Frontier Topics in Empirical Economics: Week 3
Machine Learning and Model Selection

Zibin Huang [1]

[1] College of Business, Shanghai University of Finance and Economics

Spetember 27, 2024

Machine Learning and Model Selection: Introduction

- In the last lecture, we learn some non-parametric and semi-parametric methods
- We now have many tools in our box beyond linear regression
  - Kernel regression, local polynomial regression
  - Series regression, partial linear regression
- Which method should we choose?

Machine Learning and Model Selection: Introduction

- In the last lecture, we learn some non-parametric and semi-parametric methods
- We now have many tools in our box beyond linear regression
  - Kernel regression, local polynomial regression
  - Series regression, partial linear regression
- Which method should we choose?

Machine Learning and Model Selection: Introduction

- In the last lecture, we learn some non-parametric and semi-parametric methods
- We now have many tools in our box beyond linear regression
  - Kernel regression, local polynomial regression
  - Series regression, partial linear regression
- Which method should we choose?

Machine Learning and Model Selection: Introduction

- In the last lecture, we learn some non-parametric and semi-parametric methods
- We now have many tools in our box beyond linear regression
  - Kernel regression, local polynomial regression
  - Series regression, partial linear regression
- Which method should we choose?

Machine Learning and Model Selection: Introduction

- In the last lecture, we learn some non-parametric and semi-parametric methods
- We now have many tools in our box beyond linear regression
  - Kernel regression, local polynomial regression
  - Series regression, partial linear regression
- Which method should we choose?

Machine Learning and Model Selection: Introduction

- In the last lecture, we learn some non-parametric and semi-parametric methods
- We now have many tools in our box beyond linear regression
  - Kernel regression, local polynomial regression
  - Series regression, partial linear regression
- Which method should we choose?

Machine Learning and Model Selection: Introduction

- Even for a given method, such as simple regression
- The functional form is still flexible
  - Why linear? Simple? Why not $y = lnx + x^3 + e$?
  - What covariates to include?
    In Mincer equation, we regression wage on edu, exp, and exp^2. Why not edu^3?

Machine Learning and Model Selection: Introduction

- Even for a given method, such as simple regression
- The functional form is still flexible
  - Why linear? Simple? Why not $y = lnx + x^3 + e$?
  - What covariates to include?

    In Mincer equation, we regression *wage* on *edu*, *exp*, and exp^2. Why not edu^3?

Machine Learning and Model Selection: Introduction

- Even for a given method, such as simple regression
- The functional form is still flexible
  - Why linear? Simple? Why not $y = lnx + x^3 + e$?
  - What covariates to include?
    In Mincer equation, we regression *wage* on *edu*, *exp*, and exp^2. Why not edu^3?

Machine Learning and Model Selection: Introduction

- Even for a given method, such as simple regression
- The functional form is still flexible
  - Why linear? Simple? Why not $y = lnx + x^3 + e$?
  - What covariates to include?
    In Mincer equation, we regression *wage* on *edu*, *exp*, and exp^2. Why not edu^3?

Machine Learning and Model Selection: Introduction

- Even for a given method, such as simple regression
- The functional form is still flexible
  - Why linear? Simple? Why not $y = lnx + x^3 + e$?
  - What covariates to include?
    In Mincer equation, we regression *wage* on *edu*, *exp*, and exp^2. Why not edu^3?

Machine Learning and Model Selection: Introduction

- Model selection issue has been ignored in applied economics for such a long time
- More due to data availability issue
- Nowadays, more and more datasets are available with huge sizes
- BIG DATA! More chances!
- We should seriously consider model selection issue

Machine Learning and Model Selection: Introduction

- Model selection issue has been ignored in applied economics for such a long time
- More due to data availability issue
- Nowadays, more and more datasets are available with huge sizes
- BIG DATA! More chances!
- We should seriously consider model selection issue

Machine Learning and Model Selection: Introduction

- Model selection issue has been ignored in applied economics for such a long time
- More due to data availability issue
- Nowadays, more and more datasets are available with huge sizes
- BIG DATA! More chances!
- We should seriously consider model selection issue

Machine Learning and Model Selection: Introduction

- Model selection issue has been ignored in applied economics for such a long time
- More due to data availability issue
- Nowadays, more and more datasets are available with huge sizes
- BIG DATA! More chances!
- We should seriously consider model selection issue

Machine Learning and Model Selection: Introduction

- Model selection issue has been ignored in applied economics for such a long time
- More due to data availability issue
- Nowadays, more and more datasets are available with huge sizes
- BIG DATA! More chances!
- We should seriously consider model selection issue

Machine Learning and Model Selection: Introduction

- Model selection issue has been ignored in applied economics for such a long time
- More due to data availability issue
- Nowadays, more and more datasets are available with huge sizes
- BIG DATA! More chances!
- We should seriously consider model selection issue

Machine Learning and Model Selection: Introduction

- There are two approaches to choose a model
  - Data driven method (Machine learning)
  - Prior causal structure (DAG)
- Data driven method focus on using purely data to determine the model without prior information
- Prior causal structure means that we determine the model with assumed causal links and economic knowledge

Machine Learning and Model Selection: Introduction

- There are two approaches to choose a model
  - Data driven method (Machine learning)
  - Prior causal structure (DAG)
- Data driven method focus on using purely data to determine the model without prior information
- Prior causal structure means that we determine the model with assumed causal links and economic knowledge

Machine Learning and Model Selection: Introduction

- There are two approaches to choose a model
  - Data driven method (Machine learning)
  - Prior causal structure (DAG)
- Data driven method focus on using purely data to determine the model without prior information
- Prior causal structure means that we determine the model with assumed causal links and economic knowledge

Machine Learning and Model Selection: Introduction

- There are two approaches to choose a model
  - Data driven method (Machine learning)
  - Prior causal structure (DAG)
- Data driven method focus on using purely data to determine the model without prior information
- Prior causal structure means that we determine the model with assumed causal links and economic knowledge

Machine Learning and Model Selection: Introduction

- There are two approaches to choose a model
  - Data driven method (Machine learning)
  - Prior causal structure (DAG)
- Data driven method focus on using purely data to determine the model without prior information
- Prior causal structure means that we determine the model with assumed causal links and economic knowledge

Machine Learning and Model Selection: Introduction

- There are two approaches to choose a model
  - Data driven method (Machine learning)
  - Prior causal structure (DAG)
- Data driven method focus on using purely data to determine the model without prior information
- Prior causal structure means that we determine the model with assumed causal links and economic knowledge

Machine Learning and Model Selection: Introduction

- Today, we will discuss the data driven model selection method first
- We select models only using data
- We do not put our economic knowledge into the process
- Let's first introduce a major statistical concept: Bias-variance tradeoff

Machine Learning and Model Selection: Introduction

- Today, we will discuss the data driven model selection method first
- We select models only using data
- We do not put our economic knowledge into the process
- Let's first introduce a major statistical concept: Bias-variance tradeoff

Machine Learning and Model Selection: Introduction

- Today, we will discuss the data driven model selection method first
- We select models only using data
- We do not put our economic knowledge into the process
- Let's first introduce a major statistical concept: Bias-variance tradeoff

Machine Learning and Model Selection: Introduction

- Today, we will discuss the data driven model selection method first
- We select models only using data
- We do not put our economic knowledge into the process
- Let's first introduce a major statistical concept: Bias-variance tradeoff

Machine Learning and Model Selection: Introduction

- Today, we will discuss the data driven model selection method first
- We select models only using data
- We do not put our economic knowledge into the process
- Let's first introduce a major statistical concept: Bias-variance tradeoff

Machine Learning and Model Selection: Bias-variance Tradeoff

- A traditional linear model

$$y = x\beta + \epsilon \tag{1}$$

- A model with quadratic term

$$y = x\beta + x^2\alpha + \epsilon \tag{2}$$

- A non-parametric model

$$y = g(x) + \epsilon \tag{3}$$

- Why not always the second or the third one?

Machine Learning and Model Selection: Bias-variance Tradeoff

- A traditional linear model

$$y = x\beta + \epsilon \tag{1}$$

- A model with quadratic term

$$y = x\beta + x^2\alpha + \epsilon \tag{2}$$

- A non-parametric model

$$y = g(x) + \epsilon \tag{3}$$

- Why not always the second or the third one?

Machine Learning and Model Selection: Bias-variance Tradeoff

- A traditional linear model

$$y = x\beta + \epsilon \tag{1}$$

- A model with quadratic term

$$y = x\beta + x^2\alpha + \epsilon \tag{2}$$

- A non-parametric model

$$y = g(x) + \epsilon \tag{3}$$

- Why not always the second or the third one?

Machine Learning and Model Selection: Bias-variance Tradeoff

- A traditional linear model

$$y = x\beta + \epsilon \tag{1}$$

- A model with quadratic term

$$y = x\beta + x^2\alpha + \epsilon \tag{2}$$

- A non-parametric model

$$y = g(x) + \epsilon \tag{3}$$

- Why not always the second or the third one?

Machine Learning and Model Selection: Bias-variance Tradeoff

- A traditional linear model

$$y = x\beta + \epsilon \tag{1}$$

- A model with quadratic term

$$y = x\beta + x^2\alpha + \epsilon \tag{2}$$

- A non-parametric model

$$y = g(x) + \epsilon \tag{3}$$

- Why not always the second or the third one?

Machine Learning and Model Selection: Bias-variance Tradeoff

- Model A

$$y = x_1^{'}\beta + \epsilon \tag{4}$$

- Model B

$$y = x_1^{'}\beta_1 + x_2^{'}\beta_2 + \epsilon \tag{5}$$

- Why not always the second one?
- Always better to have a more complicated model?

Machine Learning and Model Selection: Bias-variance Tradeoff

- Model A

$$y = x_1^{'}\beta + \epsilon \tag{4}$$

- Model B

$$y = x_1^{'}\beta_1 + x_2^{'}\beta_2 + \epsilon \tag{5}$$

- Why not always the second one?
- Always better to have a more complicated model?

Machine Learning and Model Selection: Bias-variance Tradeoff

- Model A

$$y = x_1^{'}\beta + \epsilon \tag{4}$$

- Model B

$$y = x_1^{'}\beta_1 + x_2^{'}\beta_2 + \epsilon \tag{5}$$

- Why not always the second one?
- Always better to have a more complicated model?

Machine Learning and Model Selection: Bias-variance Tradeoff

- Model A

$$y = x_1^{'}\beta + \epsilon \tag{4}$$

- Model B

$$y = x_1^{'}\beta_1 + x_2^{'}\beta_2 + \epsilon \tag{5}$$

- Why not always the second one?
- Always better to have a more complicated model?

Machine Learning and Model Selection: Bias-variance Tradeoff

- Model A

$$y = x_1^{'}\beta + \epsilon \tag{4}$$

- Model B

$$y = x_1^{'}\beta_1 + x_2^{'}\beta_2 + \epsilon \tag{5}$$

- Why not always the second one?
- Always better to have a more complicated model?

Machine Learning and Model Selection: Bias-variance Tradeoff

- **Model Selection: Bias vs. Variance**

  Assume that:

$$Y = f(X) + \epsilon$$

- $\hat{f}(x)$ is a model trained by some data
- It will be changed when sample is changed: $\hat{f}^1(x), \hat{f}^2(x) \ldots$
- Expectation $E[\hat{f}(x)]$ is taken over different samples
- How good is the model?

Machine Learning and Model Selection: Bias-variance Tradeoff

- **Model Selection: Bias vs. Variance**
  Assume that:

$$Y = f(X) + \epsilon$$

- $\hat{f}(x)$ is a model trained by some data
- It will be changed when sample is changed: $\hat{f}^1(x), \hat{f}^2(x)$...
- Expectation $E[\hat{f}(x)]$ is taken over different samples
- How good is the model?

Machine Learning and Model Selection: Bias-variance Tradeoff

- **Model Selection: Bias vs. Variance**
  Assume that:

$$Y = f(X) + \epsilon$$

- $\hat{f}(x)$ is a model trained by some data
- It will be changed when sample is changed: $\hat{f}^1(x), \hat{f}^2(x)...$
- Expectation $E[\hat{f}(x)]$ is taken over different samples
- How good is the model?

Machine Learning and Model Selection: Bias-variance Tradeoff

- **Model Selection: Bias vs. Variance**

  Assume that:

$$Y = f(X) + \epsilon$$

- $\hat{f}(x)$ is a model trained by some data
- It will be changed when sample is changed: $\hat{f}^1(x), \hat{f}^2(x)...$
- Expectation $E[\hat{f}(x)]$ is taken over different samples
- How good is the model?

Machine Learning and Model Selection: Bias-variance Tradeoff

- **Model Selection: Bias vs. Variance**
  Assume that:

$$Y = f(X) + \epsilon$$

- $\hat{f}(x)$ is a model trained by some data
- It will be changed when sample is changed: $\hat{f}^1(x), \hat{f}^2(x)...$
- Expectation $E[\hat{f}(x)]$ is taken over different samples
- How good is the model?

Machine Learning and Model Selection: Bias-variance Tradeoff

- **Model Selection: Bias vs. Variance**

  Assume that:

$$Y = f(X) + \epsilon$$

- $\hat{f}(x)$ is a model trained by some data
- It will be changed when sample is changed: $\hat{f}^1(x), \hat{f}^2(x)$...
- Expectation $E[\hat{f}(x)]$ is taken over different samples
- How good is the model?

Machine Learning and Model Selection: Bias-variance Tradeoff

- The prediction mean squared error at some point x_0:

$$\begin{aligned} E[(Y - \hat{f}(x_0))^2 | X = x_0] &= \sigma_\epsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - E\hat{f}(x_0)]^2 \\ &= \text{irreducible error} + Bias^2 + Variance \end{aligned}$$

- Model complexity $\Longrightarrow$ Bias $\downarrow$, Variance $\uparrow$
- Super complicated model $\Longrightarrow$ Variance $\uparrow\uparrow\uparrow$ (very sensitive when data change)
- Overfit current data $\Longrightarrow$ Poor out-of-sample prediction

Machine Learning and Model Selection: Bias-variance Tradeoff

- The prediction mean squared error at some point x_0:

$$
\begin{aligned}
E[(Y - \hat{f}(x_0))^2 | X = x_0] &= \sigma_\epsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - E\hat{f}(x_0)]^2 \\
&= \text{irreducible error} + Bias^2 + Variance
\end{aligned}
$$

- Model complexity $\Rightarrow$ Bias $\downarrow$, Variance $\uparrow$
- Super complicated model $\Rightarrow$ Variance $\uparrow\uparrow\uparrow$ (very sensitive when data change)
- Overfit current data $\Rightarrow$ Poor out-of-sample prediction

Machine Learning and Model Selection: Bias-variance Tradeoff

- The prediction mean squared error at some point x_0:

$$\begin{aligned} E[(Y - \hat{f}(x_0))^2 | X = x_0] &= \sigma_\epsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - E\hat{f}(x_0)]^2 \\ &= \text{irreducible error} + \mathit{Bias}^2 + \mathit{Variance} \end{aligned}$$

- Model complexity ⇒ Bias ↓, Variance ↑
- Super complicated model ⇒ Variance ↑↑↑ (very sensitive when data change)
- Overfit current data ⇒ Poor out-of-sample prediction

Machine Learning and Model Selection: Bias-variance Tradeoff

- The prediction mean squared error at some point x_0:

$$\begin{aligned} E[(Y - \hat{f}(x_0))^2 | X = x_0] &= \sigma_\epsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - E\hat{f}(x_0)]^2 \\ &= \text{irreducible error} + Bias^2 + Variance \end{aligned}$$

- Model complexity $\Rightarrow$ Bias $\downarrow$, Variance $\uparrow$
- Super complicated model $\Rightarrow$ Variance $\uparrow\uparrow\uparrow$ (very sensitive when data change)
- Overfit current data $\Rightarrow$ Poor out-of-sample prediction

Machine Learning and Model Selection: Bias-variance Tradeoff

- The prediction mean squared error at some point x_0:

$$\begin{aligned} E[(Y - \hat{f}(x_0))^2 | X = x_0] &= \sigma_\epsilon^2 + [E\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - E\hat{f}(x_0)]^2 \\ &= \text{irreducible error} + Bias^2 + Variance \end{aligned}$$

- Model complexity $\Rightarrow$ Bias $\downarrow$, Variance $\uparrow$
- Super complicated model $\Rightarrow$ Variance $\uparrow\uparrow\uparrow$ (very sensitive when data change)
- Overfit current data $\Rightarrow$ Poor out-of-sample prediction

Machine Learning and Model Selection: An Example of Overfitting

- Consider a data generating process

$$Y = 1 + 1.5X + \epsilon$$
$$\epsilon \sim N(0, 100)$$

It is a noisy process.

- Simulate 30 observations from this process
- Let's start to fit it with different polynomials
- Green line is the true DGP
- Red line is the fitting function

Machine Learning and Model Selection: An Example of Overfitting

- Consider a data generating process

$$Y = 1 + 1.5X + \epsilon$$
$$\epsilon \sim N(0, 100)$$

It is a noisy process.

- Simulate 30 observations from this process
- Let's start to fit it with different polynomials
- Green line is the true DGP
- Red line is the fitting function

Machine Learning and Model Selection: An Example of Overfitting

- Consider a data generating process

$$Y = 1 + 1.5X + \epsilon$$
$$\epsilon \sim N(0, 100)$$

  It is a noisy process.

- Simulate 30 observations from this process
- Let's start to fit it with different polynomials
- Green line is the true DGP
- Red line is the fitting function

Machine Learning and Model Selection: An Example of Overfitting

- Consider a data generating process

$$Y = 1 + 1.5X + \epsilon$$
$$\epsilon \sim N(0, 100)$$

  It is a noisy process.
- Simulate 30 observations from this process
- Let's start to fit it with different polynomials
- Green line is the true DGP
- Red line is the fitting function

Machine Learning and Model Selection: An Example of Overfitting

- Consider a data generating process

$$Y = 1 + 1.5X + \epsilon$$
$$\epsilon \sim N(0, 100)$$

  It is a noisy process.

- Simulate 30 observations from this process
- Let's start to fit it with different polynomials
- Green line is the true DGP
- Red line is the fitting function

Machine Learning and Model Selection: An Example of Overfitting

- Consider a data generating process

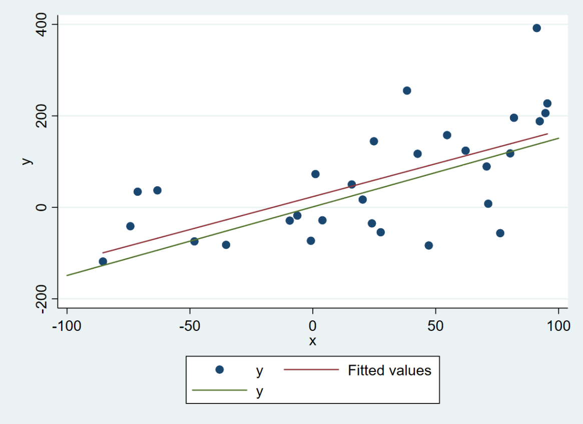
$$Y = 1 + 1.5X + \epsilon$$
$$\epsilon \sim N(0, 100)$$

It is a noisy process.

- Simulate 30 observations from this process
- Let's start to fit it with different polynomials
- Green line is the true DGP
- Red line is the fitting function

Machine Learning and Model Selection: An Example of Overfitting

Figure: First Order (Linear) Fitting



Machine Learning and Model Selection: An Example of Overfitting

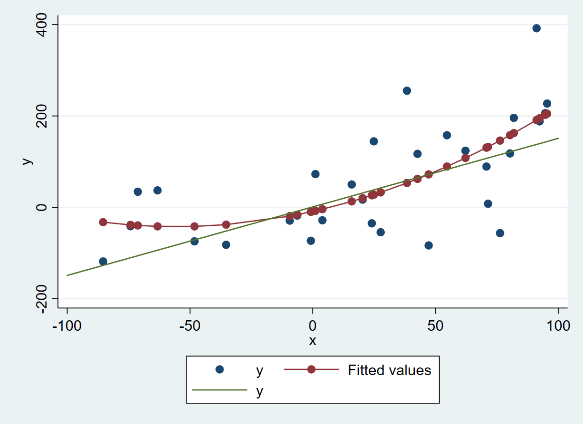
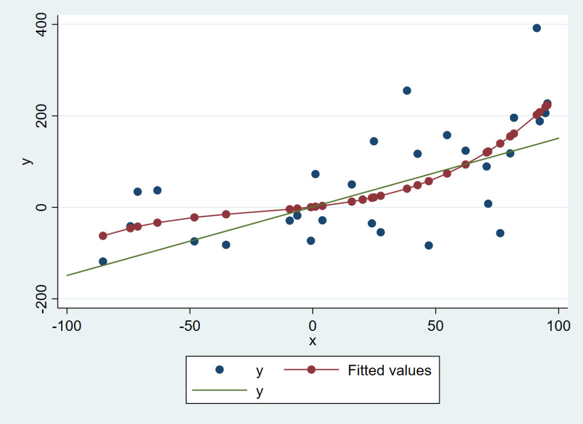


Figure: Second Order (Quadratic) Fitting

Machine Learning and Model Selection: An Example of Overfitting

Figure: Third Order (Cubic) Fitting



Machine Learning and Model Selection: An Example of Overfitting

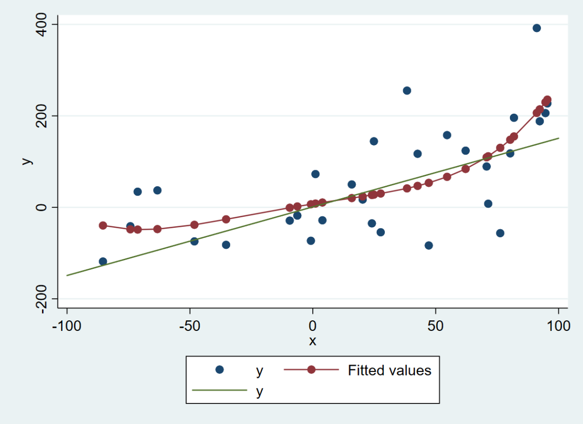
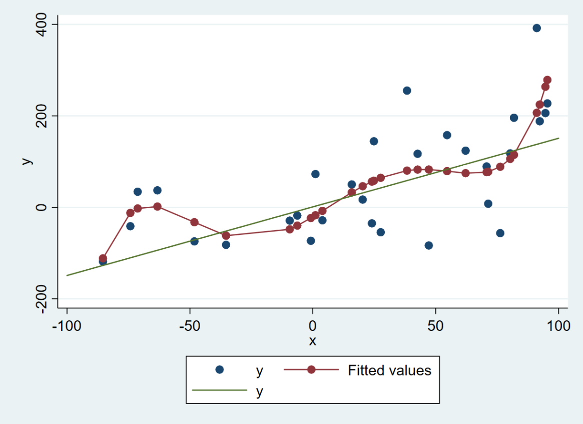


Figure: Fourth Order Fitting

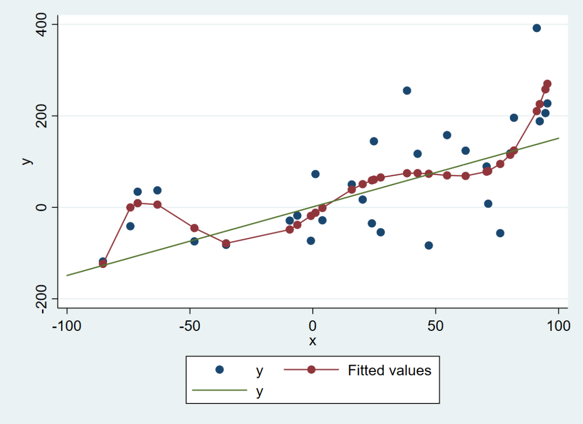
Machine Learning and Model Selection: An Example of Overfitting

Figure: Fifth Order Fitting



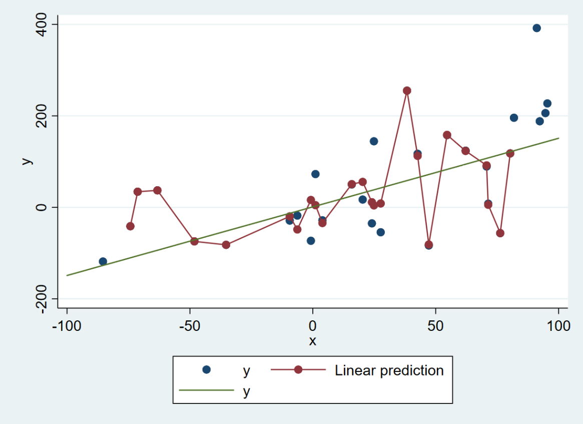
Machine Learning and Model Selection: An Example of Overfitting

Figure: Sixth Order Fitting



Machine Learning and Model Selection: An Example of Overfitting

Figure: Twentieth Order Fitting

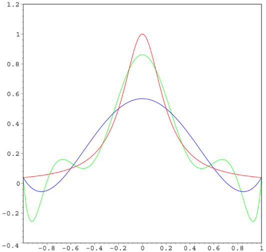


Machine Learning and Model Selection: An Example of Overfitting

High order polynomials: Picking up noises, not signals!!!
Bad out-of-sample prediction!!!

Machine Learning and Model Selection: An Example of Overfitting

- We have actually learned two kinds of overfitting
- Runge phenomenon and Gibbs phenomenon



Machine Learning and Model Selection: Goodness of Fit

There are many ways to measure the goodness of fit, considering overfitting

- Adjusted R-squared: the proportion of explained variations in y
  Still remember why we need to adjust for the number of regressors?
- AIC: Akaike Information Criterion
  $AIC = 2k + nln(RSS/n)$, k is the number of regressors
- BIC: Bayesian Information Criterion
  This is motivated by the Bayesian approach to model selection

Machine Learning and Model Selection: Goodness of Fit

There are many ways to measure the goodness of fit, considering overfitting

- Adjusted R-squared: the proportion of explained variations in y
  Still remember why we need to adjust for the number of regressors?
- AIC: Akaike Information Criterion
  $AIC = 2k + nln(RSS/n)$, k is the number of regressors
- BIC: Bayesian Information Criterion
  This is motivated by the Bayesian approach to model selection

Machine Learning and Model Selection: Goodness of Fit

There are many ways to measure the goodness of fit, considering overfitting

- Adjusted R-squared: the proportion of explained variations in y
  Still remember why we need to adjust for the number of regressors?
- AIC: Akaike Information Criterion
  $AIC = 2k + nln(RSS/n)$, k is the number of regressors
- BIC: Bayesian Information Criterion
  This is motivated by the Bayesian approach to model selection

Machine Learning and Model Selection: Goodness of Fit

There are many ways to measure the goodness of fit, considering overfitting

- Adjusted R-squared: the proportion of explained variations in y
  Still remember why we need to adjust for the number of regressors?
- AIC: Akaike Information Criterion
  $AIC = 2k + nln(RSS/n)$, k is the number of regressors
- BIC: Bayesian Information Criterion
  This is motivated by the Bayesian approach to model selection

Machine Learning and Model Selection: Goodness of Fit

There are many ways to measure the goodness of fit, considering overfitting

- Adjusted R-squared: the proportion of explained variations in y
  Still remember why we need to adjust for the number of regressors?
- AIC: Akaike Information Criterion
  $AIC = 2k + nln(RSS/n)$, k is the number of regressors
- BIC: Bayesian Information Criterion
  This is motivated by the Bayesian approach to model selection

Machine Learning and Model Selection: Goodness of Fit

Another important measure is Cross-Validation (CV)

- The basic idea is to separate all samples into training sample and validation sample
- Training sample is used to train (estimate) the model
- Validation sample is then used to check the "out-of-sample" prediction
- We deliberately leave some observations out of estimation
- They can be used to check the model fit and avoid overfitting

Machine Learning and Model Selection: Goodness of Fit

Another important measure is Cross-Validation (CV)

- The basic idea is to separate all samples into training sample and validation sample
- Training sample is used to train (estimate) the model
- Validation sample is then used to check the "out-of-sample" prediction
- We deliberately leave some observations out of estimation
- They can be used to check the model fit and avoid overfitting

Machine Learning and Model Selection: Goodness of Fit

Another important measure is Cross-Validation (CV)

- The basic idea is to separate all samples into training sample and validation sample
- Training sample is used to train (estimate) the model
- Validation sample is then used to check the ”out-of-sample” prediction
- We deliberately leave some observations out of estimation
- They can be used to check the model fit and avoid overfitting

Machine Learning and Model Selection: Goodness of Fit

Another important measure is Cross-Validation (CV)

- The basic idea is to separate all samples into training sample and validation sample
- Training sample is used to train (estimate) the model
- Validation sample is then used to check the ”out-of-sample” prediction
- We deliberately leave some observations out of estimation
- They can be used to check the model fit and avoid overfitting

Machine Learning and Model Selection: Goodness of Fit

Another important measure is Cross-Validation (CV)

- The basic idea is to separate all samples into training sample and validation sample
- Training sample is used to train (estimate) the model
- Validation sample is then used to check the "out-of-sample" prediction
- We delibrately leave some observations out of estimation
- They can be used to check the model fit and avoid overfitting

Machine Learning and Model Selection: Goodness of Fit

Another important measure is Cross-Validation (CV)

- The basic idea is to separate all samples into training sample and validation sample
- Training sample is used to train (estimate) the model
- Validation sample is then used to check the "out-of-sample" prediction
- We deliberately leave some observations out of estimation
- They can be used to check the model fit and avoid overfitting

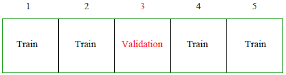
Machine Learning and Model Selection: Goodness of Fit

Another important measure is Cross-Validation (CV)

- The basic idea is to separate all samples into training sample and validation sample
- Training sample is used to train (estimate) the model
- Validation sample is then used to check the "out-of-sample" prediction
- We deliberately leave some observations out of estimation
- They can be used to check the model fit and avoid overfitting

Machine Learning and Model Selection: Goodness of Fit

Here is the process of CV



1	2	3	4	5
Train	Train	Validation	Train	Train

- First, we separate all samples into K parts
- Each time, we choose K-1 parts to train (estimate) the model
- We then use the remaining one part k to calculate the mean squared predicted error MSE_k
- We rotate the samples K times so that each part is used as the validation sample once, and have K pieces of MSE_k
- We take the average of them to have: $CV = \frac{1}{k}\sum_{k=1}^{K} MSE_k$
- This is called "K-fold Cross-Validation"

Machine Learning and Model Selection: Goodness of Fit

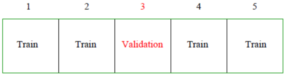
Here is the process of CV

1	2	3	4	5
Train	Train	Validation	Train	Train

- First, we separate all samples into K parts
- Each time, we choose K-1 parts to train (estimate) the model
- We then use the remaining one part k to calculate the mean squared predicted error MSE_k
- We rotate the samples K times so that each part is used as the validation sample once, and have K pieces of MSE_k
- We take the average of them to have: $CV = \frac{1}{k}\sum_{k=1}^{K} MSE_k$
- This is called "K-fold Cross-Validation"

Machine Learning and Model Selection: Goodness of Fit

Here is the process of CV

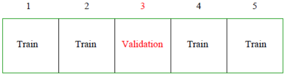


1	2	3	4	5
Train	Train	Validation	Train	Train

- First, we separate all samples into K parts
- Each time, we choose K-1 parts to train (estimate) the model
- We then use the remaining one part k to calculate the mean squared predicted error MSE_k
- We rotate the samples K times so that each part is used as the validation sample once, and have K pieces of MSE_k
- We take the average of them to have: $CV = \frac{1}{k}\sum_{k=1}^{K} MSE_k$
- This is called "K-fold Cross-Validation"

Machine Learning and Model Selection: Goodness of Fit

Here is the process of CV



1	2	3	4	5
Train	Train	Validation	Train	Train

- First, we separate all samples into K parts
- Each time, we choose K-1 parts to train (estimate) the model
- We then use the remaining one part k to calculate the mean squared predicted error MSE_k
- We rotate the samples K times so that each part is used as the validation sample once, and have K pieces of MSE_k
- We take the average of them to have: $CV = \frac{1}{k}\sum_{k=1}^{K} MSE_k$
- This is called "K-fold Cross-Validation"

Machine Learning and Model Selection: Goodness of Fit

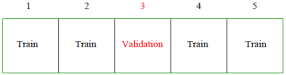
Here is the process of CV

1	2	3	4	5
Train	Train	Validation	Train	Train

- First, we separate all samples into K parts
- Each time, we choose K-1 parts to train (estimate) the model
- We then use the remaining one part k to calculate the mean squared predicted error MSE_k
- We rotate the samples K times so that each part is used as the validation sample once, and have K pieces of MSE_k
- We take the average of them to have: $CV = \frac{1}{k}\sum_{k=1}^{K} MSE_k$
- This is called "K-fold Cross-Validation"

Machine Learning and Model Selection: Goodness of Fit

Here is the process of CV

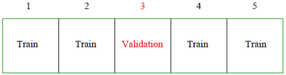


1	2	3	4	5
Train	Train	Validation	Train	Train

- First, we separate all samples into K parts
- Each time, we choose K-1 parts to train (estimate) the model
- We then use the remaining one part k to calculate the mean squared predicted error MSE_k
- We rotate the samples K times so that each part is used as the validation sample once, and have K pieces of MSE_k
- We take the average of them to have: $CV = \frac{1}{k}\sum_{k=1}^{K} MSE_k$
- This is called "K-fold Cross-Validation"

Machine Learning and Model Selection: Goodness of Fit

Here is the process of CV



1	2	3	4	5
Train	Train	Validation	Train	Train

- First, we separate all samples into K parts
- Each time, we choose K-1 parts to train (estimate) the model
- We then use the remaining one part k to calculate the mean squared predicted error MSE_k
- We rotate the samples K times so that each part is used as the validation sample once, and have K pieces of MSE_k
- We take the average of them to have: $CV = \frac{1}{k}\sum_{k=1}^{K} MSE_k$
- This is called "K-fold Cross-Validation"

Machine Learning and Model Selection: Goodness of Fit

Here is the process of CV

1	2	3	4	5
Train	Train	Validation	Train	Train

- First, we separate all samples into K parts
- Each time, we choose K-1 parts to train (estimate) the model
- We then use the remaining one part k to calculate the mean squared predicted error MSE_k
- We rotate the samples K times so that each part is used as the validation sample once, and have K pieces of MSE_k
- We take the average of them to have: $CV = \frac{1}{k} \sum_{k=1}^{K} MSE_k$
- This is called "K-fold Cross-Validation"

Machine Learning and Model Selection: Goodness of Fit

- CV measures the goodness of the out-of-sample prediction
- You have some data that is not used in the estimation and use it to check your estimation validity
- It helps you to determine which model fits better to the data, in terms of out-of-sample prediction
- Smaller CV means better fitting

Machine Learning and Model Selection: Goodness of Fit

- CV measures the goodness of the out-of-sample prediction
- You have some data that is not used in the estimation and use it to check your estimation validity
- It helps you to determine which model fits better to the data, in terms of out-of-sample prediction
- Smaller CV means better fitting

Machine Learning and Model Selection: Goodness of Fit

- CV measures the goodness of the out-of-sample prediction
- You have some data that is not used in the estimation and use it to check your estimation validity
- It helps you to determine which model fits better to the data, in terms of out-of-sample prediction
- Smaller CV means better fitting

Machine Learning and Model Selection: Goodness of Fit

- CV measures the goodness of the out-of-sample prediction
- You have some data that is not used in the estimation and use it to check your estimation validity
- It helps you to determine which model fits better to the data, in terms of out-of-sample prediction
- Smaller CV means better fitting

Machine Learning and Model Selection: Goodness of Fit

- CV measures the goodness of the out-of-sample prediction
- You have some data that is not used in the estimation and use it to check your estimation validity
- It helps you to determine which model fits better to the data, in terms of out-of-sample prediction
- Smaller CV means better fitting

Machine Learning and Model Selection: Goodness of Fit

- Now we have some measures of goodness
- That is, the "standard" of what is a "good" model
- Would that be possible to have an automatic algorithm to find a good model for us?
- This is where machine learning kicks in

Machine Learning and Model Selection: Goodness of Fit

- Now we have some measures of goodness
- That is, the "standard" of what is a "good" model
- Would that be possible to have an automatic algorithm to find a good model for us?
- This is where machine learning kicks in

Machine Learning and Model Selection: Goodness of Fit

- Now we have some measures of goodness
- That is, the "standard" of what is a "good" model
- Would that be possible to have an automatic algorithm to find a good model for us?
- This is where machine learning kicks in

Machine Learning and Model Selection: Goodness of Fit

- Now we have some measures of goodness
- That is, the "standard" of what is a "good" model
- Would that be possible to have an automatic algorithm to find a good model for us?
- This is where machine learning kicks in

Machine Learning and Model Selection: Goodness of Fit

- Now we have some measures of goodness
- That is, the "standard" of what is a "good" model
- Would that be possible to have an automatic algorithm to find a good model for us?
- This is where machine learning kicks in

Machine Learning and Model Selection: Machine Learning

- What is machine learning?
  "Machine learning (ML) is an umbrella term for solving problems for which development of algorithms by human programmers would be cost-prohibitive, and instead the problems are solved by helping machines 'discover' their 'own' algorithms, without needing to be explicitly told what to do by any human-developed algorithms." from *Wikipedia*

Machine Learning and Model Selection: Machine Learning

- What is machine learning?
  "Machine learning (ML) is an umbrella term for solving problems for which development of algorithms by human programmers would be cost-prohibitive, and instead the problems are solved by helping machines 'discover' their 'own' algorithms, without needing to be explicitly told what to do by any human-developed algorithms." from *Wikipedia*

Machine Learning and Model Selection: Machine Learning

Machine learning usage in Economics

- Main target: How complicated the model should be?
- How to *predict* Y given X?
- When Y is discrete: Classification
- When Y is continuous: Prediction
- There are so many machine learning algorithms
- We briefly introduce three of them: Penalized regression, Tree-based method, Neural network

Machine Learning and Model Selection: Machine Learning

Machine learning usage in Economics

- Main target: How complicated the model should be?
- How to *predict* Y given X?
- When Y is discrete: Classification
- When Y is continuous: Prediction
- There are so many machine learning algorithms
- We briefly introduce three of them: Penalized regression, Tree-based method, Neural network

Machine Learning and Model Selection: Machine Learning

Machine learning usage in Economics

- Main target: How complicated the model should be?
- How to *predict* Y given X?
- When Y is discrete: Classification
- When Y is continuous: Prediction
- There are so many machine learning algorithms
- We briefly introduce three of them: Penalized regression, Tree-based method, Neural network

Machine Learning and Model Selection: Machine Learning

Machine learning usage in Economics

- Main target: How complicated the model should be?
- How to *predict* Y given X?
- When Y is discrete: Classification
- When Y is continuous: Prediction
- There are so many machine learning algorithms
- We briefly introduce three of them: Penalized regression, Tree-based method, Neural network

Machine Learning and Model Selection: Machine Learning

Machine learning usage in Economics

- Main target: How complicated the model should be?
- How to *predict* Y given X?
- When Y is discrete: Classification
- When Y is continuous: Prediction
- There are so many machine learning algorithms
- We briefly introduce three of them: Penalized regression, Tree-based method, Neural network

Machine Learning and Model Selection: Machine Learning

Machine learning usage in Economics

- Main target: How complicated the model should be?
- How to *predict* Y given X?
- When Y is discrete: Classification
- When Y is continuous: Prediction
- There are so many machine learning algorithms
- We briefly introduce three of them: Penalized regression, Tree-based method, Neural network

Machine Learning and Model Selection: Machine Learning

Machine learning usage in Economics

- Main target: How complicated the model should be?
- How to *predict* Y given X?
- When Y is discrete: Classification
- When Y is continuous: Prediction
- There are so many machine learning algorithms
- We briefly introduce three of them: Penalized regression, Tree-based method, Neural network

Machine Learning and Model Selection: Machine Learning

Machine learning usage in Economics

- Main target: How complicated the model should be?
- How to *predict* Y given X?
- When Y is discrete: Classification
- When Y is continuous: Prediction
- There are so many machine learning algorithms
- We briefly introduce three of them: Penalized regression, Tree-based method, Neural network

Machine Learning and Model Selection: Penalized Regressions

- Let's consider a linear regression
- What if I have so many potential regressors?
- For instance, you have a household survey with 1000 questions
- Is there an automatic way to select the best predictors?

Machine Learning and Model Selection: Penalized Regressions

- Let's consider a linear regression
- What if I have so many potential regressors?
- For instance, you have a household survey with 1000 questions
- Is there an automatic way to select the best predictors?

Machine Learning and Model Selection: Penalized Regressions

- Let's consider a linear regression
- What if I have so many potential regressors?
- For instance, you have a household survey with 1000 questions
- Is there an automatic way to select the best predictors?

Machine Learning and Model Selection: Penalized Regressions

- Let's consider a linear regression
- What if I have so many potential regressors?
- For instance, you have a household survey with 1000 questions
- Is there an automatic way to select the best predictors?

Machine Learning and Model Selection: Penalized Regressions

- Let's consider a linear regression
- What if I have so many potential regressors?
- For instance, you have a household survey with 1000 questions
- Is there an automatic way to select the best predictors?

Machine Learning and Model Selection: Penalized Regressions

- Linear function: $y_i = x_i'\beta + \epsilon_i$
- OLS: $\hat{\beta}^{OLS} = argmin \sum_i (y_i - x_i'\beta)^2$

  All regressors x play roles.
- We estimate β by minimizing SSR $\Longrightarrow$ More β means smaller SSR
- We need a mechanism to penalize the usage of β

Machine Learning and Model Selection: Penalized Regressions

- Linear function: $y_i = x_i^{\prime}\beta + \epsilon_i$
- OLS: $\hat{\beta}^{OLS} = argmin \sum_i (y_i - x_i^{\prime}\beta)^2$
  All regressors x play roles.
- We estimate β by minimizing SSR $\Longrightarrow$ More β means smaller SSR
- We need a mechanism to penalize the usage of β

Machine Learning and Model Selection: Penalized Regressions

- Linear function: $y_i = x_i^{\prime}\beta + \epsilon_i$
- OLS: $\hat{\beta}^{OLS} = argmin \sum_i (y_i - x_i^{\prime}\beta)^2$
  All regressors x play roles.
- We estimate β by minimizing SSR $\Longrightarrow$ More β means smaller SSR
- We need a mechanism to penalize the usage of β

Machine Learning and Model Selection: Penalized Regressions

- Linear function: $y_i = x_i^{'}\beta + \epsilon_i$
- OLS: $\hat{\beta}^{OLS} = argmin \sum_i (y_i - x_i^{'}\beta)^2$
  All regressors x play roles.
- We estimate β by minimizing SSR $\Rightarrow$ More β means smaller SSR
- We need a mechanism to penalize the usage of β

Machine Learning and Model Selection: Penalized Regressions

- Linear function: $y_i = x_i^{'}\beta + \epsilon_i$
- OLS: $\hat{\beta}^{OLS} = argmin \sum_i (y_i - x_i^{'}\beta)^2$
  All regressors x play roles.
- We estimate β by minimizing SSR $\Rightarrow$ More β means smaller SSR
- We need a mechanism to penalize the usage of β

Machine Learning and Model Selection: Penalized Regressions

- Penalized: $\hat{\beta}^{Pen} = argmin \sum_i (y_i - x_i' \beta)^2 + \lambda(||\beta||_p)^p$
  p=1: Lasso regression, drop some x with small prediction power
  p=2: Ridge regression, shrink some x with small prediction power
- λ: tuning parameter, how strong we penalize additional "x"
- How to choose λ? Cross-validation
- Combination: Elastic Net
  $\hat{\beta}^{Pen} = argmin \sum_i (y_i - x_i' \beta)^2 + \lambda(\alpha||\beta||_1 + (1-\alpha)(||\beta||_2)^2)$

Machine Learning and Model Selection: Penalized Regressions

- Penalized: $\hat{\beta}^{Pen} = argmin \sum_i (y_i - x_i^{\prime}\beta)^2 + \lambda(||\beta||_p)^p$
  p=1: Lasso regression, drop some x with small prediction power
  p=2: Ridge regression, shrink some x with small prediction power
- λ: tuning parameter, how strong we penalize additional "x"
- How to choose λ? Cross-validation
- Combination: Elastic Net
  $\hat{\beta}^{Pen} = argmin \sum_i (y_i - x_i^{\prime}\beta)^2 + \lambda(\alpha||\beta||_1 + (1-\alpha)(||\beta||_2)^2)$

Machine Learning and Model Selection: Penalized Regressions

- Penalized: $\hat{\beta}^{Pen} = argmin \sum_i (y_i - x_i^{\prime}\beta)^2 + \lambda(||\beta||_p)^p$
  p=1: Lasso regression, drop some x with small prediction power
  p=2: Ridge regression, shrink some x with small prediction power
- λ: tuning parameter, how strong we penalize additional "x"
- How to choose λ? Cross-validation
- Combination: Elastic Net
  $\hat{\beta}^{Pen} = argmin \sum_i (y_i - x_i^{\prime}\beta)^2 + \lambda(\alpha||\beta||_1 + (1-\alpha)(||\beta||_2)^2)$

Machine Learning and Model Selection: Penalized Regressions

- Penalized: $\hat{\beta}^{Pen} = argmin \sum_i (y_i - x_i^{\prime}\beta)^2 + \lambda(||\beta||_p)^p$
  p=1: Lasso regression, drop some x with small prediction power
  p=2: Ridge regression, shrink some x with small prediction power
- λ: tuning parameter, how strong we penalize additional "x"
- How to choose λ? Cross-validation
- Combination: Elastic Net
  $\hat{\beta}^{Pen} = argmin \sum_i (y_i - x_i^{\prime}\beta)^2 + \lambda(\alpha||\beta||_1 + (1-\alpha)(||\beta||_2)^2)$

Machine Learning and Model Selection: Penalized Regressions

- Penalized: $\hat{\beta}^{Pen} = argmin \sum_i (y_i - x_i'\beta)^2 + \lambda(||\beta||_p)^p$
  p=1: Lasso regression, drop some x with small prediction power
  p=2: Ridge regression, shrink some x with small prediction power
- λ: tuning parameter, how strong we penalize additional ”x”
- How to choose λ? Cross-validation
- Combination: Elastic Net
  $\hat{\beta}^{Pen} = argmin \sum_i (y_i - x_i'\beta)^2 + \lambda(\alpha||\beta||_1 + (1-\alpha)(||\beta||_2)^2)$

Machine Learning and Model Selection: Tree-based Method

- Tree-based methods partition the feature (X) space into a set of rectangles, and then fit a simple model (constant) in each one.
- Classification and Regression Tree (CART)
- Partition into regions $R_1, R_2 ... R_M$, assign average value in a region as the predicted value
  $\hat{f}(x_i) = \sum_{m=1}^{M} c_m I(x \in R_m)$
- How to partition (Grow the tree)?

Machine Learning and Model Selection: Tree-based Method

- Tree-based methods partition the feature (X) space into a set of rectangles, and then fit a simple model (constant) in each one.
- Classification and Regression Tree (CART)
- Partition into regions $R_1, R_2 ... R_M$, assign average value in a region as the predicted value
  $\hat{f}(x_i) = \sum_{m=1}^{M} c_m I(x \in R_m)$
- How to partition (Grow the tree)?

Machine Learning and Model Selection: Tree-based Method

- Tree-based methods partition the feature (X) space into a set of rectangles, and then fit a simple model (constant) in each one.
- Classification and Regression Tree (CART)
- Partition into regions $R_1, R_2 ... R_M$, assign average value in a region as the predicted value
  $\hat{f}(x_i) = \sum_{m=1}^{M} c_m I(x \in R_m)$
- How to partition (Grow the tree)?

Machine Learning and Model Selection: Tree-based Method

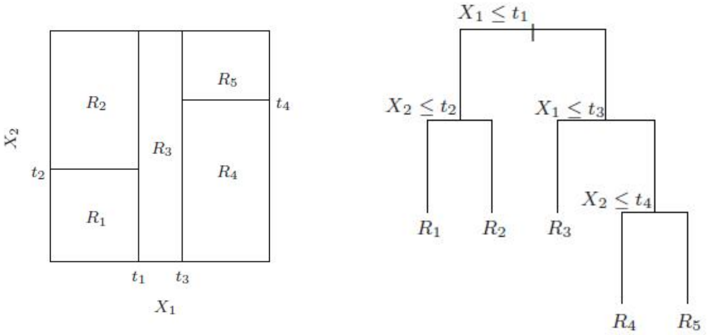
- Tree-based methods partition the feature (X) space into a set of rectangles, and then fit a simple model (constant) in each one.
- Classification and Regression Tree (CART)
- Partition into regions $R_1, R_2 ... R_M$, assign average value in a region as the predicted value
  $\hat{f}(x_i) = \sum_{m=1}^{M} c_m I(x \in R_m)$
- How to partition (Grow the tree)?

Machine Learning and Model Selection: Tree-based Method

- Tree-based methods partition the feature (X) space into a set of rectangles, and then fit a simple model (constant) in each one.
- Classification and Regression Tree (CART)
- Partition into regions $R_1, R_2 ... R_M$, assign average value in a region as the predicted value
  $\hat{f}(x_i) = \sum_{m=1}^{M} c_m I(x \in R_m)$
- How to partition (Grow the tree)?

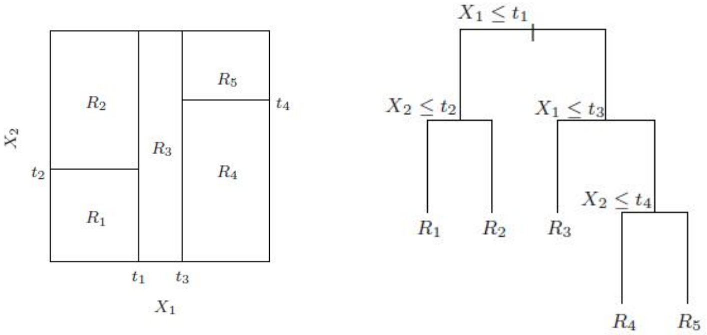
Machine Learning and Model Selection: Tree-based Method

- We use recursive binary partitions
- $(X_1, t_1) \to ((X_2, t_2), (X_1, t_3)) \to (X_2, t_4)$



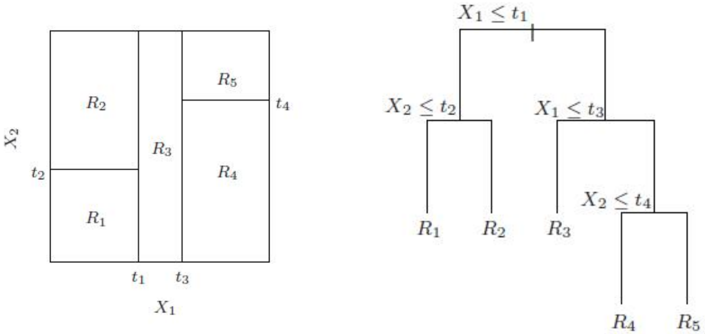
Machine Learning and Model Selection: Tree-based Method

- We use recursive binary partitions
- $(X_1, t_1) \rightarrow ((X_2, t_2), (X_1, t_3)) \rightarrow (X_2, t_4)$



Machine Learning and Model Selection: Tree-based Method

- We use recursive binary partitions
- $(X_1, t_1) \to ((X_2, t_2), (X_1, t_3)) \to (X_2, t_4)$



Machine Learning and Model Selection: Tree-based Method

- Two choices: continue partitioning or stop + where to partition
- Greedy algorithm
- For each region R_m (leaf), we define:

$$\text{Size (\# of obs): } N_m = \{x_i \in R_m\}$$

$$\text{Fitted value (mean as fit): } \hat{c}_m = \frac{1}{N_m} \sum_{x \in R_m} y_i$$

$$\text{SSE (error in leaf): } Q_m(T) = \frac{1}{N_m} \sum_{x \in R_m} (y_i - \hat{c}_m)^2$$

Machine Learning and Model Selection: Tree-based Method

- Two choices: continue partitioning or stop + where to partition
- Greedy algorithm
- For each region R_m (leaf), we define:

$$\text{Size (\# of obs): } N_m = \{x_i \in R_m\}$$

$$\text{Fitted value (mean as fit): } \hat{c}_m = \frac{1}{N_m} \sum_{x \in R_m} y_i$$

$$\text{SSE (error in leaf): } Q_m(T) = \frac{1}{N_m} \sum_{x \in R_m} (y_i - \hat{c}_m)^2$$

Machine Learning and Model Selection: Tree-based Method

- Two choices: continue partitioning or stop + where to partition
- Greedy algorithm
- For each region R_m (leaf), we define:

$$\text{Size (\# of obs)}: \; N_m = \{x_i \in R_m\}$$

$$\text{Fitted value (mean as fit)}: \; \hat{c}_m = \frac{1}{N_m} \sum_{x \in R_m} y_i$$

$$\text{SSE (error in leaf)}: \; Q_m(T) = \frac{1}{N_m} \sum_{x \in R_m} (y_i - \hat{c}_m)^2$$

Machine Learning and Model Selection: Tree-based Method

- Two choices: continue partitioning or stop + where to partition
- Greedy algorithm
- For each region R_m (leaf), we define:

$$\text{Size (\# of obs):}\ N_m = \{x_i \in R_m\}$$

$$\text{Fitted value (mean as fit):}\ \hat{c}_m = \frac{1}{N_m} \sum_{x \in R_m} y_i$$

$$\text{SSE (error in leaf):}\ Q_m(T) = \frac{1}{N_m} \sum_{x \in R_m} (y_i - \hat{c}_m)^2$$

Machine Learning and Model Selection: Tree-based Method

- First, conditional on continuing grow, how to determine partition?
- For $j-th$ predictor, cut position s
- Define half plane $R_1(j,s)=\{X|X_j\leq s\}, R_2(j,s)=\{X|X_j>s\}$
- How to find (j,s) in each branch? Minimize SSE (Easy)

$$\min_{j,s}[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2]$$

- Here c_1 and c_2 are conditional means (in leaf 1 and 2)

Machine Learning and Model Selection: Tree-based Method

- First, conditional on continuing grow, how to determine partition?
- For $j - th$ predictor, cut position s
- Define half plane $R_1(j,s) = \{X|X_j \leq s\}, R_2(j,s) = \{X|X_j > s\}$
- How to find (j,s) in each branch? Minimize SSE (Easy)

$$\min_{j,s}[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2]$$

- Here c_1 and c_2 are conditional means (in leaf 1 and 2)

Machine Learning and Model Selection: Tree-based Method

- First, conditional on continuing grow, how to determine partition?
- For $j - th$ predictor, cut position s
- Define half plane $R_1(j,s) = \{X|X_j \leq s\}, R_2(j,s) = \{X|X_j > s\}$
- How to find (j,s) in each branch? Minimize SSE (Easy)

$$\min_{j,s}[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2]$$

- Here c_1 and c_2 are conditional means (in leaf 1 and 2)

Machine Learning and Model Selection: Tree-based Method

- First, conditional on continuing grow, how to determine partition?
- For $j - th$ predictor, cut position s
- Define half plane $R_1(j,s) = \{X|X_j \leq s\}, R_2(j,s) = \{X|X_j > s\}$
- How to find (j,s) in each branch? Minimize SSE (Easy)

$$\min_{j,s}[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2]$$

- Here c_1 and c_2 are conditional means (in leaf 1 and 2)

Machine Learning and Model Selection: Tree-based Method

- First, conditional on continuing grow, how to determine partition?
- For $j - th$ predictor, cut position s
- Define half plane $R_1(j,s) = \{X|X_j \leq s\}, R_2(j,s) = \{X|X_j > s\}$
- How to find (j,s) in each branch? Minimize SSE (Easy)

$$\min_{j,s}[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2]$$

- Here c_1 and c_2 are conditional means (in leaf 1 and 2)

Machine Learning and Model Selection: Tree-based Method

- First, conditional on continuing grow, how to determine partition?
- For $j - th$ predictor, cut position s
- Define half plane $R_1(j,s) = \{X|X_j \leq s\}, R_2(j,s) = \{X|X_j > s\}$
- How to find (j,s) in each branch? Minimize SSE (Easy)

$$\min_{j,s}[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2]$$

- Here c_1 and c_2 are conditional means (in leaf 1 and 2)

Machine Learning and Model Selection: Tree-based Method

- Second, how to choose to continue growing the tree or stop?
- Too large → Overfitting; Too small → Losing information
- Grow a big tree T_0, then prune it!
  - Step 1: Grow T_0 when some minimum node size is reached (say 10)
  - Step 2: Pruning. Choose the tree $T \subset T_0$ with the lowest cost function $C_\alpha(T)$
  - $T \subset T_0$ means any tree T that can be obtained by collapsing any number of internal nodes in T_0

Machine Learning and Model Selection: Tree-based Method

- Second, how to choose to continue growing the tree or stop?
- Too large → Overfitting; Too small → Losing information
- Grow a big tree T_0, then prune it!
  - Step 1: Grow T_0 when some minimum node size is reached (say 10)
  - Step 2: Pruning. Choose the tree $T \subset T_0$ with the lowest cost function $C_\alpha(T)$.
  - $T \subset T_0$ means any tree T that can be obtained by collapsing any number of internal nodes in T_0

Machine Learning and Model Selection: Tree-based Method

- Second, how to choose to continue growing the tree or stop?
- Too large → Overfitting; Too small → Losing information
- Grow a big tree T_0, then prune it!
    - Step 1: Grow T_0 when some minimum node size is reached (say 10)
    - Step 2: Pruning. Choose the tree $T \subset T_0$ with the lowest cost function $C_\alpha(T)$.
    - $T \subset T_0$ means any tree T that can be obtained by collapsing any number of internal nodes in T_0

Machine Learning and Model Selection: Tree-based Method

- Second, how to choose to continue growing the tree or stop?
- Too large → Overfitting; Too small → Losing information
- Grow a big tree T_0, then prune it!
    - Step 1: Grow T_0 when some minimum node size is reached (say 10)
    - Step 2: Pruning. Choose the tree $T \subset T_0$ with the lowest cost function $C_\alpha(T)$.
    - $T \subset T_0$ means any tree T that can be obtained by collapsing any number of internal nodes in T_0

Machine Learning and Model Selection: Tree-based Method

- Second, how to choose to continue growing the tree or stop?
- Too large → Overfitting; Too small → Losing information
- Grow a big tree T_0, then prune it!
  - Step 1: Grow T_0 when some minimum node size is reached (say 10)
  - Step 2: Pruning. Choose the tree $T \subset T_0$ with the lowest cost function $C_\alpha(T)$.
  - $T \subset T_0$ means any tree T that can be obtained by collapsing any number of internal nodes in T_0

Machine Learning and Model Selection: Tree-based Method

- Second, how to choose to continue growing the tree or stop?
- Too large → Overfitting; Too small → Losing information
- Grow a big tree T_0, then prune it!
  - Step 1: Grow T_0 when some minimum node size is reached (say 10)
  - Step 2: Pruning. Choose the tree $T \subset T_0$ with the lowest cost function $C_\alpha(T)$.
  - $T \subset T_0$ means any tree T that can be obtained by collapsing any number of internal nodes in T_0

Machine Learning and Model Selection: Tree-based Method

- Second, how to choose to continue growing the tree or stop?
- Too large → Overfitting; Too small → Losing information
- Grow a big tree T_0, then prune it!
  - Step 1: Grow T_0 when some minimum node size is reached (say 10)
  - Step 2: Pruning. Choose the tree $T \subset T_0$ with the lowest cost function $C_\alpha(T)$.
  - $T \subset T_0$ means any tree T that can be obtained by collapsing any number of internal nodes in T_0

Machine Learning and Model Selection: Tree-based Method

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$$

α as the tuning parameter; $|T|$ as number of terminal nodes

- Total SSE (bias) + Size penalty
- α determines how hard to penalize tree size

Machine Learning and Model Selection: Tree-based Method

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$$

α as the tuning parameter; $|T|$ as number of terminal nodes

- Total SSE (bias) + Size penalty
- α determines how hard to penalize tree size

Machine Learning and Model Selection: Tree-based Method

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$$

α as the tuning parameter; $|T|$ as number of terminal nodes

- Total SSE (bias) + Size penalty
- α determines how hard to penalize tree size

Machine Learning and Model Selection: Random Forests

- Using sub-sampling or bagging to reduce variance of a single tree
- Draw a lot of different samples (1,2,...B) with sub-sampling ($n < N$) (Jackknife) or bagging ($n = N$) (Bootstrap)
- De-correlation: In each split, randomly select m variables to do the partition

$$\hat{f}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

$$V(\hat{f}) \approx \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- Random Forests = Tree Method + Sampling average (Many De-correlated Trees)
- To reduce $V(\hat{f})$: $B \uparrow$ (more sampling), $\rho \downarrow$ (smaller correlation)

Machine Learning and Model Selection: Random Forests

- Using sub-sampling or bagging to reduce variance of a single tree
- Draw a lot of different samples (1,2,...B) with sub-sampling ($n < N$) (Jackknife) or bagging ($n = N$) (Bootstrap)
- De-correlation: In each split, randomly select m variables to do the partition

$$\hat{f}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

$$V(\hat{f}) \approx \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- Random Forests = Tree Method + Sampling average (Many De-correlated Trees)
- To reduce $V(\hat{f})$: $B \uparrow$ (more sampling), $\rho \downarrow$ (smaller correlation)

Machine Learning and Model Selection: Random Forests

- Using sub-sampling or bagging to reduce variance of a single tree
- Draw a lot of different samples (1,2,...B) with sub-sampling ($n < N$) (Jackknife) or bagging ($n = N$) (Bootstrap)
- De-correlation: In each split, randomly select m variables to do the partition

$$\hat{f}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

$$V(\hat{f}) \approx \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- Random Forests = Tree Method + Sampling average (Many De-correlated Trees)
- To reduce $V(\hat{f})$: $B \uparrow$ (more sampling), $\rho \downarrow$ (smaller correlation)

Machine Learning and Model Selection: Random Forests

- Using sub-sampling or bagging to reduce variance of a single tree
- Draw a lot of different samples (1,2,...B) with sub-sampling ($n < N$) (Jackknife) or bagging ($n = N$) (Bootstrap)
- De-correlation: In each split, randomly select m variables to do the partition

$$\hat{f}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

$$V(\hat{f}) \approx \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- Random Forests = Tree Method + Sampling average (Many De-correlated Trees)
- To reduce $V(\hat{f})$: $B \uparrow$ (more sampling), $\rho \downarrow$ (smaller correlation)

Machine Learning and Model Selection: Random Forests

- Using sub-sampling or bagging to reduce variance of a single tree
- Draw a lot of different samples (1,2,...B) with sub-sampling ($n < N$) (Jackknife) or bagging ($n = N$) (Bootstrap)
- De-correlation: In each split, randomly select m variables to do the partition

$$\hat{f}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

$$V(\hat{f}) \approx \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- Random Forests = Tree Method + Sampling average (Many De-correlated Trees)
- To reduce $V(\hat{f})$: $B \uparrow$ (more sampling), $\rho \downarrow$ (smaller correlation)

Machine Learning and Model Selection: Random Forests

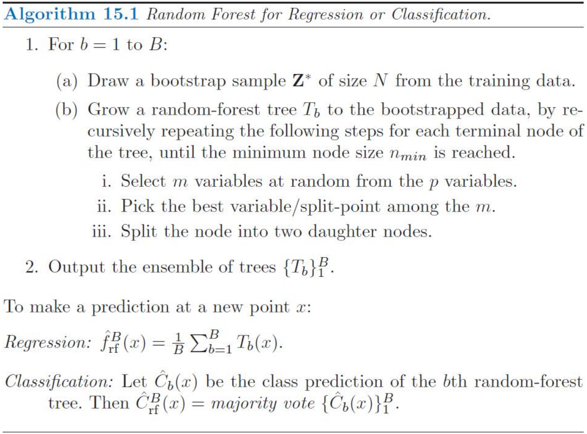
- Using sub-sampling or bagging to reduce variance of a single tree
- Draw a lot of different samples (1,2,...B) with sub-sampling ($n < N$) (Jackknife) or bagging ($n = N$) (Bootstrap)
- De-correlation: In each split, randomly select m variables to do the partition

$$\hat{f}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

$$V(\hat{f}) \approx \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- Random Forests $=$ Tree Method $+$ Sampling average (Many De-correlated Trees)
- To reduce $V(\hat{f})$: $B \uparrow$ (more sampling), $\rho \downarrow$ (smaller correlation)

Machine Learning and Model Selection: Random Forests



Algorithm 15.1 *Random Forest for Regression or Classification.*

```
1. For $b = 1$ to $B$:
    (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size $N$ from the training data.
    (b) Grow a random-forest tree $T_b$ to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{min}$ is reached.
        i. Select $m$ variables at random from the $p$ variables.
        ii. Pick the best variable/split-point among the $m$.
        iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.
```

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

Machine Learning and Model Selection: Random Forests

- We reduce the variance by bagging (B) and de-correlation (ρ)
- This is a method similar to kernels and nearest-neighbor method
  Making predictions using weighted averages of "nearby" observations
- Difference: Weighting scheme
  Nearest Neighbor: Not adaptive; Random Forests: Adaptive
- An important application of Random Forests in Economics is Causal Forests

Machine Learning and Model Selection: Random Forests

- We reduce the variance by bagging (B) and de-correlation (ρ)
- This is a method similar to kernels and nearest-neighbor method
  Making predictions using weighted averages of ”nearby” observations
- Difference: Weighting scheme
  Nearest Neighbor: Not adaptive; Random Forests: Adaptive
- An important application of Random Forests in Economics is Causal Forests

Machine Learning and Model Selection: Random Forests

- We reduce the variance by bagging (B) and de-correlation (ρ)
- This is a method similar to kernels and nearest-neighbor method
  Making predictions using weighted averages of "nearby" observations
- Difference: Weighting scheme
  Nearest Neighbor: Not adaptive; Random Forests: Adaptive
- An important application of Random Forests in Economics is Causal Forests

Machine Learning and Model Selection: Random Forests

- We reduce the variance by bagging (B) and de-correlation (ρ)
- This is a method similar to kernels and nearest-neighbor method
  Making predictions using weighted averages of "nearby" observations
- Difference: Weighting scheme
  Nearest Neighbor: Not adaptive; Random Forests: Adaptive
- An important application of Random Forests in Economics is Causal Forests

Machine Learning and Model Selection: Random Forests

- We reduce the variance by bagging (B) and de-correlation (ρ)
- This is a method similar to kernels and nearest-neighbor method
  Making predictions using weighted averages of "nearby" observations
- Difference: Weighting scheme
  Nearest Neighbor: Not adaptive; Random Forests: Adaptive
- An important application of Random Forests in Economics is Causal Forests

Machine Learning and Model Selection: Causal Forests

- Main topic in causal inference: Treatment effect
  Mostly ATE, LATE etc.
- Heterogeneous Treatment Effect
  Cherry picking? $\Longrightarrow$ Institutional restrictions on trials
- Unexpected heterogeneity
- Wager and Athey develop a machine learning tool, Causal Forests (An extension of Random Forests)
- To reveal the true underlying heterogeneous treatment effects

Machine Learning and Model Selection: Causal Forests

- Main topic in causal inference: Treatment effect
  Mostly ATE, LATE etc.
- Heterogeneous Treatment Effect
  Cherry picking? $\Longrightarrow$ Institutional restrictions on trials
- Unexpected heterogeneity
- Wager and Athey develop a machine learning tool, Causal Forests (An extension of Random Forests)
- To reveal the true underlying heterogeneous treatment effects

Machine Learning and Model Selection: Causal Forests

- Main topic in causal inference: Treatment effect
  Mostly ATE, LATE etc.
- Heterogeneous Treatment Effect
  Cherry picking? ⇒ Institutional restrictions on trials
- Unexpected heterogeneity
- Wager and Athey develop a machine learning tool, Causal Forests (An extension of Random Forests)
- To reveal the true underlying heterogeneous treatment effects

Machine Learning and Model Selection: Causal Forests

- Main topic in causal inference: Treatment effect
  Mostly ATE, LATE etc.
- Heterogeneous Treatment Effect
  Cherry picking? $\Longrightarrow$ Institutional restrictions on trials
- Unexpected heterogeneity
- Wager and Athey develop a machine learning tool, Causal Forests (An extension of Random Forests)
- To reveal the true underlying heterogeneous treatment effects

Machine Learning and Model Selection: Causal Forests

- Main topic in causal inference: Treatment effect
  Mostly ATE, LATE etc.
- Heterogeneous Treatment Effect
  Cherry picking? ⇒ Institutional restrictions on trials
- Unexpected heterogeneity
- Wager and Athey develop a machine learning tool, Causal Forests (An extension of Random Forests)
- To reveal the true underlying heterogeneous treatment effects

Machine Learning and Model Selection: Causal Forests

- Main topic in causal inference: Treatment effect
  Mostly ATE, LATE etc.
- Heterogeneous Treatment Effect
  Cherry picking? $\Rightarrow$ Institutional restrictions on trials
- Unexpected heterogeneity
- Wager and Athey develop a machine learning tool, Causal Forests (An extension of Random Forests)
- To reveal the true underlying heterogeneous treatment effects

Machine Learning and Model Selection: Causal Forests

- It tells us how to divide groups to get the ”real” heterogeneous TE
- Data of (X_i, Y_i, W_i), W_i is treatment assignment. L as a leaf (region).
- Treatment effect: $\tau(x) = E[Y_i^{(1)} - Y_i^{(0)} | X_i = x]$
- Unconfoundness: $\{Y_i^{(0)}, Y_i^{(1)}\} \perp W_i | X_i$
- Tips: We assume unconfoundness here, which means that causal forests is not a method to deal with endogeneity issue

Machine Learning and Model Selection: Causal Forests

- It tells us how to divide groups to get the "real" heterogeneous TE
- Data of (X_i, Y_i, W_i), W_i is treatment assignment. L as a leaf (region).
- Treatment effect: $\tau(x) = E[Y_i^{(1)} - Y_i^{(0)} | X_i = x]$
- Unconfoundness: $\{Y_i^{(0)}, Y_i^{(1)}\} \perp W_i | X_i$
- Tips: We assume unconfoundness here, which means that causal forests is not a method to deal with endogeneity issue

Machine Learning and Model Selection: Causal Forests

- It tells us how to divide groups to get the "real" heterogeneous TE
- Data of (X_i, Y_i, W_i), W_i is treatment assignment. L as a leaf (region).
- Treatment effect: $\tau(x) = E[Y_i^{(1)} - Y_i^{(0)} | X_i = x]$
- Unconfoundness: $\{Y_i^{(0)}, Y_i^{(1)}\} \perp W_i | X_i$
- Tips: We assume unconfoundness here, which means that causal forests is not a method to deal with endogeneity issue

Machine Learning and Model Selection: Causal Forests

- It tells us how to divide groups to get the "real" heterogeneous TE
- Data of (X_i, Y_i, W_i), W_i is treatment assignment. L as a leaf (region).
- Treatment effect: $\tau(x) = E[Y_i^{(1)} - Y_i^{(0)} | X_i = x]$
- Unconfoundness: $\{Y_i^{(0)}, Y_i^{(1)}\} \perp W_i | X_i$
- Tips: We assume unconfoundness here, which means that causal forests is not a method to deal with endogeneity issue

Machine Learning and Model Selection: Causal Forests

- It tells us how to divide groups to get the "real" heterogeneous TE
- Data of (X_i, Y_i, W_i), W_i is treatment assignment. L as a leaf (region).
- Treatment effect: $\tau(x) = E[Y_i^{(1)} - Y_i^{(0)} | X_i = x]$
- Unconfoundness: $\{Y_i^{(0)}, Y_i^{(1)}\} \perp W_i | X_i$
- Tips: We assume unconfoundness here, which means that causal forests is not a method to deal with endogeneity issue

Machine Learning and Model Selection: Causal Forests

- It tells us how to divide groups to get the "real" heterogeneous TE
- Data of (X_i, Y_i, W_i), W_i is treatment assignment. L as a leaf (region).
- Treatment effect: $\tau(x) = E[Y_i^{(1)} - Y_i^{(0)} | X_i = x]$
- Unconfoundness: $\{Y_i^{(0)}, Y_i^{(1)}\} \perp W_i | X_i$
- Tips: We assume unconfoundness here, which means that causal forests is not a method to deal with endogeneity issue

Machine Learning and Model Selection: Causal Forests

- Estimation of TE: Given x in leaf $L(x)$, the difference of the average outcome Y for treated/non-treated group

$$\hat{\tau}(x) = \frac{1}{|\{i:W_i=1,X_i\in L\}|}\sum_{\{i:W_i=1,X_i\in L\}}^{Y_i} Y_i - \frac{1}{|\{i:W_i=0,X_i\in L\}|}\sum_{\{i:W_i=0,X_i\in L\}}^{Y_i} Y_i$$

- Implement the Random Forests using a criterion: maximizing variance of $\hat{\tau}(X_i)$

Machine Learning and Model Selection: Causal Forests

- Estimation of TE: Given x in leaf $L(x)$, the difference of the average outcome Y for treated/non-treated group

$$\hat{\tau}(x) = \frac{1}{|\{i:W_i=1,X_i\in L\}|}\sum_{\{i:W_i=1,X_i\in L\}}^{Y_i} Y_i - \frac{1}{|\{i:W_i=0,X_i\in L\}|}\sum_{\{i:W_i=0,X_i\in L\}}^{Y_i} Y_i$$

- Implement the Random Forests using a criterion: maximizing variance of $\hat{\tau}(X_i)$

Machine Learning and Model Selection: Causal Forests

- Estimation of TE: Given x in leaf $L(x)$, the difference of the average outcome Y for treated/non-treated group

$$\hat{\tau}(x) = \frac{1}{|\{i:W_i=1,X_i\in L\}|}\sum_{\{i:W_i=1,X_i\in L\}}^{Y_i} Y_i - \frac{1}{|\{i:W_i=0,X_i\in L\}|}\sum_{\{i:W_i=0,X_i\in L\}}^{Y_i} Y_i$$

- Implement the Random Forests using a criterion: maximizing variance of $\hat{\tau}(X_i)$

Machine Learning and Model Selection: Causal Forests

- A tree is honesty, if for each training sample i, it is either used to estimate τ or used to decide splits
- Double-Sample Trees: Averagely divide samples into two parts I and J. Grow the tree using I and then estimate τ in each leaf using J.
- Honest Causal Forests is consistent and asymptotically normal

Machine Learning and Model Selection: Causal Forests

- A tree is honesty, if for each training sample i, it is either used to estimate τ or used to decide splits
- Double-Sample Trees: Averagely divide samples into two parts I and J. Grow the tree using I and then estimate τ in each leaf using J.
- Honest Causal Forests is consistent and asymptotically normal

Machine Learning and Model Selection: Causal Forests

- A tree is honesty, if for each training sample i, it is either used to estimate τ or used to decide splits
- Double-Sample Trees: Averagely divide samples into two parts I and J. Grow the tree using I and then estimate τ in each leaf using J.
- Honest Causal Forests is consistent and asymptotically normal

Machine Learning and Model Selection: Causal Forests

- A tree is honesty, if for each training sample i, it is either used to estimate τ or used to decide splits
- Double-Sample Trees: Averagely divide samples into two parts I and J. Grow the tree using I and then estimate τ in each leaf using J.
- Honest Causal Forests is consistent and asymptotically normal

Machine Learning and Model Selection: Application of Causal Forests

- Paper report
  Levy (2021) Social Media, News Consumption, and Polarization: Evidence from a Field Experiment
- Please also read Online Appendix C.5

Machine Learning and Model Selection: Application of Causal Forests

- Paper report
  Levy (2021) Social Media, News Consumption, and Polarization: Evidence from a Field Experiment
- Please also read Online Appendix C.5

Machine Learning and Model Selection: Application of Causal Forests

- Paper report
  Levy (2021) Social Media, News Consumption, and Polarization: Evidence from a Field Experiment
- Please also read Online Appendix C.5

Machine Learning and Model Selection: Neural Networks

- Another widely used machine learning method is Neural Networks
- It attracts people's attention during these days in media
- AI, Chatgpt, AlphaGo...Sky Net

Machine Learning and Model Selection: Neural Networks

- Another widely used machine learning method is Neural Networks
- It attracts people's attention during these days in media
- AI, Chatgpt, AlphaGo...Sky Net

Machine Learning and Model Selection: Neural Networks

- Another widely used machine learning method is Neural Networks
- It attracts people's attention during these days in media
- AI, Chatgpt, AlphaGo...Sky Net

Machine Learning and Model Selection: Neural Networks

- Another widely used machine learning method is Neural Networks
- It attracts people's attention during these days in media
- AI, Chatgpt, AlphaGo...Sky Net

Machine Learning and Model Selection: Neural Networks

- Consider a single layer classification model, where Y_k refers to each choice/class

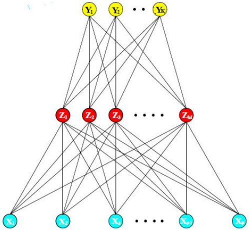


FIGURE 11.2. *Schematic of a single hidden layer, feed-forward neural network.*

- X - Input; Z- Hidden layer/unit; Y - Output

Machine Learning and Model Selection: Neural Networks

- Consider a single layer classification model, where Y_k refers to each choice/class

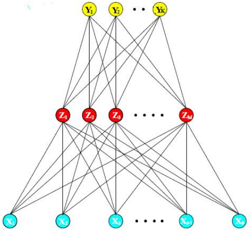


FIGURE 11.2. *Schematic of a single hidden layer, feed-forward neural network.*

- X - Input; Z- Hidden layer/unit; Y - Output

Machine Learning and Model Selection: Neural Networks

- Consider a single layer classification model, where Y_k refers to each choice/class

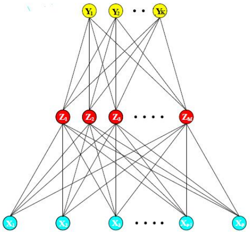


FIGURE 11.2. *Schematic of a single hidden layer, feed-forward neural network.*

- X - Input; Z- Hidden layer/unit; Y - Output

Machine Learning and Model Selection: Neural Networks

- Step 1: from input X to hidden unit Z

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), m = 1, ..., M$$

- σ is a nonlinear function (Step or Logit)
- This nonlinearity is important: make NN differ from linear regression
- It is called the activation function
- Step 2: from hidden unit Z to output Y

$$T_k = \beta_{0k} + \beta_k^T Z, k = 1, ..., K$$
$$f_k(X) = g_k(T), k = 1, ..., K$$

- g is a nonlinear function (Step or Logit)

Machine Learning and Model Selection: Neural Networks

- Step 1: from input X to hidden unit Z

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), m = 1, ..., M$$

- σ is a nonlinear function (Step or Logit)
- This nonlinearity is important: make NN differ from linear regression
- It is called the activation function
- Step 2: from hidden unit Z to output Y

$$T_k = \beta_{0k} + \beta_k^T Z, k = 1, ..., K$$
$$f_k(X) = g_k(T), k = 1, ..., K$$

- g is a nonlinear function (Step or Logit)

Machine Learning and Model Selection: Neural Networks

- Step 1: from input X to hidden unit Z

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), m = 1, ..., M$$

- σ is a nonlinear function (Step or Logit)
- This nonlinearity is important: make NN differ from linear regression
- It is called the activation function
- Step 2: from hidden unit Z to output Y

$$T_k = \beta_{0k} + \beta_k^T Z, k = 1, ..., K$$
$$f_k(X) = g_k(T), k = 1, ..., K$$

- g is a nonlinear function (Step or Logit)

Machine Learning and Model Selection: Neural Networks

- Step 1: from input X to hidden unit Z

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), m = 1, ..., M$$

- σ is a nonlinear function (Step or Logit)
- This nonlinearity is important: make NN differ from linear regression
- It is called the activation function
- Step 2: from hidden unit Z to output Y

$$T_k = \beta_{0k} + \beta_k^T Z, k = 1, ..., K$$
$$f_k(X) = g_k(T), k = 1, ..., K$$

- g is a nonlinear function (Step or Logit)

Machine Learning and Model Selection: Neural Networks

- Step 1: from input X to hidden unit Z

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), m = 1, \ldots, M$$

- σ is a nonlinear function (Step or Logit)
- This nonlinearity is important: make NN differ from linear regression
- It is called the activation function
- Step 2: from hidden unit Z to output Y

$$T_k = \beta_{0k} + \beta_k^T Z, k = 1, \ldots, K$$
$$f_k(X) = g_k(T), k = 1, \ldots, K$$

- g is a nonlinear function (Step or Logit)

Machine Learning and Model Selection: Neural Networks

- Step 1: from input X to hidden unit Z

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), m = 1, ..., M$$

- σ is a nonlinear function (Step or Logit)
- This nonlinearity is important: make NN differ from linear regression
- It is called the activation function
- Step 2: from hidden unit Z to output Y

$$T_k = \beta_{0k} + \beta_k^T Z, k = 1, ..., K$$
$$f_k(X) = g_k(T), k = 1, ..., K$$

- g is a nonlinear function (Step or Logit)

Machine Learning and Model Selection: Neural Networks

- Step 1: from input X to hidden unit Z

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X), m = 1, ..., M$$

- σ is a nonlinear function (Step or Logit)
- This nonlinearity is important: make NN differ from linear regression
- It is called the activation function
- Step 2: from hidden unit Z to output Y

$$\begin{aligned} T_k &= \beta_{0k} + \beta_k^T Z, k = 1, ..., K \\ f_k(X) &= g_k(T), k = 1, ..., K \end{aligned}$$

- g is a nonlinear function (Step or Logit)

Machine Learning and Model Selection: Neural Networks

- Why do we call this Neural Networks?
- Because it was first developed as models for the human brain
- Each unit represents a neuron
- Connections are synapses
- There can be multiple layers
- When step function is used for σ and g, neurons fire when signal passed to the unit $(\alpha_{0m} + \alpha_m^T X)$ exceeds some threshold

Machine Learning and Model Selection: Neural Networks

- Why do we call this Neural Networks?
- Because it was first developed as models for the human brain
- Each unit represents a neuron
- Connections are synapses
- There can be multiple layers
- When step function is used for σ and g, neurons fire when signal passed to the unit $(\alpha_{0m} + \alpha_m^T X)$ exceeds some threshold

Machine Learning and Model Selection: Neural Networks

- Why do we call this Neural Networks?
- Because it was first developed as models for the human brain
- Each unit represents a neuron
- Connections are synapses
- There can be multiple layers
- When step function is used for σ and g, neurons fire when signal passed to the unit $(\alpha_{0m} + \alpha_m^T X)$ exceeds some threshold

Machine Learning and Model Selection: Neural Networks

- Why do we call this Neural Networks?
- Because it was first developed as models for the human brain
- Each unit represents a neuron
- Connections are synapses
- There can be multiple layers
- When step function is used for σ and g, neurons fire when signal passed to the unit $(\alpha_{0m} + \alpha_m^T X)$ exceeds some threshold

Machine Learning and Model Selection: Neural Networks

- Why do we call this Neural Networks?
- Because it was first developed as models for the human brain
- Each unit represents a neuron
- Connections are synapses
- There can be multiple layers
- When step function is used for σ and g, neurons fire when signal passed to the unit $(\alpha_{0m} + \alpha_m^T X)$ exceeds some threshold

Machine Learning and Model Selection: Neural Networks

- Why do we call this Neural Networks?
- Because it was first developed as models for the human brain
- Each unit represents a neuron
- Connections are synapses
- There can be multiple layers
- When step function is used for σ and g, neurons fire when signal passed to the unit $(\alpha_{0m} + \alpha_m^T X)$ exceeds some threshold

Machine Learning and Model Selection: Neural Networks

- Why do we call this Neural Networks?
- Because it was first developed as models for the human brain
- Each unit represents a neuron
- Connections are synapses
- There can be multiple layers
- When step function is used for σ and g, neurons fire when signal passed to the unit $(\alpha_{0m} + \alpha_m^T X)$ exceeds some threshold

Machine Learning and Model Selection: Neural Networks

- How to estimate this model?
- Simply nonlinear Least Square
- How to avoid overfitting?
- Regularize the optimization problem min $R(\theta)$ with a penalty term:

$$\min R(\theta) + \lambda J(\theta)$$

$$J(\theta) = \sum_{km} \beta_{km}^2 + \sum_{ml} \alpha_{mp}^2$$

- λ is a tuning parameter

Machine Learning and Model Selection: Neural Networks

- How to estimate this model?
- Simply nonlinear Least Square
- How to avoid overfitting?
- Regularize the optimization problem min $R(\theta)$ with a penalty term:

$$\min R(\theta) + \lambda J(\theta)$$
$$J(\theta) = \sum_{km} \beta_{km}^2 + \sum_{ml} \alpha_{mp}^2$$

- λ is a tuning parameter

Machine Learning and Model Selection: Neural Networks

- How to estimate this model?
- Simply nonlinear Least Square
- How to avoid overfitting?
- Regularize the optimization problem min $R(\theta)$ with a penalty term:

$$\min R(\theta) + \lambda J(\theta)$$
$$J(\theta) = \sum_{km} \beta_{km}^2 + \sum_{ml} \alpha_{mp}^2$$

- λ is a tuning parameter

Machine Learning and Model Selection: Neural Networks

- How to estimate this model?
- Simply nonlinear Least Square
- How to avoid overfitting?
- Regularize the optimization problem min $R(\theta)$ with a penalty term:

$$\min R(\theta) + \lambda J(\theta)$$

$$J(\theta) = \sum_{km} \beta_{km}^2 + \sum_{ml} \alpha_{mp}^2$$

- λ is a tuning parameter

Machine Learning and Model Selection: Neural Networks

- How to estimate this model?
- Simply nonlinear Least Square
- How to avoid overfitting?
- Regularize the optimization problem min $R(\theta)$ with a penalty term:

$$\min R(\theta) + \lambda J(\theta)$$

$$J(\theta) = \sum_{km} \beta_{km}^2 + \sum_{ml} \alpha_{mp}^2$$

- λ is a tuning parameter

Machine Learning and Model Selection: Neural Networks

- How to estimate this model?
- Simply nonlinear Least Square
- How to avoid overfitting?
- Regularize the optimization problem min $R(\theta)$ with a penalty term:

$$\min R(\theta) + \lambda J(\theta)$$
$$J(\theta) = \sum_{km} \beta_{km}^2 + \sum_{ml} \alpha_{mp}^2$$

- λ is a tuning parameter

Machine Learning and Model Selection: Conclusion

- Model complexity is double-edged: Bias-variance tradeoff
- In general, there are many standards to evaluate model's goodness-of-fit
  CV, AIC, BIC
- Machine learning gives you automatic algorithms to select model
  Penalized regression, Tree-based method (Random Forests), Neural Networks
- An important new application in economics is Causal Forests
  Can be used to detect heterogeneous treatment effect

Machine Learning and Model Selection: Conclusion

- Model complexity is double-edged: Bias-variance tradeoff
- In general, there are many standards to evaluate model's goodness-of-fit
  CV, AIC, BIC
- Machine learning gives you automatic algorithms to select model
  Penalized regression, Tree-based method (Random Forests), Neural Networks
- An important new application in economics is Causal Forests
  Can be used to detect heterogeneous treatment effect

Machine Learning and Model Selection: Conclusion

- Model complexity is double-edged: Bias-variance tradeoff
- In general, there are many standards to evaluate model's goodness-of-fit
  CV, AIC, BIC
- Machine learning gives you automatic algorithms to select model
  Penalized regression, Tree-based method (Random Forests), Neural Networks
- An important new application in economics is Causal Forests
  Can be used to detect heterogeneous treatment effect

Machine Learning and Model Selection: Conclusion

- Model complexity is double-edged: Bias-variance tradeoff
- In general, there are many standards to evaluate model's goodness-of-fit
  CV, AIC, BIC
- Machine learning gives you automatic algorithms to select model
  Penalized regression, Tree-based method (Random Forests), Neural Networks
- An important new application in economics is Causal Forests
  Can be used to detect heterogeneous treatment effect

Machine Learning and Model Selection: Conclusion

- Model complexity is double-edged: Bias-variance tradeoff
- In general, there are many standards to evaluate model's goodness-of-fit
  CV, AIC, BIC
- Machine learning gives you automatic algorithms to select model
  Penalized regression, Tree-based method (Random Forests), Neural Networks
- An important new application in economics is Causal Forests
  Can be used to detect heterogeneous treatment effect

Machine Learning and Model Selection: Conclusion

- But remember, these are only statistical tools
- The most important method is still your ECONOMIC intuition!
- Never exclude education from a wage equation, even if AIC/BIC told you so!

Machine Learning and Model Selection: Conclusion

- But remember, these are only statistical tools
- The most important method is still your ECONOMIC intuition!
- Never exclude education from a wage equation, even if AIC/BIC told you so!

Machine Learning and Model Selection: Conclusion

- But remember, these are only statistical tools
- The most important method is still your ECONOMIC intuition!
- Never exclude education from a wage equation, even if AIC/BIC told you so!

Machine Learning and Model Selection: Conclusion

- But remember, these are only statistical tools
- The most important method is still your ECONOMIC intuition!
- Never exclude education from a wage equation, even if AIC/BIC told you so!

Machine Learning and Model Selection: Conclusion

- In this lecture, we focus on model selection conditional on Unconfoundness assumption
- Thus, we discuss more on model prediction but not causal structure
- This is a totally data driven method with no prior knowledge in economics
- Next lecture, we will turn to variable (model) selection based on our proposed causal structure
- We will introduce a new tool to deal with this issue: Causal Graph

Machine Learning and Model Selection: Conclusion

- In this lecture, we focus on model selection conditional on Unconfoundness assumption
- Thus, we discuss more on model prediction but not causal structure
- This is a totally data driven method with no prior knowledge in economics
- Next lecture, we will turn to variable (model) selection based on our proposed causal structure
- We will introduce a new tool to deal with this issue: Causal Graph

Machine Learning and Model Selection: Conclusion

- In this lecture, we focus on model selection conditional on Unconfoundness assumption
- Thus, we discuss more on model prediction but not causal structure
- This is a totally data driven method with no prior knowledge in economics
- Next lecture, we will turn to variable (model) selection based on our proposed causal structure
- We will introduce a new tool to deal with this issue: Causal Graph

Machine Learning and Model Selection: Conclusion

- In this lecture, we focus on model selection conditional on Unconfoundness assumption
- Thus, we discuss more on model prediction but not causal structure
- This is a totally data driven method with no prior knowledge in economics
- Next lecture, we will turn to variable (model) selection based on our proposed causal structure
- We will introduce a new tool to deal with this issue: Causal Graph

Machine Learning and Model Selection: Conclusion

- In this lecture, we focus on model selection conditional on Unconfoundness assumption
- Thus, we discuss more on model prediction but not causal structure
- This is a totally data driven method with no prior knowledge in economics
- Next lecture, we will turn to variable (model) selection based on our proposed causal structure
- We will introduce a new tool to deal with this issue: Causal Graph

Machine Learning and Model Selection: Conclusion

- In this lecture, we focus on model selection conditional on Unconfoundness assumption
- Thus, we discuss more on model prediction but not causal structure
- This is a totally data driven method with no prior knowledge in economics
- Next lecture, we will turn to variable (model) selection based on our proposed causal structure
- We will introduce a new tool to deal with this issue: Causal Graph

References

Levy, Ro'ee. 2021. "Social Media, News Consumption, and Polarization: Evidence from a Field Experiment." *American Economic Review* 111 (3):831–870.