squares algorithm

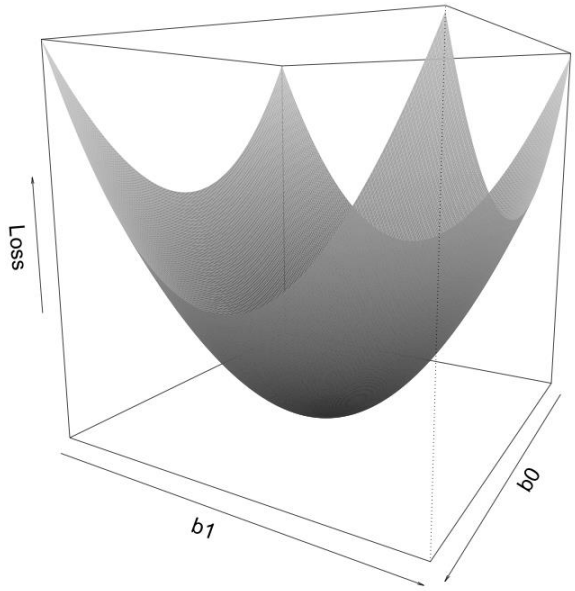
Ridge Regression

Find the values of b_0 and b_1 that make the regularized LS loss

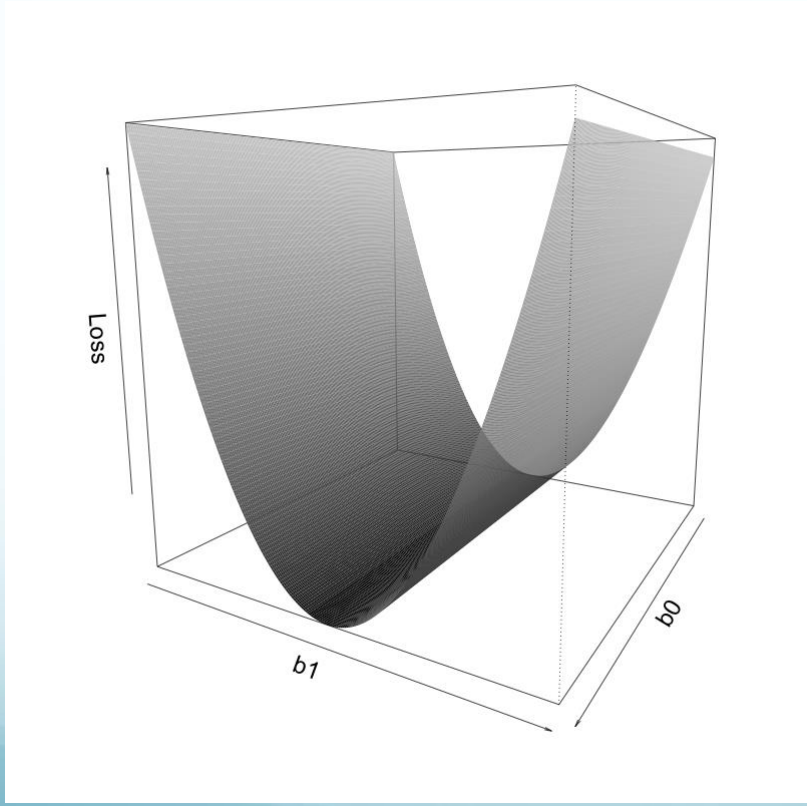
$$\sum_i (\text{observed price}_i - (b_0 + b_1 \text{area}_i))^2 + \lambda(b_0^2 + b_1^2)$$

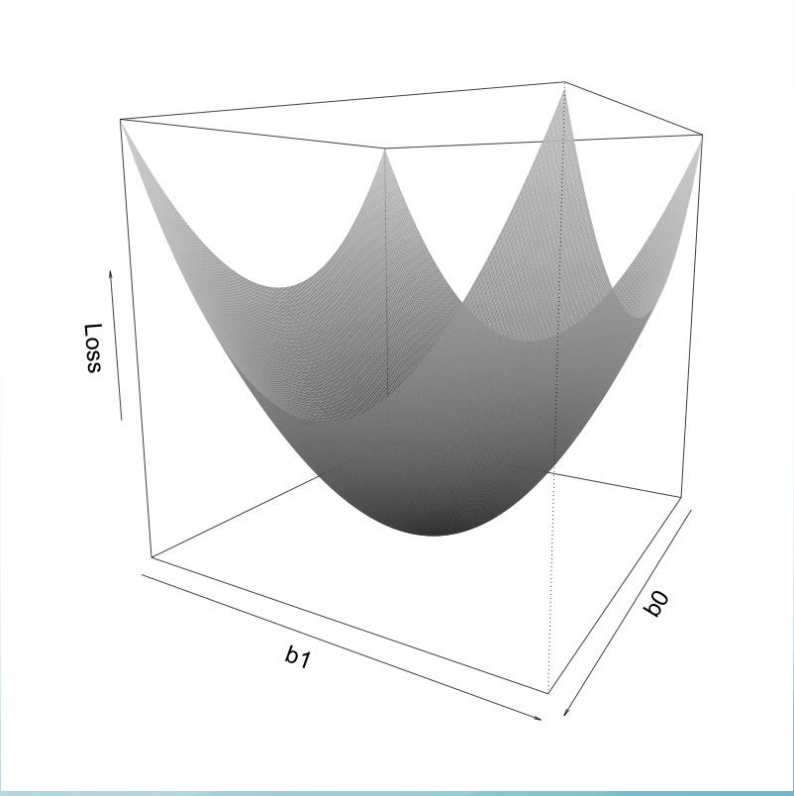
as small as possible (for some $\lambda \geq 0$).

- Quadratic (squared) L2 **penalty term** is called **L2 regularization**
- **Regularization hyperparameter is ƛ**



Ridge Regression Loss





Lasso Regression

- Add a **regularization penalty** to the loss function
- Absolute value or L1 penalty term is called **L1 regularization**
- **Lasso regression algorithm** is L1 penalty term with Least Squares algorithm

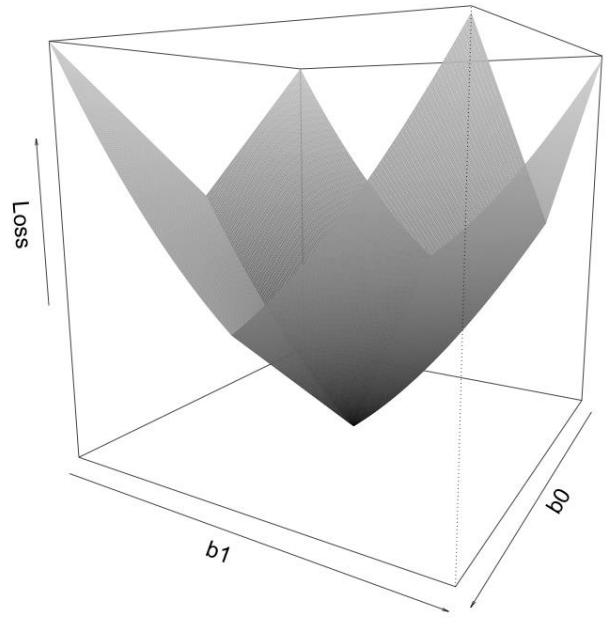
Lasso Regression

Find the values of b_0 and b_1 that make the regularized LS loss

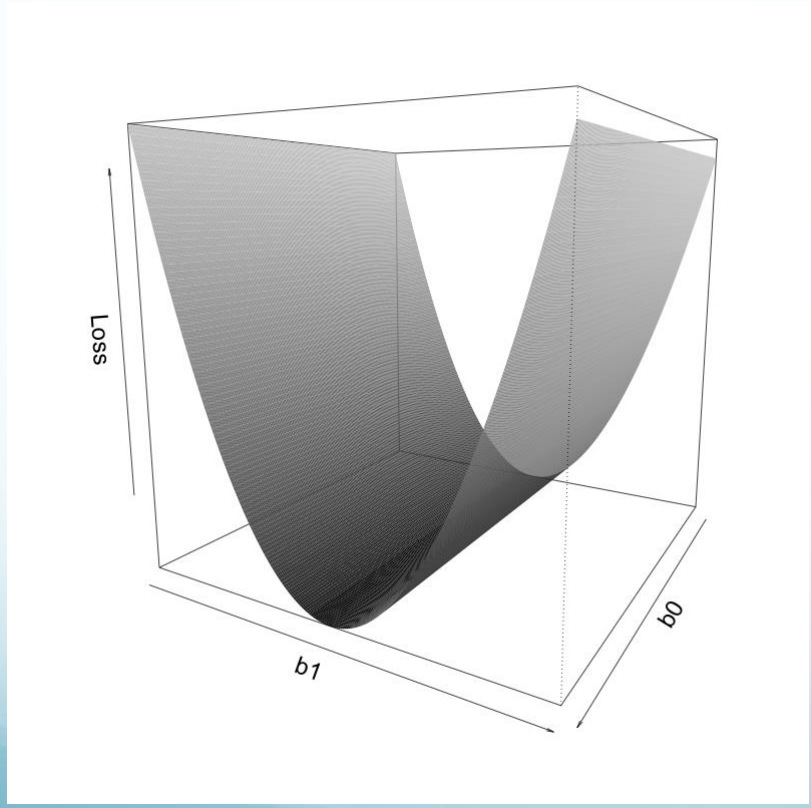
$$\sum_i (\text{observed price}_i - (b_0 + b_1 \text{area}_i))^2 + \lambda(|b_0| + |b_1|)$$

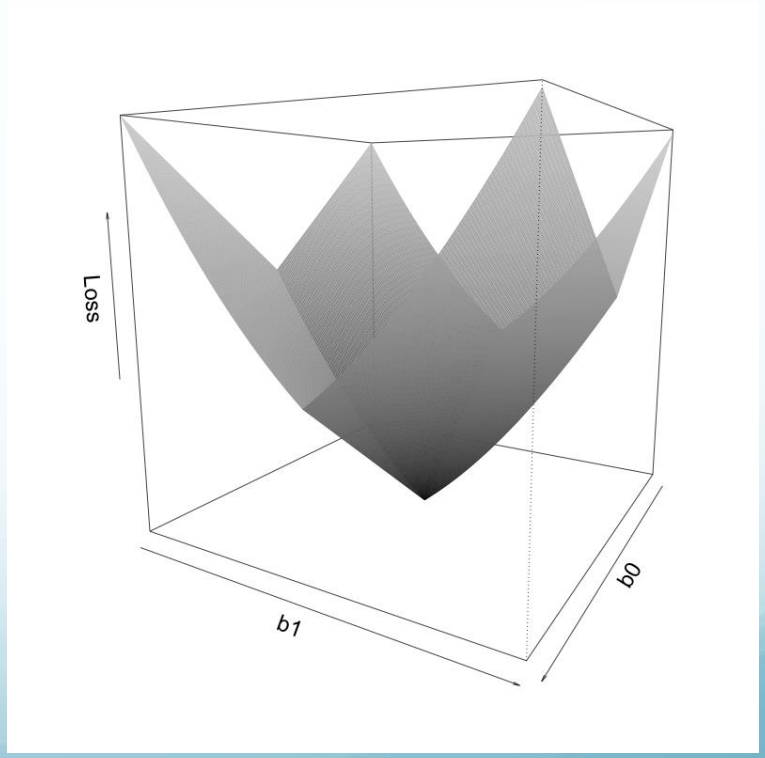
as small as possible (for some $\lambda \geq 0$).

- Absolute value or L1 **penalty term** is called **L1 regularization**
- **Regularization hyperparameter is ƛ**



Lasso Regression Loss





How to choose hyperparameters?

- We can use **cross validation!**

1. Decide on a range of potential values for the penalty term, λ. Note that many software implementations will do this for you.
2. Split the data into V (e.g., $V = 10$) nonoverlapping folds of approximately the same size.
3. Remove the first fold (this fold will play the role of the pseudo-validation set), and use the remaining $V - 1$ folds (which will play the role of the pseudo-training set) to train the regularized LS fit using each value of λ.
4. Calculate the error (e.g., mean squared error (MSE)) for each of the regularized LS fits—one for each λ—using the first withheld CV-fold pseudo-validation set.

How to choose hyperparameters?

- We can use **cross validation!**

5. Replace the withheld first fold and now remove the second fold (the second fold will now play the role of the pseudo-validation set). Use the remaining $V - 1$ folds to train the algorithm for each value of λ. Evaluate the fits using the withheld second fold (e.g., using MSE).

6. Repeat this process until all the V folds have been used as the withheld validation set, resulting in V measurements of the algorithm's performance for each λ.

7. For each value of λ, calculate the average of the V errors. The average of the V errors is called the *CV error*.

8. Select the λ that had the lowest CV error, or that you judge to be the best (e.g., taking stability into consideration).

Your turn: Regularization and CV

Please get the Jupyter notebook

Go to:
https://colab.research.google.com/drive/12z5-caCx9wWoGzRJEuDBkHwEdGSfq_mA?usp=sharing

Save a copy to your Google Drive and keep notes there...

[Yu, Barter 2024]

Today's Learning Objectives

Students will be able to:

- Review: Identify **overfitting** in terms of **bias** and **variance**
- Apply **model selection** for **overfitting**
- Understand the need for **cross validation** to address overfitting
- Apply **cross validation** and **regularization** to address overfitting

Citations:

Yu, B., & Barter, R. L. (2024). Veridical data science: The practice of responsible data analysis and decision making. The MIT Press.

Shah. C. (2020) A hands-on introduction to data science. Cambridge University Press.

Data 100, Fall 2024, UC Berkeley.

Baharan Mirzasoleiman, UCLA CS M148 Winter 2024 Lecture 4-6 Notes

Sebatstian Raschka, University of Wisconsin, Stat451, Fall 2020, Lecture 8 Notes