Model Selection
Prediction Error

- Prediction Error = $\sigma^2 + \text{MSE}(\theta, \theta^*)$
- The variance $\sigma^2$ is irreducible error. In regression, irreducible error arises because $Y$ is not completely determined by $X$.
- $\text{MSE}(\theta, \theta^*)$ is reducible error. Reducible error varies from learning algorithm to learning algorithm. It presents opportunities for improvement.
- $\text{MSE}(\theta, \theta^*) = \mathbb{E}_X[(\theta(x) - \theta^*)^2]$
Bias-Variance Decomposition

- **Theorem**: $\text{MSE}(\theta, \theta^*) = \text{Bias}[\theta, \theta^*]^2 + \text{Var}[\theta]$

- So error is a combination of bias and variance in our estimator.

- Ideally, we would reduce both, but this is often impossible.

- Instead, we usually trade off one against the other.
Proof of Bias-Variance Decomposition

$$\text{MSE}(\theta, \theta^*) = E_X[(\theta(x) - \theta^*)^2] = E_X[\theta^2(x) - 2\theta^*\theta(x) + (\theta^*)^2] = E_X[\theta^2(x)] - 2\theta^*E_X[\theta(x)] + (\theta^*)^2$$

$$\text{Bias}[\theta, \theta^*]^2 = (E_X[\theta(x)] - \theta^*)^2 = (E_X[\theta(x)] - \theta^*)(E_X[\theta(x)] - \theta^*) = (E_X[\theta(x)])^2 - 2\theta^*E_X[\theta(x)] + (\theta^*)^2$$

$$\text{Var}[\theta] = E_X[(\theta(x) - E_X[\theta(x)])^2] = E_X[\theta^2(x)] - (E_X[\theta(x)])^2$$

$$\text{Bias}[\theta, \theta^*]^2 + \text{Var}[\theta] = (E_X[\theta(x)])^2 - 2\theta^*E_X[\theta(x)] + (\theta^*)^2 + E_X[\theta^2(x)] - (E_X[\theta(x)])^2$$
Mean Squared Error

- $\text{MSE}(\theta, \theta^*) = \text{Bias}[\theta, \theta^*]^2 + \text{Var}[\theta]$
- **Fact**: For an unbiased estimator, the MSE is just the variance!
  - $\text{MSE}(X_1) = \sigma$
  - $\text{MSE}(\bar{X}) = \sigma/n$
  - $\text{MSE}(\bar{X}) < \text{MSE}(X_1)$
Bias-Variance Tradeoff

Image Source
Model Selection

Find a model that appropriately balances complexity and generalization capabilities: i.e., that optimizes the bias-variance tradeoff.

Feature Selection

- High bias, low variance (underfitting)
  - Choose a simple relationship among variables (e.g., linear).
  - Select only a few key features; each choice introduces bias.

- Low bias, high variance (overfitting)
  - Make few structural assumptions.
  - Throw in everything in the kitchen sink!
Overfitting

Low bias, high variance

- A model overfits when it “memorizes” the observed data
- Overfit models do not generalize well to unobserved data

Image Source
Overfitting

- Problem: Models are always biased towards training data
- A model overfits when it “memorizes” the training data
- Overfit models cannot generalize to test data
- Solution: Use test data to evaluate models to mitigate the risk that they overfit
Curse of Dimensionality

Adding new features to a model (i.e., increasing the dimensionality) in the hopes of improving performance will eventually degrade performance.
Curse of Dimensionality

As the dimensionality of the data (i.e., the number of features) increases, “the volume of the space increases so fast that the available data become sparse.”

Example from Stack Exchange:

- To find on your favorite kind of cookie among four possible flavors (sweet, salty, bitter, sour), requires eating four cookies.
- If there is an additional dimension, e.g., color, and there are three possible colors, you now now have to eat $4 \times 3 = 12$ cookies to find your favorite.
- Add another dimension, e.g., shape, with five possibilities, and you now have to eat $4 \times 3 \times 5 = 60$ cookies!
Curse of Dimensionality

As the dimensionality of the data (i.e., the number of features) increases, “the volume of the space increases so fast that the available data become sparse.”

To cover 20% of the population:
- Need 20% of the data in 1 dimension: \((.2)^1 \sim .2\)
- Need 45% of the data in 2 dimensions: \((.45)^2 \sim .2\)
- Need 58% of the data in 3 dimensions: \((.58)^3 \sim .2\)
Linear Regression, Regularized
Regularization

- The bias-variance decomposition suggests trading bias for variance.
- Regularization is a technique that introduces bias to reduce variance.
- Shrinkage is a form of regularization that shrinks estimates towards zero.
- This technique discourages learning a more complex, flexible model, avoiding the risk of overfitting.
Regularizers

● A regularizer is a penalty term that is added to an objective function (e.g., minimize the sum of the squared residuals) to penalize large coefficients.

● Two popular choices lead to two popular variants on standard regression:
  ○ Ridge regression: Minimizes the sum of the coefficients squared
  ○ LASSO: Minimizes the sum their absolute values
    ■ Least absolute shrinkage and selection operator
  ○ Elastic Net: Minimizes a combination of the two
Regularizers Visualized in 2d

Assuming two coefficients, an increase in one is offset by a decrease in the other

\[ \sum_i |\theta_i|^p = |\theta_1| + |\theta_2| \]

Image Source
• The new, regularized objective must balance the original objective against the regularization term.
• This balance is achieved via a parameter, \( \lambda \).
• Higher \( \lambda \) increases bias, so decreases variance.
Variable Selection

• Identifying independent variables whose relationship to the dependent variable is “important”.

• Simple heuristic for eliminating variables from a model: is the coefficient (essentially) zero?

• Ridge regression does not set coefficients to zero, unless $\lambda = 0$, so it cannot be used for variable selection.

• So, while ridge regression is useful for prediction, it is less effective when the goal is to explain relationships among variables.

• LASSO, however, can set some coefficients to zero, so it is more popular widely used when the goal is to build an interpretable model.
Cross Validation
Training vs. Test Data

- Divide data into two sets: training set and test set
- As their names suggest:
  - Train your model on the training set
  - Test your model on the test set
- Goal is to build a model that generalizes well from the training set to the test set, so that it will hopefully also generalize well from in-sample data to out-of-sample data.
- To achieve this goal, the test set should be representative of the training set, and should be large enough to obtain statistically significant results.
Train vs. Test Error

- The points are the data
- The black line represents the true relationship
- The various colors refer to different estimators (linear, quadratic, & something crazy).

- The grey curve shows the training error. It decreases indefinitely.
- The red curve shows the test error. It has an elbow.
- The colored boxes correspond to the colored fits in the left plot.
Simple (e.g., linear) models are highly biased; as such, they often **underfit**, meaning they fail to capture regularities in the data.

Otoh, they are not sensitive to noise (i.e., they assume so much bias that they don’t change much with the data), so are comparatively low variance.

More complicated models are less biased. Because of their flexibility, they end up modeling noise (as well as signal), and consequently **overfit**.

Flexible models have high variance, b/c the models themselves can vary enormously with the data.
Holdout Method

- Partition our training data into a large training set and smaller testing set
- Train model on the training data
- Test model accuracy on the testing data
- Data are often shuffled first (e.g., if they were compiled by different sources)
Cross validation

● Partition data multiple times
  ○ If you want to partition your data 10 times, create 10 folds, and then use each fold as a test set, and the rest of the data as a training set
  ○ Average accuracy across all partitions to approximate model accuracy

● This is cross validation
  ○ Typical to use $k$ partitions for $k$-fold cross validation (usually $k = 10$)
  ○ Leave-one-out cross validation: cross validation, to the extreme: $k = n$, the sample size