Text as Data

Justin Grimmer

Associate Professor
Department of Political Science
University of Chicago

August 22nd, 2017

Discovery and Measurement

What is the research process? (Grimmer, Roberts, and Stewart 2017)

1) Discovery: a hypothesis or view of the world
2) Measurement according to some organization
3) Causal Inference: effect of some intervention

Text as data methods assist at each stage of research process

Measurement

Two approaches to measurement

1) Use an existing classification scheme to categorize documents (Today)
2) Simultaneously discover categories and measure prevalence (repurpose discovery methods) (Wednesday)

Types of Classification Problems

Topic: What is this text about?

Types of Classification Problems

Topic: What is this text about?

- Policy area of legislation
  $\Rightarrow$ {Agriculture, Crime, Environment, ...}
- Campaign agendas
  $\Rightarrow$ {Abortion, Campaign, Finance, Taxing, ... }

Types of Classification Problems

Topic: What is this text about?

- Policy area of legislation
  $\Rightarrow$ {Agriculture, Crime, Environment, ...}
- Campaign agendas
  $\Rightarrow$ {Abortion, Campaign, Finance, Taxing, ... }

Sentiment: What is said in this text? [Public Opinion]

Types of Classification Problems

Topic: What is this text about?

- Policy area of legislation
  $\Rightarrow$ {Agriculture, Crime, Environment, ...}
- Campaign agendas
  $\Rightarrow$ {Abortion, Campaign, Finance, Taxing, ... }

Sentiment: What is said in this text? [Public Opinion]

- Positions on legislation
  $\Rightarrow$ { Support, Ambiguous, Oppose }
- Positions on Court Cases
  $\Rightarrow$ { Agree with Court, Disagree with Court }
- Liberal/Conservative Blog Posts
  $\Rightarrow$ { Liberal, Middle, Conservative, No Ideology Expressed }

Types of Classification Problems

Topic: What is this text about?

- Policy area of legislation
  $\Rightarrow$ {Agriculture, Crime, Environment, ...}
- Campaign agendas
  $\Rightarrow$ {Abortion, Campaign, Finance, Taxing, ... }

Sentiment: What is said in this text? [Public Opinion]

- Positions on legislation
  $\Rightarrow$ { Support, Ambiguous, Oppose }
- Positions on Court Cases
  $\Rightarrow$ { Agree with Court, Disagree with Court }
- Liberal/Conservative Blog Posts
  $\Rightarrow$ { Liberal, Middle, Conservative, No Ideology Expressed }

Style/Tone: How is it said?

Types of Classification Problems

Topic: What is this text about?

- Policy area of legislation
  $\Rightarrow$ {Agriculture, Crime, Environment, ...}
- Campaign agendas
  $\Rightarrow$ {Abortion, Campaign, Finance, Taxing, ... }

Sentiment: What is said in this text? [Public Opinion]

- Positions on legislation
  $\Rightarrow$ { Support, Ambiguous, Oppose }
- Positions on Court Cases
  $\Rightarrow$ { Agree with Court, Disagree with Court }
- Liberal/Conservative Blog Posts
  $\Rightarrow$ { Liberal, Middle, Conservative, No Ideology Expressed }

Style/Tone: How is it said?

- Taunting in floor statements
  $\Rightarrow$ { Partisan Taunt, Intra party taunt, Agency taunt, ... }
- Negative campaigning
  $\Rightarrow$ { Negative ad, Positive ad}

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{not, credit claiming}\}$

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{not, credit claiming}\}$

We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{not, credit claiming}\}$

We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) \quad = \quad \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2$$

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{not, credit claiming}\}$

We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \\
\widehat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \right\}
\end{aligned}
$$

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{not, credit claiming}\}$

We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \\
\widehat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \right\} \\
&= \left(\boldsymbol{X}' \boldsymbol{X} \right)^{-1} \boldsymbol{X}' \boldsymbol{Y}
\end{aligned}
$$

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{not, credit claiming}\}$

We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \\
\widehat{\boldsymbol{\beta}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \right\} \\
&= \left(\boldsymbol{X}' \boldsymbol{X} \right)^{-1} \boldsymbol{X}' \boldsymbol{Y}
\end{aligned}
$$

Problem:

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{not, credit claiming}\}$

We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \\
\widehat{\boldsymbol{\beta}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \right\} \\
&= \left(\boldsymbol{X}' \boldsymbol{X} \right)^{-1} \boldsymbol{X}' \boldsymbol{Y}
\end{aligned}
$$

Problem:

- J will likely be large (perhaps $J > N$)

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{not, credit claiming}\}$

We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \\
\widehat{\boldsymbol{\beta}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \right\} \\
&= \left(\boldsymbol{X}' \boldsymbol{X} \right)^{-1} \boldsymbol{X}' \boldsymbol{Y}
\end{aligned}
$$

Problem:

- J will likely be large (perhaps $J > N$)
- There many correlated variables

Regression models

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{not, credit claiming}\}$

We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \\
\widehat{\boldsymbol{\beta}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \boldsymbol{\beta}' \boldsymbol{x}_i \right)^2 \right\} \\
&= \left(\boldsymbol{X}' \boldsymbol{X} \right)^{-1} \boldsymbol{X}' \boldsymbol{Y}
\end{aligned}
$$

Problem:

- J will likely be large (perhaps $J > N$)
- There many correlated variables

Predictions will be variable

Mean Square Error

Suppose θ is some value of the true parameter

Mean Square Error

Suppose θ is some value of the true parameter

Bias:

Mean Square Error

Suppose θ is some value of the true parameter
Bias:

$$\text{Bias} \quad = \quad E[\widehat{\theta} - \theta]$$

Mean Square Error

Suppose θ is some value of the true parameter
Bias:

$$\text{Bias} \quad = \quad E[\widehat{\theta} - \theta]$$

We may care about average distance from truth

Mean Square Error

Suppose θ is some value of the true parameter
Bias:

$$\text{Bias} \quad = \quad E[\widehat{\theta} - \theta]$$

We may care about average distance from truth

$$\mathrm{E}[(\hat{\theta} - \theta)^2]$$

Mean Square Error

Suppose θ is some value of the true parameter
Bias:

$$\text{Bias} \quad = \quad E[\widehat{\theta} - \theta]$$

We may care about average distance from truth

$$\mathrm{E}[(\hat{\theta} - \theta)^2] \quad = \quad E[\hat{\theta}^2] - 2\theta E[\hat{\theta}] + \theta^2$$

Mean Square Error

Suppose θ is some value of the true parameter
Bias:

$$\text{Bias} = E[\widehat{\theta} - \theta]$$

We may care about average distance from truth

$$\begin{aligned} \mathrm{E}[(\hat{\theta} - \theta)^2] &= E[\hat{\theta}^2] - 2\theta E[\hat{\theta}] + \theta^2 \\ &= E[\hat{\theta}^2] - E[\hat{\theta}]^2 + E[\hat{\theta}]^2 - 2\theta E[\hat{\theta}] + \theta^2 \end{aligned}$$

Mean Square Error

Suppose θ is some value of the true parameter
Bias:

$$\text{Bias} = E[\widehat{\theta} - \theta]$$

We may care about average distance from truth

$$\begin{aligned}
\mathrm{E}[(\hat{\theta} - \theta)^2] &= E[\hat{\theta}^2] - 2\theta E[\hat{\theta}] + \theta^2 \\
&= E[\hat{\theta}^2] - E[\hat{\theta}]^2 + E[\hat{\theta}]^2 - 2\theta E[\hat{\theta}] + \theta^2 \\
&= E[\hat{\theta}^2] - E[\hat{\theta}]^2 + (E[\widehat{\theta} - \theta])^2
\end{aligned}$$

Mean Square Error

Suppose θ is some value of the true parameter
Bias:

$$\text{Bias} = E[\widehat{\theta} - \theta]$$

We may care about average distance from truth

$$\begin{aligned}
\mathrm{E}[(\hat{\theta} - \theta)^2] &= E[\hat{\theta}^2] - 2\theta E[\hat{\theta}] + \theta^2 \\
&= E[\hat{\theta}^2] - E[\hat{\theta}]^2 + E[\hat{\theta}]^2 - 2\theta E[\hat{\theta}] + \theta^2 \\
&= E[\hat{\theta}^2] - E[\hat{\theta}]^2 + (E[\widehat{\theta} - \theta])^2 \\
&= \mathrm{Var}(\theta) + \mathrm{Bias}^2
\end{aligned}$$

Mean Square Error

Suppose θ is some value of the true parameter
Bias:

$$\text{Bias} = E[\widehat{\theta} - \theta]$$

We may care about average distance from truth

$$\begin{aligned}
\mathrm{E}[(\hat{\theta} - \theta)^2] &= E[\hat{\theta}^2] - 2\theta E[\hat{\theta}] + \theta^2 \\
&= E[\hat{\theta}^2] - E[\hat{\theta}]^2 + E[\hat{\theta}]^2 - 2\theta E[\hat{\theta}] + \theta^2 \\
&= E[\hat{\theta}^2] - E[\hat{\theta}]^2 + (E[\widehat{\theta} - \theta])^2 \\
&= \mathrm{Var}(\theta) + \mathrm{Bias}^2
\end{aligned}$$

To reduce MSE, we are willing to induce bias to decrease variance $\rightsquigarrow$ methods that shrink coefficeints toward zero

Ridge Regression

Penalty for model complexity

Ridge Regression

Penalty for model complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})$$

Ridge Regression

Penalty for model complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) \;=\; \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2$$

Ridge Regression

Penalty for model complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \underbrace{\lambda \sum_{j=1}^{J} \beta_j^2}_{\text{Penalty}}$$

Ridge Regression

Penalty for model complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \underbrace{\lambda \sum_{j=1}^{J} \beta_j^2}_{\text{Penalty}}$$

where:

Ridge Regression

Penalty for model complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \underbrace{\lambda \sum_{j=1}^{J} \beta_j^2}_{\text{Penalty}}$$

where:

- $\beta_0 \rightsquigarrow$ intercept

Ridge Regression

Penalty for model complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \underbrace{\lambda \sum_{j=1}^{J} \beta_j^2}_{\text{Penalty}}$$

where:

- $\beta_0 \rightsquigarrow$ intercept
- $\lambda \rightsquigarrow$ penalty parameter

Ridge Regression

Penalty for model complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \underbrace{\lambda \sum_{j=1}^{J} \beta_j^2}_{\text{Penalty}}$$

where:

- $\beta_0 \rightsquigarrow$ intercept
- $\lambda \rightsquigarrow$ penalty parameter
- Standardized $\boldsymbol{X}$ (coefficients on same scale)

Ridge Regression $\rightsquigarrow$ Optimization

$$\boldsymbol{\beta}^{\text{Ridge}} = \text{arg min}_{\boldsymbol{\beta}} \{f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})\}$$

Ridge Regression $\rightsquigarrow$ Optimization

$$\begin{aligned}
\boldsymbol{\beta}^{\text{Ridge}} &= \arg\min_{\boldsymbol{\beta}} \{f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y})\} \\
&= \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \beta_j^2 \right\}
\end{aligned}$$

Ridge Regression $\rightsquigarrow$ Optimization

$$
\begin{aligned}
\boldsymbol{\beta}^{\text{Ridge}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) \right\} \\
&= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \beta_j^2 \right\} \\
&= \text{arg min}_{\boldsymbol{\beta}} \left\{ (\boldsymbol{Y} - \boldsymbol{X}'\boldsymbol{\beta})'(\boldsymbol{Y} - \boldsymbol{X}'\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}'\boldsymbol{\beta} \right\}
\end{aligned}
$$

Demean the data and set $\beta_0 = \bar{y} = \sum_{i=1}^{N} \frac{y_i}{N}$

Ridge Regression $\rightsquigarrow$ Optimization

$$
\begin{aligned}
\boldsymbol{\beta}^{\text{Ridge}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) \right\} \\
&= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \beta_j^2 \right\} \\
&= \text{arg min}_{\boldsymbol{\beta}} \left\{ (\boldsymbol{Y} - \boldsymbol{X}'\boldsymbol{\beta})'(\boldsymbol{Y} - \boldsymbol{X}'\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}'\boldsymbol{\beta} \right\} \\
&= \left(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_J \right)^{-1} \boldsymbol{X}'\boldsymbol{Y}
\end{aligned}
$$

Demean the data and set $\beta_0 = \bar{y} = \sum_{i=1}^{N} \frac{y_i}{N}$

Ridge Regression $\rightsquigarrow$ Optimization

$$
\begin{aligned}
\boldsymbol{\beta}^{\text{Ridge}} &= \text{arg min}_{\boldsymbol{\beta}} \left\{ f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) \right\} \\
&= \text{arg min}_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \beta_j^2 \right\} \\
&= \text{arg min}_{\boldsymbol{\beta}} \left\{ (\boldsymbol{Y} - \boldsymbol{X}'\boldsymbol{\beta})'(\boldsymbol{Y} - \boldsymbol{X}'\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}'\boldsymbol{\beta} \right\} \\
&= \left(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_J \right)^{-1} \boldsymbol{X}'\boldsymbol{Y}
\end{aligned}
$$

Demean the data and set $\beta_0 = \bar{y} = \sum_{i=1}^{N} \frac{y_i}{N}$

Ridge Regression$\rightsquigarrow$ Intuition (1)

Suppose $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$.

Ridge Regression$\rightsquigarrow$ Intuition (1)

Suppose $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$.

$$\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^{'}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y}$$

Ridge Regression$\rightsquigarrow$ Intuition (1)

Suppose $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$.

$$\begin{aligned}\widehat{\beta} &= \left(\boldsymbol{X}^{'}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y} \\ &= \boldsymbol{X}^{'}\boldsymbol{Y}\end{aligned}$$

Ridge Regression$\rightsquigarrow$ Intuition (1)

Suppose $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$.

$$\begin{aligned}
\widehat{\beta} &= \left(\boldsymbol{X}^{'}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y} \\
&= \boldsymbol{X}^{'}\boldsymbol{Y} \\
\beta^{\text{ridge}} &= \left(\boldsymbol{X}^{'}\boldsymbol{X} + \lambda\boldsymbol{I}_J\right)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y}
\end{aligned}$$

Ridge Regression$\rightsquigarrow$ Intuition (1)

Suppose $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$.

$$\begin{aligned}
\widehat{\beta} &= \left(\boldsymbol{X}^{'}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y} \\
&= \boldsymbol{X}^{'}\boldsymbol{Y} \\
\beta^{\text{ridge}} &= \left(\boldsymbol{X}^{'}\boldsymbol{X} + \lambda\boldsymbol{I}_J\right)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y} \\
&= (\boldsymbol{I}_j + \lambda\boldsymbol{I}_j)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y}
\end{aligned}$$

Ridge Regression$\rightsquigarrow$ Intuition (1)

Suppose $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$.

$$\begin{aligned}
\widehat{\beta} &= \left(\boldsymbol{X}^{'}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y} \\
&= \boldsymbol{X}^{'}\boldsymbol{Y} \\
\beta^{\text{ridge}} &= \left(\boldsymbol{X}^{'}\boldsymbol{X} + \lambda\boldsymbol{I}_J\right)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y} \\
&= (\boldsymbol{I}_j + \lambda\boldsymbol{I}_j)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y} \\
&= (\boldsymbol{I}_j + \lambda\boldsymbol{I}_j)^{-1}\widehat{\beta}
\end{aligned}$$

Ridge Regression$\rightsquigarrow$ Intuition (1)

Suppose $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$.

$$
\begin{aligned}
\widehat{\boldsymbol{\beta}} &= \left(\boldsymbol{X}^{'}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y} \\
&= \boldsymbol{X}^{'}\boldsymbol{Y} \\
\boldsymbol{\beta}^{\text{ridge}} &= \left(\boldsymbol{X}^{'}\boldsymbol{X} + \lambda\boldsymbol{I}_J\right)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y} \\
&= (\boldsymbol{I}_j + \lambda\boldsymbol{I}_j)^{-1}\boldsymbol{X}^{'}\boldsymbol{Y} \\
&= (\boldsymbol{I}_j + \lambda\boldsymbol{I}_j)^{-1}\widehat{\boldsymbol{\beta}} \\
\beta_j^{\text{Ridge}} &= \frac{\widehat{\beta}_j}{1+\lambda}
\end{aligned}
$$

Ridge Regression$\rightsquigarrow$ Intuition (2)

$$\begin{aligned} \boldsymbol{\beta}_j &\sim \text{Normal}(0, \tau^2) \\ y_i &\sim \text{Normal}(\beta_0 + \boldsymbol{x}_i^{'}\boldsymbol{\beta}, \sigma^2) \end{aligned}$$

Ridge Regression$\rightsquigarrow$ Intuition (2)

$$\begin{aligned} \boldsymbol{\beta}_j &\sim \text{Normal}(0, \tau^2) \\ y_i &\sim \text{Normal}(\beta_0 + \boldsymbol{x}_i^{'}\boldsymbol{\beta}, \sigma^2) \end{aligned}$$

$$p(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{Y}) \propto \prod_{j=1}^{J} p(\beta_j) \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i, \boldsymbol{\beta})$$

Ridge Regression$\rightsquigarrow$ Intuition (2)

$$\begin{aligned} \boldsymbol{\beta}_j &\sim \text{Normal}(0, \tau^2) \\ y_i &\sim \text{Normal}(\beta_0 + \boldsymbol{x}_i^{'}\boldsymbol{\beta}, \sigma^2) \end{aligned}$$

$$\begin{aligned} p(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{Y}) &\propto \prod_{j=1}^{J} p(\beta_j) \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i, \boldsymbol{\beta}) \\ &\propto \prod_{j=1}^{J} \frac{1}{\sqrt{2\pi}\tau} \exp\left(-\frac{\beta_j^2}{2\tau^2}\right) \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \beta_0 - \boldsymbol{x}_i^{'}\boldsymbol{\beta})^2}{2\sigma^2}\right) \end{aligned}$$

Ridge Regression$\rightsquigarrow$ Intuition (2)

$$\log p(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{Y}) \quad = \quad -\sum_{j=1}^{J} \frac{\beta_j^2}{2\tau^2} - \sum_{i=1}^{N} \frac{(y_i - \beta_0 - \boldsymbol{x}'\boldsymbol{\beta})^2}{2\sigma^2}$$

Ridge Regression$\rightsquigarrow$ Intuition (2)

$$\begin{aligned}
\log p(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{Y}) &= -\sum_{j=1}^{J} \frac{\beta_j^2}{2\tau^2} - \sum_{i=1}^{N} \frac{(y_i - \beta_0 - \boldsymbol{x}'\boldsymbol{\beta})^2}{2\sigma^2} \\
-2\sigma^2 \log p(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} (y_i - \beta_0 - \boldsymbol{x}'\boldsymbol{\beta})^2 + \sum_{j=1}^{J} \frac{\sigma^2}{\tau^2} \beta_j^2
\end{aligned}$$

Ridge Regression$\rightsquigarrow$ Intuition (2)

$$\begin{aligned}
\log p(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{Y}) &= -\sum_{j=1}^{J} \frac{\beta_j^2}{2\tau^2} - \sum_{i=1}^{N} \frac{(y_i - \beta_0 - \boldsymbol{x}'\boldsymbol{\beta})^2}{2\sigma^2} \\
-2\sigma^2 \log p(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} (y_i - \beta_0 - \boldsymbol{x}'\boldsymbol{\beta})^2 + \sum_{j=1}^{J} \frac{\sigma^2}{\tau^2} \beta_j^2
\end{aligned}$$

where:

Ridge Regression$\rightsquigarrow$ Intuition (2)

$$\begin{aligned}
\log p(\boldsymbol{\beta}|\boldsymbol{X},\boldsymbol{Y}) &= -\sum_{j=1}^{J}\frac{\beta_j^2}{2\tau^2} - \sum_{i=1}^{N}\frac{(y_i-\beta_0-\boldsymbol{x}'\boldsymbol{\beta})^2}{2\sigma^2} \\
-2\sigma^2\log p(\boldsymbol{\beta}|\boldsymbol{X},\boldsymbol{Y}) &= \sum_{i=1}^{N}(y_i-\beta_0-\boldsymbol{x}'\boldsymbol{\beta})^2 + \sum_{j=1}^{J}\frac{\sigma^2}{\tau^2}\beta_j^2
\end{aligned}$$

where:

- $\lambda = \frac{\sigma^2}{\tau^2}$

Ridge Regression $\rightsquigarrow$ Intuition (3)

Definition

Suppose $\boldsymbol{X}$ *is an* $N \times J$ *matrix. Then* $\boldsymbol{X}$ *can be written as:*

$$\boldsymbol{X} = \underbrace{\boldsymbol{U}}_{N \times N} \underbrace{\boldsymbol{S}}_{N \times J} \underbrace{\boldsymbol{V}'}_{J \times J}$$

Where:

$$\begin{aligned} \boldsymbol{U}'\boldsymbol{U} &= \boldsymbol{I}_N \\ \boldsymbol{V}'\boldsymbol{V} &= \boldsymbol{V}\boldsymbol{V}' = \boldsymbol{I}_J \end{aligned}$$

$\boldsymbol{S}$ *contains* $\min(N, J)$ *singular values,* $\sqrt{\lambda_j} \geq 0$ *down the diagonal and then 0's for the remaining entries*

Ridge Regression $\rightsquigarrow$ Intuition (3)

Recall: PCA:

$$\frac{1}{N}\boldsymbol{X}'\boldsymbol{X} = \underbrace{\boldsymbol{W}}_{\text{eigenvectors}} \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_J \end{pmatrix} \underbrace{\boldsymbol{W}'}_{\text{eigenvectors}}$$

Ridge Regression $\rightsquigarrow$ Intuition (3)

Recall: PCA:

$$\frac{1}{N}\boldsymbol{X}'\boldsymbol{X} = \underbrace{\boldsymbol{W}}_{\text{eigenvectors}} \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_J \end{pmatrix} \underbrace{\boldsymbol{W}'}_{\text{eigenvectors}}$$

Using SVD:

Ridge Regression $\rightsquigarrow$ Intuition (3)

Recall: PCA:

$$\frac{1}{N}\boldsymbol{X}'\boldsymbol{X} = \underbrace{\boldsymbol{W}}_{\text{eigenvectors}} \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_J \end{pmatrix} \underbrace{\boldsymbol{W}'}_{\text{eigenvectors}}$$

Using SVD:

$$\frac{1}{N}\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}' \underbrace{\left(\boldsymbol{U}'\boldsymbol{U}\right)}_{\boldsymbol{I}_J} \boldsymbol{S}\boldsymbol{V}'$$

Ridge Regression $\rightsquigarrow$ Intuition (3)

Recall: PCA:

$$\frac{1}{N}\boldsymbol{X}'\boldsymbol{X} = \underbrace{\boldsymbol{W}}_{\text{eigenvectors}} \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_J \end{pmatrix} \underbrace{\boldsymbol{W}'}_{\text{eigenvectors}}$$

Using SVD:

$$\begin{aligned} \frac{1}{N}\boldsymbol{X}'\boldsymbol{X} &= \boldsymbol{V}\boldsymbol{S}' \underbrace{\left(\boldsymbol{U}'\boldsymbol{U}\right)}_{\boldsymbol{I}_J} \boldsymbol{S}\boldsymbol{V}' \\ &= \frac{1}{N}\boldsymbol{V}\boldsymbol{S}'\boldsymbol{S}\boldsymbol{V}' \end{aligned}$$

Ridge Regression $\rightsquigarrow$ Intuition (3)

Recall: PCA:

$$\frac{1}{N}\boldsymbol{X}'\boldsymbol{X} = \underbrace{\boldsymbol{W}}_{\text{eigenvectors}} \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_J \end{pmatrix} \underbrace{\boldsymbol{W}'}_{\text{eigenvectors}}$$

Using SVD:

$$\begin{aligned} \frac{1}{N}\boldsymbol{X}'\boldsymbol{X} &= \boldsymbol{V}\boldsymbol{S}' \underbrace{\left(\boldsymbol{U}'\boldsymbol{U}\right)}_{\boldsymbol{I}_J} \boldsymbol{S}\boldsymbol{V}' \\ &= \frac{1}{N}\boldsymbol{V}\boldsymbol{S}'\boldsymbol{S}\boldsymbol{V}' \\ &= \underbrace{\boldsymbol{V}}_{\text{eigenvectors}} \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_J \end{pmatrix} \underbrace{\boldsymbol{V}'}_{\text{eigenvectors}} \end{aligned}$$

Ridge Regression $\rightsquigarrow$ Intuition (3)

We can write the predicted values for a regular regression as

$$\begin{aligned}\hat{Y} &= \boldsymbol{X}\hat{\boldsymbol{\beta}} \\ &= \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{Y} \\ &= \boldsymbol{U}\boldsymbol{U}'\boldsymbol{Y} = \sum_{j=1}^{J}\boldsymbol{u}_j\boldsymbol{u}_j'\boldsymbol{Y}\end{aligned}$$

Ridge Regression $\rightsquigarrow$ Intuition (3)

We can write the predicted values for a regular regression as

$$\begin{aligned}\hat{Y} &= \boldsymbol{X}\hat{\boldsymbol{\beta}} \\ &= \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{Y} \\ &= \boldsymbol{U}\boldsymbol{U}'\boldsymbol{Y} = \sum_{j=1}^{J}\boldsymbol{u}_j\boldsymbol{u}_j'\boldsymbol{Y}\end{aligned}$$

We can write $\boldsymbol{\beta}^{\text{ridge}}$ as

Ridge Regression $\rightsquigarrow$ Intuition (3)

We can write the predicted values for a regular regression as

$$\begin{aligned}\hat{Y} &= \boldsymbol{X}\hat{\boldsymbol{\beta}} \\ &= \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{Y} \\ &= \boldsymbol{U}\boldsymbol{U}'\boldsymbol{Y} = \sum_{j=1}^{J}\boldsymbol{u}_j\boldsymbol{u}_j'\boldsymbol{Y}\end{aligned}$$

We can write $\boldsymbol{\beta}^{\text{ridge}}$ as

$$\hat{Y}^{\text{ridge}} = \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{I}_J\right)^{-1}\boldsymbol{X}'\boldsymbol{Y}$$

Ridge Regression $\rightsquigarrow$ Intuition (3)

We can write the predicted values for a regular regression as

$$\begin{aligned}\hat{Y} &= \boldsymbol{X}\hat{\boldsymbol{\beta}} \\ &= \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{Y} \\ &= \boldsymbol{U}\boldsymbol{U}'\boldsymbol{Y} = \sum_{j=1}^{J}\boldsymbol{u}_j\boldsymbol{u}_j'\boldsymbol{Y}\end{aligned}$$

We can write $\boldsymbol{\beta}^{\text{ridge}}$ as

$$\begin{aligned}\hat{Y}^{\text{ridge}} &= \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{I}_J\right)^{-1}\boldsymbol{X}'\boldsymbol{Y} \\ &= \boldsymbol{U}\tilde{\boldsymbol{S}}\boldsymbol{U}'\boldsymbol{Y}\end{aligned}$$

Where

$$\tilde{\boldsymbol{S}} = \left[\boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S} + \lambda\boldsymbol{I}_J)^{-1}\boldsymbol{S}\right]$$

Ridge Regression $\rightsquigarrow$ Intuition (3)

We can write the predicted values for a regular regression as

$$\begin{aligned}\hat{Y} &= \boldsymbol{X}\hat{\boldsymbol{\beta}} \\ &= \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{Y} \\ &= \boldsymbol{U}\boldsymbol{U}'\boldsymbol{Y} = \sum_{j=1}^{J}\boldsymbol{u}_j\boldsymbol{u}_j'\boldsymbol{Y}\end{aligned}$$

We can write $\boldsymbol{\beta}^{\text{ridge}}$ as

$$\begin{aligned}\hat{Y}^{\text{ridge}} &= \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{I}_J\right)^{-1}\boldsymbol{X}'\boldsymbol{Y} \\ &= \boldsymbol{U}\tilde{\boldsymbol{S}}\boldsymbol{U}'\boldsymbol{Y}\end{aligned}$$

Where

$$\tilde{\boldsymbol{S}} = \left[\boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S} + \lambda\boldsymbol{I}_J)^{-1}\boldsymbol{S}\right]$$

Which we can write as:

$$\hat{Y}^{\text{ridge}} = \sum_{j=1}^{J}\boldsymbol{u}_j\frac{\lambda_j}{\lambda_j + \lambda}\boldsymbol{u}_j'\boldsymbol{Y}$$

Degrees of Freedom for Ridge

We will say that the degrees of freedom for Ridge regression with penalty λ is

$$\text{dof}(\lambda) = \sum_{j=1}^{J} \frac{\lambda_j}{\lambda_j + \lambda}$$

Lasso Regression Objective Function

Different Penalty for Model Complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \underbrace{|\beta_j|}_{\text{Penalty}}$$

Lasso Regression Objective Function

Different Penalty for Model Complexity

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \underbrace{|\beta_j|}_{\text{Penalty}}$$

Lasso Regression Optimization

Definition

Coordinate Descent Algorithms:

Consider $g : \Re^J \rightarrow \Re$. *Our goal is to find* $\boldsymbol{x}^* \in \Re^J$ *such that* $g(\boldsymbol{x}^*) \leq g(\boldsymbol{x})$ *for all* $\boldsymbol{x} \in \Re$.

To find $\boldsymbol{x}^*$*:*

Until convergence: for each iteration t *and each coordinate* j

$$x_j^{t+1} = arg\ min_{x_j \in \Re} g(x_1^{t+1}, x_2^{t+1}, \ldots, x_{j-1}^{t+1}, x_j, x_{j+1}^t, \ldots, x_J^t)$$

Lasso Regression Optimization: Coordinate Descent

$$\tilde{f}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} |\beta_j|$$

Lasso Regression Optimization: Coordinate Descent

$$\tilde{f}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} |\beta_j|$$

- Case 1: If $\beta_j = 0 \rightsquigarrow$ not differentiable. But $\beta_j = 0$

Lasso Regression Optimization: Coordinate Descent

$$\tilde{f}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} |\beta_j|$$

- Case 1: If $\beta_j = 0 \rightsquigarrow$ not differentiable. But $\beta_j = 0$
- Case 2: If $\beta_j > (<) 0 \rightsquigarrow$ differentiable $\rightsquigarrow$ differentiate and solve for β_j

Lasso Regression Optimization: Coordinate Descent

$$\tilde{f}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} |\beta_j|$$

- Case 1: If $\beta_j = 0 \rightsquigarrow$ not differentiable. But $\beta_j = 0$
- Case 2: If $\beta_j > (<) 0 \rightsquigarrow$ differentiable $\rightsquigarrow$ differentiate and solve for β_j

Define $\tilde{y}_i^j = \beta_0 + \sum_{l \neq j} x_{il} \beta_l$

Lasso Regression Optimization: Coordinate Descent

$$\tilde{f}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} |\beta_j|$$

- Case 1: If $\beta_j = 0 \rightsquigarrow$ not differentiable. But $\beta_j = 0$
- Case 2: If $\beta_j > (<) 0 \rightsquigarrow$ differentiable $\rightsquigarrow$ differentiate and solve for β_j

Define $\tilde{y}_i^j = \beta_0 + \sum_{l \neq j} x_{il} \beta_l$
$r^j \equiv \frac{1}{N} \sum_{i=1}^{N} x_{ij} (y_i - \tilde{y}_i^j)$

Lasso Regression Optimization: Coordinate Descent

$$\tilde{f}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} |\beta_j|$$

- Case 1: If $\beta_j = 0 \rightsquigarrow$ not differentiable. But $\beta_j = 0$
- Case 2: If $\beta_j > (<) 0 \rightsquigarrow$ differentiable $\rightsquigarrow$ differentiate and solve for β_j

Define $\tilde{y}_i^j = \beta_0 + \sum_{l \neq j} x_{il} \beta_l$
$r^j \equiv \frac{1}{N} \sum_{i=1}^{N} x_{ij} (y_i - \tilde{y}_i^j)$
Update step for β_j is

Lasso Regression Optimization: Coordinate Descent

$$\tilde{f}(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) \;=\; \frac{1}{2N}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{J}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{J}|\beta_j|$$

- Case 1: If $\beta_j = 0 \rightsquigarrow$ not differentiable. But $\beta_j = 0$
- Case 2: If $\beta_j > (<) 0 \rightsquigarrow$ differentiable $\rightsquigarrow$ differentiate and solve for β_j

Define $\tilde{y}_i^j = \beta_0 + \sum_{l \neq j} x_{il}\beta_l$
$r^j \equiv \frac{1}{N}\sum_{i=1}^{N} x_{ij}(y_i - \tilde{y}_i^j)$
Update step for β_j is

$$\beta_j \;\leftarrow\; \text{sign}(r^j)\max(|r^j| - \lambda, 0)$$

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Suppose again $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Suppose again $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) \;\; = \;\; (Y - \boldsymbol{X}\boldsymbol{\beta})^{'}(Y - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{J} |\beta_j|$$

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Suppose again $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= (Y - \boldsymbol{X}\boldsymbol{\beta})^{'}(Y - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{J} |\beta_j| \\
&= -2\boldsymbol{X}^{'}\boldsymbol{Y}\boldsymbol{\beta} + \boldsymbol{\beta}^{'}\boldsymbol{\beta} + \lambda \sum_{j=1}^{J} |\beta_j|
\end{aligned}
$$

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Suppose again $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= (Y - \boldsymbol{X}\boldsymbol{\beta})^{'}(Y - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{J} |\beta_j| \\
&= -2\boldsymbol{X}^{'}\boldsymbol{Y}\boldsymbol{\beta} + \boldsymbol{\beta}^{'}\boldsymbol{\beta} + \lambda \sum_{j=1}^{J} |\beta_j|
\end{aligned}
$$

The coefficient is

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Suppose again $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= (Y - \boldsymbol{X}\boldsymbol{\beta})^{'}(Y - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{J} |\beta_j| \\
&= -2\boldsymbol{X}^{'}\boldsymbol{Y}\boldsymbol{\beta} + \boldsymbol{\beta}^{'}\boldsymbol{\beta} + \lambda \sum_{j=1}^{J} |\beta_j|
\end{aligned}
$$

The coefficient is

$$
\beta_j^{\text{LASSO}} = \text{sign}\left(\widehat{\beta}_j\right)\left(|\widehat{\beta}_j| - \lambda\right)_+
$$

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Suppose again $\boldsymbol{X}^{'}\boldsymbol{X} = \boldsymbol{I}_J$

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= (Y - \boldsymbol{X}\boldsymbol{\beta})^{'}(Y - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{J} |\beta_j| \\
&= -2\boldsymbol{X}^{'}\boldsymbol{Y}\boldsymbol{\beta} + \boldsymbol{\beta}^{'}\boldsymbol{\beta} + \lambda \sum_{j=1}^{J} |\beta_j|
\end{aligned}
$$

The coefficient is

$$
\beta_j^{\text{LASSO}} = \text{sign}\left(\widehat{\beta}_j\right)\left(|\widehat{\beta}_j| - \lambda\right)_+
$$

- $\text{sign}(\cdot) \rightsquigarrow 1$ or -1

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Suppose again $\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{I}_J$

$$\begin{aligned} f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) &= (Y - \boldsymbol{X}\boldsymbol{\beta})'(Y - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{J} |\beta_j| \\ &= -2\boldsymbol{X}'\boldsymbol{Y}\boldsymbol{\beta} + \boldsymbol{\beta}'\boldsymbol{\beta} + \lambda \sum_{j=1}^{J} |\beta_j| \end{aligned}$$

The coefficient is

$$\beta_j^{\text{LASSO}} = \text{sign}\left(\widehat{\beta}_j\right)\left(|\widehat{\beta}_j| - \lambda\right)_+$$

- $\text{sign}(\cdot) \rightsquigarrow 1$ or -1
- $\left(|\widehat{\beta}_j| - \lambda\right)_+ = \max(|\widehat{\beta}_j| - \lambda, 0)$

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Compare soft assignment

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Compare soft assignment

$$\beta_j^{\text{LASSO}} = \text{sign}\left(\widehat{\beta}_j\right)\left(|\widehat{\beta}_j| - \lambda\right)_+$$

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Compare soft assignment

$$\beta_j^{\text{LASSO}} = \text{sign}\left(\widehat{\beta}_j\right)\left(|\widehat{\beta}_j| - \lambda\right)_+$$

With hard assignment, selecting M biggest components

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Compare soft assignment

$$\beta_j^{\text{LASSO}} = \text{sign}\left(\widehat{\beta}_j\right)\left(|\widehat{\beta}_j| - \lambda\right)_+$$

With hard assignment, selecting M biggest components

$$\beta_j^{\text{subset}} = \widehat{\beta}_j \cdot I\left(|\widehat{\beta}_j| \geq |\widehat{\beta}_{(M)}|\right)$$

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Compare soft assignment

$$\beta_j^{\text{LASSO}} = \text{sign}\left(\widehat{\beta}_j\right)\left(|\widehat{\beta}_j| - \lambda\right)_+$$

With hard assignment, selecting M biggest components

$$\beta_j^{\text{subset}} = \widehat{\beta}_j \cdot I\left(|\widehat{\beta}_j| \geq |\widehat{\beta}_{(M)}|\right)$$

Intuition 2: Prior on coefficients $\rightsquigarrow$ Laplace "The Bayesian LASSO"

Lasso Regression$\rightsquigarrow$ Intuition 1, Soft Thresholding

Compare soft assignment

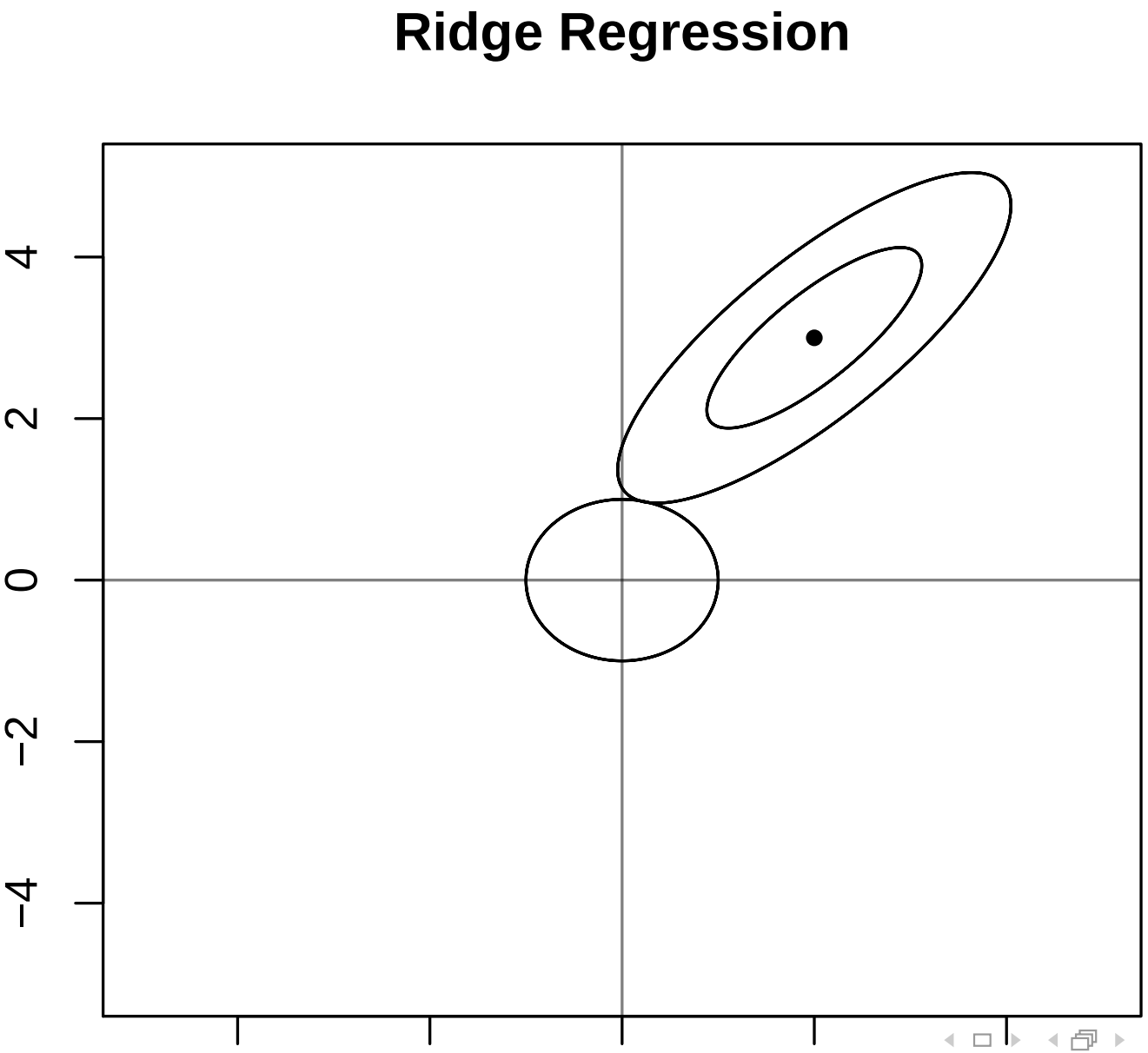
$$\beta_j^{\text{LASSO}} = \text{sign}\left(\widehat{\beta}_j\right)\left(|\widehat{\beta}_j| - \lambda\right)_+$$

With hard assignment, selecting M biggest components

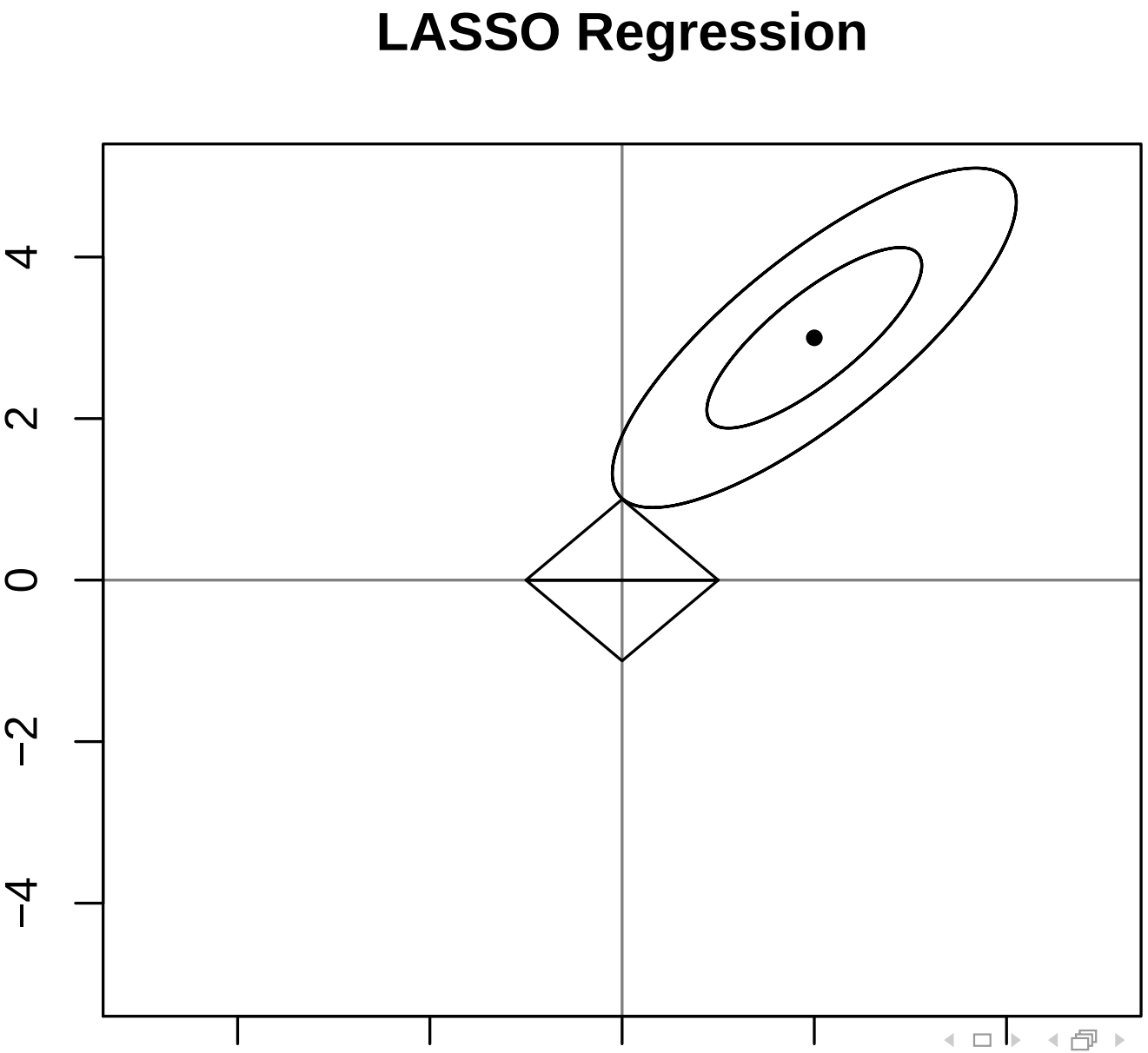
$$\beta_j^{\text{subset}} = \widehat{\beta}_j \cdot I\left(|\widehat{\beta}_j| \geq |\widehat{\beta}_{(M)}|\right)$$

Intuition 2: Prior on coefficients $\rightsquigarrow$ Laplace "The Bayesian LASSO"
Why does LASSO induce sparsity?

Comparing Ridge and LASSO



Comparing Ridge and LASSO



Comparing Ridge and LASSO

Contrast $\beta = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ and $\tilde{\beta} = (1, 0)$

Comparing Ridge and LASSO

Contrast $\beta = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ and $\tilde{\beta} = (1, 0)$

Under ridge:

Comparing Ridge and LASSO

Contrast $\beta = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ and $\tilde{\beta} = (1, 0)$

Under ridge:

$$\sum_{j=1}^{2} \beta_j^2 = \frac{1}{2} + \frac{1}{2} = 1$$

Comparing Ridge and LASSO

Contrast $\beta = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ and $\tilde{\beta} = (1, 0)$

Under ridge:

$$\begin{aligned} \sum_{j=1}^{2} \beta_j^2 &= \frac{1}{2} + \frac{1}{2} = 1 \\ \sum_{j=1}^{2} \tilde{\beta}_j^2 &= 1 + 0 = 1 \end{aligned}$$

Comparing Ridge and LASSO

Contrast $\beta = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ and $\tilde{\beta} = (1, 0)$

Under ridge:

$$\begin{aligned} \sum_{j=1}^{2} \beta_j^2 &= \frac{1}{2} + \frac{1}{2} = 1 \\ \sum_{j=1}^{2} \tilde{\beta}_j^2 &= 1 + 0 = 1 \end{aligned}$$

Under LASSO

Comparing Ridge and LASSO

Contrast $\beta = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ and $\tilde{\beta} = (1, 0)$

Under ridge:

$$\begin{aligned} \sum_{j=1}^{2} \beta_j^2 &= \frac{1}{2} + \frac{1}{2} = 1 \\ \sum_{j=1}^{2} \tilde{\beta}_j^2 &= 1 + 0 = 1 \end{aligned}$$

Under LASSO

$$\sum_{j=1}^{2} |\beta_j| = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} = \sqrt{2}$$

Comparing Ridge and LASSO

Contrast $\beta = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ and $\tilde{\beta} = (1, 0)$

Under ridge:

$$\begin{aligned} \sum_{j=1}^{2} \beta_j^2 &= \frac{1}{2} + \frac{1}{2} = 1 \\ \sum_{j=1}^{2} \tilde{\beta}_j^2 &= 1 + 0 = 1 \end{aligned}$$

Under LASSO

$$\begin{aligned} \sum_{j=1}^{2} |\beta_j| &= \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} = \sqrt{2} \\ \sum_{j=1}^{2} |\tilde{\beta}_j| &= 1 + 0 = 1 \end{aligned}$$

Ridge and LASSO: The Elastic-Net

Combining the two criteria $\rightsquigarrow$ Elastic-Net

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \left(\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right)$$

Ridge and LASSO: The Elastic-Net

Combining the two criteria $\rightsquigarrow$ Elastic-Net

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{J} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{J} \left(\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right)$$

The new update step (for coordinate descent:)

Ridge and LASSO: The Elastic-Net

Combining the two criteria $\rightsquigarrow$ Elastic-Net

$$f(\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2N}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{J}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{J}\left(\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|\right)$$

The new update step (for coordinate descent:)

$$\beta_j \leftarrow \frac{\text{sign}(r^j)\max(|r^j| - \lambda\alpha, 0)}{1 + \lambda(1-\alpha)}$$

Selecting λ

How do we determine λ? $\rightsquigarrow$ Cross validation

Selecting λ

How do we determine λ? $\rightsquigarrow$ Cross validation
Applying models gives score (probability) of document belong to class$\rightsquigarrow$ threshold to classify

Selecting λ

How do we determine λ? $\rightsquigarrow$ Cross validation
Applying models gives score (probability) of document belong to class$\rightsquigarrow$ threshold to classify

Loss Functions and Model Complexity

Suppose observations i have dependent variables Y_i and covariates $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iP})$.

Loss Functions and Model Complexity

Suppose observations i have dependent variables Y_i and covariates $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iP})$.
Assume:

$$\begin{aligned} Y_i &\sim \text{Distribution}(\mu_i, \phi) \\ \mu_i &= f(\boldsymbol{\beta}, \boldsymbol{x}_i) \end{aligned}$$

Use MLE to obtain $\hat{\boldsymbol{\beta}}$.

Loss Functions and Model Complexity

Suppose observations i have dependent variables Y_i and covariates $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iP})$.
Assume:

$$\begin{aligned} Y_i &\sim \text{Distribution}(\mu_i, \phi) \\ \mu_i &= f(\boldsymbol{\beta}, \boldsymbol{x}_i) \end{aligned}$$

Use MLE to obtain $\hat{\boldsymbol{\beta}}$.
Potential loss functions:

Loss Functions and Model Complexity

Suppose observations i have dependent variables Y_i and covariates $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iP})$.
Assume:

$$\begin{aligned} Y_i &\sim \text{Distribution}(\mu_i, \phi) \\ \mu_i &= f(\boldsymbol{\beta}, \boldsymbol{x}_i) \end{aligned}$$

Use MLE to obtain $\hat{\boldsymbol{\beta}}$.
Potential loss functions:

$$L\left(Y_i, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right)$$

Loss Functions and Model Complexity

Suppose observations i have dependent variables Y_i and covariates $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iP})$.
Assume:

$$\begin{aligned} Y_i &\sim \text{Distribution}(\mu_i, \phi) \\ \mu_i &= f(\boldsymbol{\beta}, \boldsymbol{x}_i) \end{aligned}$$

Use MLE to obtain $\hat{\boldsymbol{\beta}}$.
Potential loss functions:

$$L\left(Y_i, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right) = \left(Y_i - f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right)^2$$

Loss Functions and Model Complexity

Suppose observations i have dependent variables Y_i and covariates $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iP})$.
Assume:

$$\begin{aligned} Y_i &\sim \text{Distribution}(\mu_i, \phi) \\ \mu_i &= f(\boldsymbol{\beta}, \boldsymbol{x}_i) \end{aligned}$$

Use MLE to obtain $\hat{\boldsymbol{\beta}}$.
Potential loss functions:

$$\begin{aligned} L\left(Y_i, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right) &= \left(Y_i - f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right)^2 \\ &= \left|Y_i - f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right| \end{aligned}$$

Loss Functions and Model Complexity

Suppose observations i have dependent variables Y_i and covariates $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iP})$.
Assume:

$$\begin{aligned} Y_i &\sim \text{Distribution}(\mu_i, \phi) \\ \mu_i &= f(\boldsymbol{\beta}, \boldsymbol{x}_i) \end{aligned}$$

Use MLE to obtain $\hat{\boldsymbol{\beta}}$.
Potential loss functions:

$$\begin{aligned} L\left(Y_i, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right) &= \left(Y_i - f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right)^2 \\ &= \left|Y_i - f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right| \\ &= I\left(Y_i = I(f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i) > \tau)\right) \end{aligned}$$

Training and Test Sets

The useful "fiction" of training and test sets:

Training and Test Sets

The useful "fiction" of training and test sets:

- Training set: data set used to fit the model

Training and Test Sets

The useful "fiction" of training and test sets:

- Training set: data set used to fit the model
- Test set: data used to evaluate fit of the model

Training and Test Sets

The useful "fiction" of training and test sets:

- Training set: data set used to fit the model
- Test set: data used to evaluate fit of the model

Even if no division, useful to think about systematic components of data.

Loss Functions and Model Complexity

Suppose that we have:

Loss Functions and Model Complexity

Suppose that we have:

- Training sets, $\mathcal{T}$, with $|\mathcal{T}| = N_{\text{train}}$

Loss Functions and Model Complexity

Suppose that we have:

- Training sets, $\mathcal{T}$, with $|\mathcal{T}| = N_{\text{train}}$
- Test sets, $\mathcal{O}$ with $|\mathcal{O}| = N_{\text{test}}$

Loss Functions and Model Complexity

Suppose that we have:

- Training sets, $\mathcal{T}$, with $|\mathcal{T}| = N_{\text{train}}$
- Test sets, $\mathcal{O}$ with $|\mathcal{O}| = N_{\text{test}}$

Training (in-sample) error is:

$$=$$

Loss Functions and Model Complexity

Suppose that we have:

- Training sets, $\mathcal{T}$, with $|\mathcal{T}| = N_{\text{train}}$
- Test sets, $\mathcal{O}$ with $|\mathcal{O}| = N_{\text{test}}$

Training (in-sample) error is:

$$\text{Error}_{\text{in}} =$$

Loss Functions and Model Complexity

Suppose that we have:

- Training sets, $\mathcal{T}$, with $|\mathcal{T}| = N_{\text{train}}$
- Test sets, $\mathcal{O}$ with $|\mathcal{O}| = N_{\text{test}}$

Training (in-sample) error is:

$$\text{Error}_{\text{in}} = \sum_{i \in \mathcal{T}} \frac{1}{N_{\text{train}}} L(Y_i, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i))$$

Loss Functions and Model Complexity

Suppose that we have:

- Training sets, $\mathcal{T}$, with $|\mathcal{T}| = N_{\text{train}}$
- Test sets, $\mathcal{O}$ with $|\mathcal{O}| = N_{\text{test}}$

Training (in-sample) error is:

$$\text{Error}_{\text{in}} = \sum_{i \in \mathcal{T}} \frac{1}{N_{\text{train}}} L(Y_i, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i))$$

We'd like to estimate out of sample performance with

Loss Functions and Model Complexity

Suppose that we have:

- Training sets, $\mathcal{T}$, with $|\mathcal{T}| = N_{\text{train}}$
- Test sets, $\mathcal{O}$ with $|\mathcal{O}| = N_{\text{test}}$

Training (in-sample) error is:

$$\text{Error}_{\text{in}} = \sum_{i \in \mathcal{T}} \frac{1}{N_{\text{train}}} L(Y_i, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i))$$

We'd like to estimate out of sample performance with

$$\text{Error}_{\text{out}} = \text{E}[L(\boldsymbol{Y}_{i \in \mathcal{O}}, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_{i \in \mathcal{O}}))|\mathcal{T}]$$

Loss Functions and Model Complexity

Suppose that we have:

- Training sets, $\mathcal{T}$, with $|\mathcal{T}| = N_{\text{train}}$
- Test sets, $\mathcal{O}$ with $|\mathcal{O}| = N_{\text{test}}$

Training (in-sample) error is:

$$\text{Error}_{\text{in}} = \sum_{i \in \mathcal{T}} \frac{1}{N_{\text{train}}} L(Y_i, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i))$$

We'd like to estimate out of sample performance with

$$\text{Error}_{\text{out}} = \text{E}[L(\boldsymbol{Y}_{i \in \mathcal{O}}, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_{i \in \mathcal{O}}))|\mathcal{T}]$$

where the expectation is taken over samples for test sets and supposes we have a training set.

Loss Functions and Model Complexity

Suppose that we have:

- Training sets, $\mathcal{T}$, with $|\mathcal{T}| = N_{\text{train}}$
- Test sets, $\mathcal{O}$ with $|\mathcal{O}| = N_{\text{test}}$

Training (in-sample) error is:

$$\text{Error}_{\text{in}} = \sum_{i \in \mathcal{T}} \frac{1}{N_{\text{train}}} L(Y_i, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i))$$

We'd like to estimate out of sample performance with

$$\text{Error}_{\text{out}} = \text{E}[L(\boldsymbol{Y}_{i \in \mathcal{O}}, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_{i \in \mathcal{O}}))|\mathcal{T}]$$

where the expectation is taken over samples for test sets and supposes we have a training set.

$$\text{Error} = \text{E}\left[\text{E}[L(\boldsymbol{Y}, f(\hat{\boldsymbol{\beta}}, \boldsymbol{X}))|\mathcal{T}]\right]$$

Loss Functions and Model Complexity

Suppose $Y_i = f(\boldsymbol{x}_i) + \epsilon_i$

Loss Functions and Model Complexity

Suppose $Y_i = f(\boldsymbol{x}_i) + \epsilon_i$
Where $E[\epsilon_i] = 0$

Loss Functions and Model Complexity

Suppose $Y_i = f(\boldsymbol{x}_i) + \epsilon_i$
Where $E[\epsilon_i] = 0$
$\text{var}(\epsilon_i) = \sigma_\epsilon^2$

Loss Functions and Model Complexity

Suppose $Y_i = f(\boldsymbol{x}_i) + \epsilon_i$
Where $E[\epsilon_i] = 0$
$\text{var}(\epsilon_i) = \sigma_\epsilon^2$
Define $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}) = \hat{f}(\boldsymbol{x})$

Loss Functions and Model Complexity

Suppose $Y_i = f(\boldsymbol{x}_i) + \epsilon_i$
Where $E[\epsilon_i] = 0$
$\text{var}(\epsilon_i) = \sigma_\epsilon^2$
Define $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}) = \hat{f}(\boldsymbol{x})$
With squared error loss:

Loss Functions and Model Complexity

Suppose $Y_i = f(\boldsymbol{x}_i) + \epsilon_i$
Where $E[\epsilon_i] = 0$
$\text{var}(\epsilon_i) = \sigma_\epsilon^2$
Define $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}) = \hat{f}(\boldsymbol{x})$
With squared error loss:

$$\text{Error}(\boldsymbol{x}_0) \quad = \quad \text{E}[(Y_i - \hat{f}(\boldsymbol{x}_i))^2 | \boldsymbol{x}_i = \boldsymbol{x}_0]$$

Loss Functions and Model Complexity

Suppose $Y_i = f(\boldsymbol{x}_i) + \epsilon_i$
Where $E[\epsilon_i] = 0$
$\text{var}(\epsilon_i) = \sigma_\epsilon^2$
Define $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}) = \hat{f}(\boldsymbol{x})$
With squared error loss:

$$\begin{aligned} \text{Error}(\boldsymbol{x}_0) &= \text{E}[(Y_i - \hat{f}(\boldsymbol{x}_i))^2 | \boldsymbol{x}_i = \boldsymbol{x}_0] \\ &= \text{E}[(f(\boldsymbol{x}_i) + \epsilon_i - \hat{f}(\boldsymbol{x}_i))^2 | \boldsymbol{x}_i = \boldsymbol{x}_0] \end{aligned}$$

Loss Functions and Model Complexity

Suppose $Y_i = f(\boldsymbol{x}_i) + \epsilon_i$
Where $E[\epsilon_i] = 0$
$\text{var}(\epsilon_i) = \sigma_\epsilon^2$
Define $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}) = \hat{f}(\boldsymbol{x})$
With squared error loss:

$$\begin{aligned}
\text{Error}(\boldsymbol{x}_0) &= \text{E}[(Y_i - \hat{f}(\boldsymbol{x}_i))^2 | \boldsymbol{x}_i = \boldsymbol{x}_0] \\
&= \text{E}[(f(\boldsymbol{x}_i) + \epsilon_i - \hat{f}(\boldsymbol{x}_i))^2 | \boldsymbol{x}_i = \boldsymbol{x}_0] \\
&= \sigma_\epsilon^2 + \left[f(\boldsymbol{x}_0) - \text{E}[\hat{f}(\boldsymbol{x}_0)]\right]^2 + E[\left(\hat{f}(\boldsymbol{x}_0) - E[\hat{f}(\boldsymbol{x}_0)]\right)^2]
\end{aligned}$$

Loss Functions and Model Complexity

Suppose $Y_i = f(\boldsymbol{x}_i) + \epsilon_i$
Where $E[\epsilon_i] = 0$
$\text{var}(\epsilon_i) = \sigma_\epsilon^2$
Define $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}) = \hat{f}(\boldsymbol{x})$
With squared error loss:

$$\begin{aligned}
\text{Error}(\boldsymbol{x}_0) &= \text{E}[(Y_i - \hat{f}(\boldsymbol{x}_i))^2 | \boldsymbol{x}_i = \boldsymbol{x}_0] \\
&= \text{E}[(f(\boldsymbol{x}_i) + \epsilon_i - \hat{f}(\boldsymbol{x}_i))^2 | \boldsymbol{x}_i = \boldsymbol{x}_0] \\
&= \sigma_\epsilon^2 + \left[f(\boldsymbol{x}_0) - \text{E}[\hat{f}(\boldsymbol{x}_0)]\right]^2 + E[\left(\hat{f}(\boldsymbol{x}_0) - E[\hat{f}(\boldsymbol{x}_0)]\right)^2] \\
&= \text{Irreducible error} + \text{Bias}^2 + \text{Variance}
\end{aligned}$$

Probit Regression (for motivational purposes)

Suppose:

$$\begin{aligned} Y_i &\sim \text{Bernoulli}(\pi_i) \\ \pi_i &= \Phi(\boldsymbol{\beta}' \boldsymbol{x}_i) \end{aligned}$$

where $\Phi(\cdot)$ is the cumulative normal distribution.
Implies log-likelihood

$$\log \mathrm{L}(\boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{N} \left[Y_i \log \Phi(\boldsymbol{\beta}' \boldsymbol{x}_i) + (1 - Y_i) \log(1 - \Phi(\boldsymbol{\beta}' \boldsymbol{x}_i)) \right]$$

Log-likelihood is a loss function $\rightsquigarrow$ overly optimistic: improves with more parameters

How Do We Build A Model?

There are many ways to fit models

And many choices made when performing model fit

How do we choose?

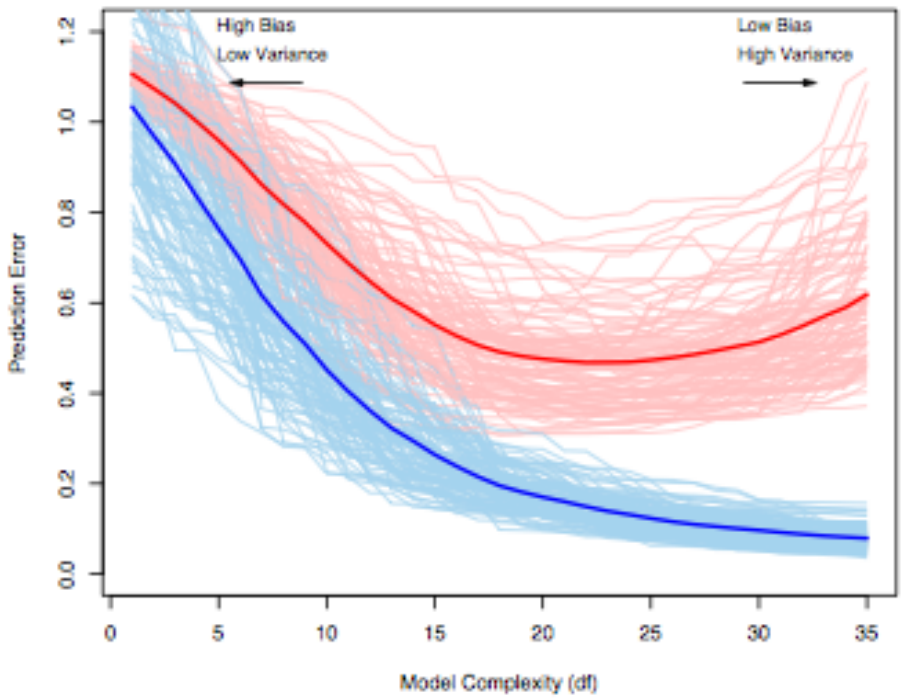


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$*, while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\mathrm{E}[\overline{\mathrm{err}}]$.

How Do We Build A Model?

There are many ways to fit models

And many choices made when performing model fit

How do we choose?

Bad way to choose:

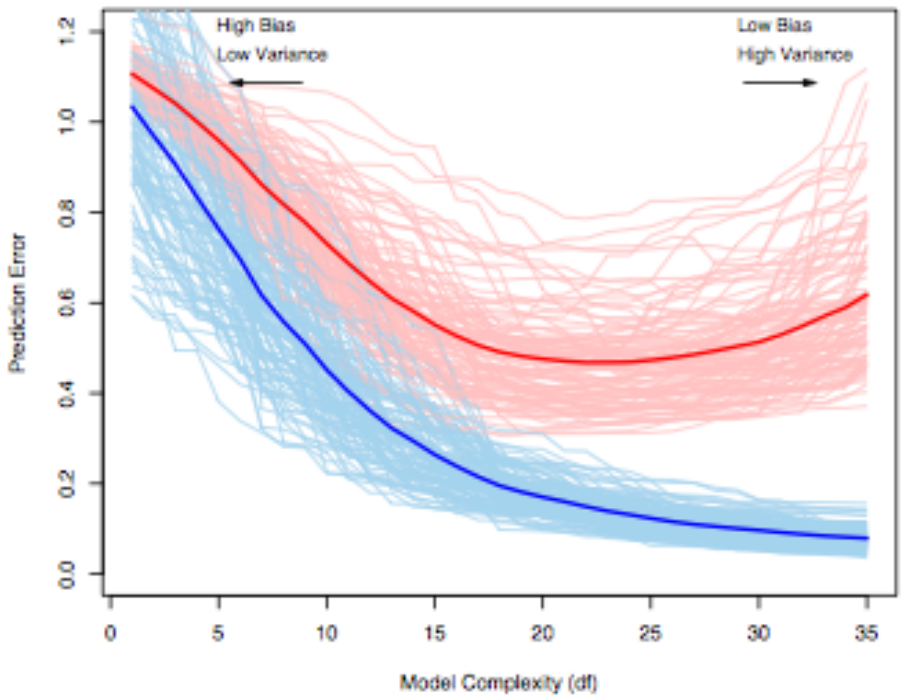


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$*, while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\mathrm{E}[\overline{\mathrm{err}}]$.

How Do We Build A Model?

There are many ways to fit models

And many choices made when performing model fit

How do we choose?

Bad way to choose: within sample model fit (HTF Figure 7.1)

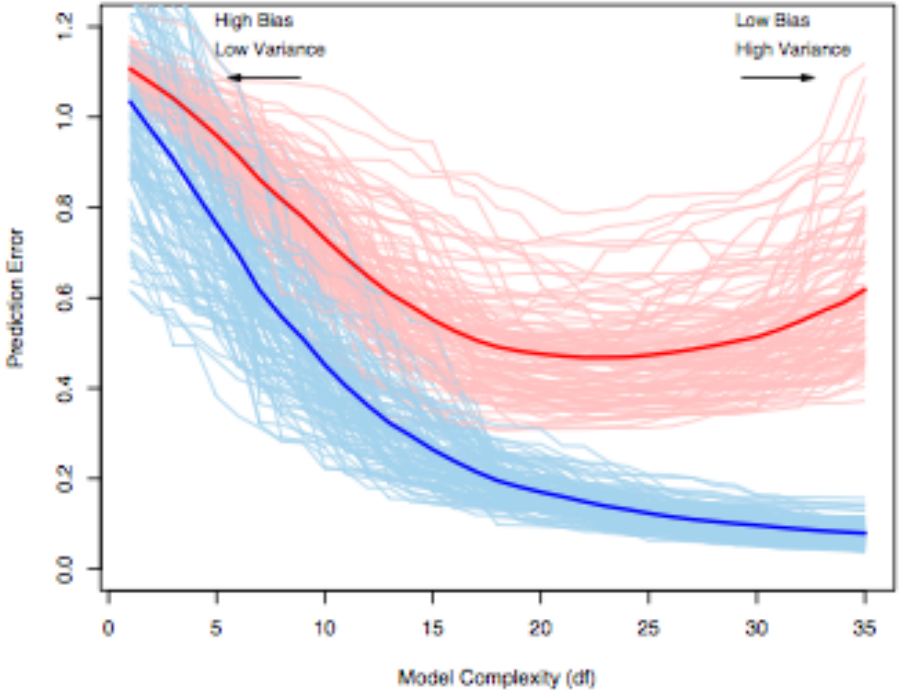


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$*, while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\mathrm{E}[\overline{\mathrm{err}}]$.

How Do We Build A Model?

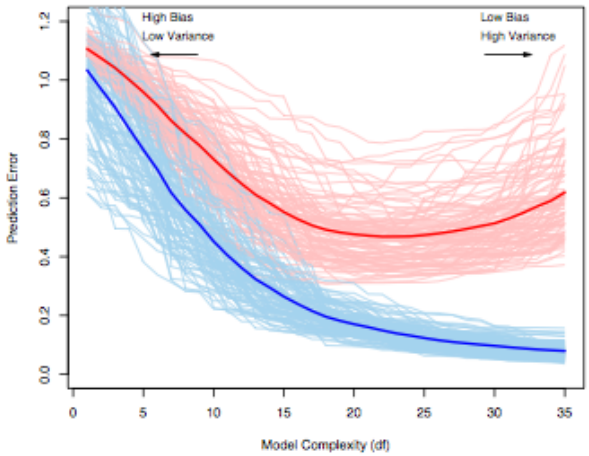


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$*, while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\mathrm{E}[\overline{\mathrm{err}}]$.

Model overfit $\rightsquigarrow$ in sample error is optimistic:

How Do We Build A Model?

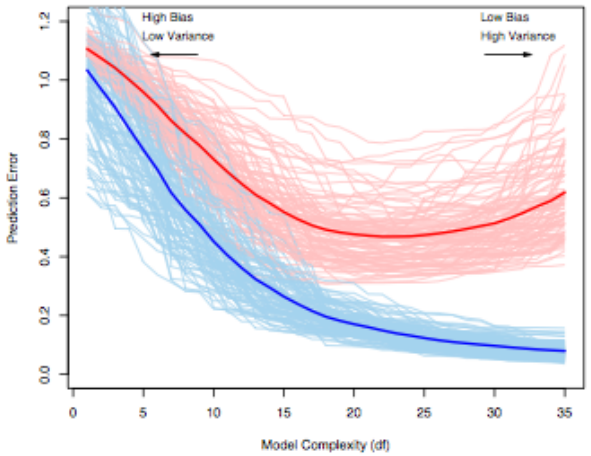


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$*, while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\mathrm{E}[\overline{\mathrm{err}}]$.

Model overfit $\rightsquigarrow$ in sample error is optimistic:

- Some model complexity captures systematic features of the data

How Do We Build A Model?

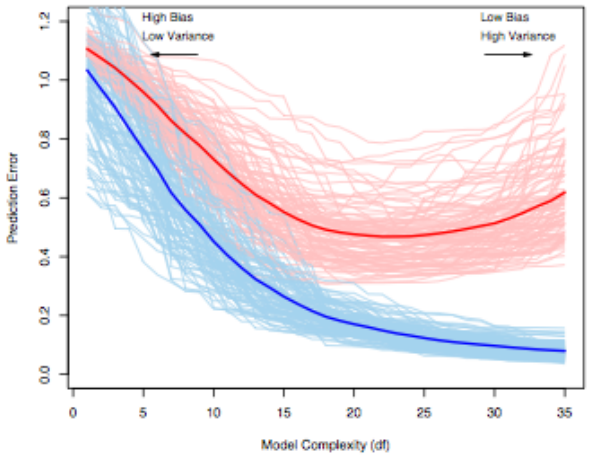


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$*, while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\mathrm{E}[\overline{\mathrm{err}}]$.

Model overfit $\rightsquigarrow$ in sample error is optimistic:

- Some model complexity captures systematic features of the data
- Characteristics found in both training and test set

How Do We Build A Model?

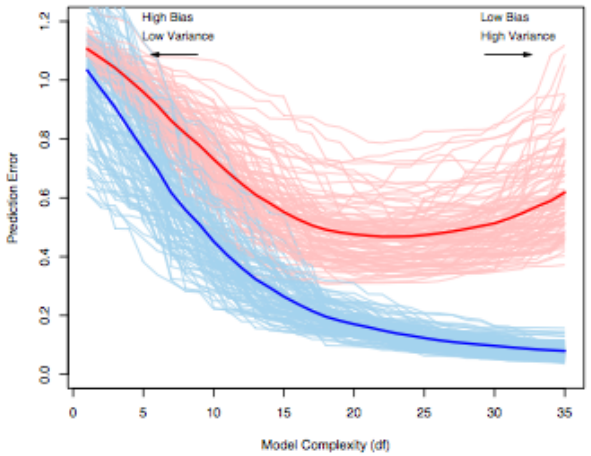


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$*, while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\mathrm{E}[\overline{\mathrm{err}}]$.

Model overfit$\rightsquigarrow$ in sample error is optimistic:

- Some model complexity captures systematic features of the data
- Characteristics found in both training and test set
- Reduces error in both training and test set

How Do We Build A Model?

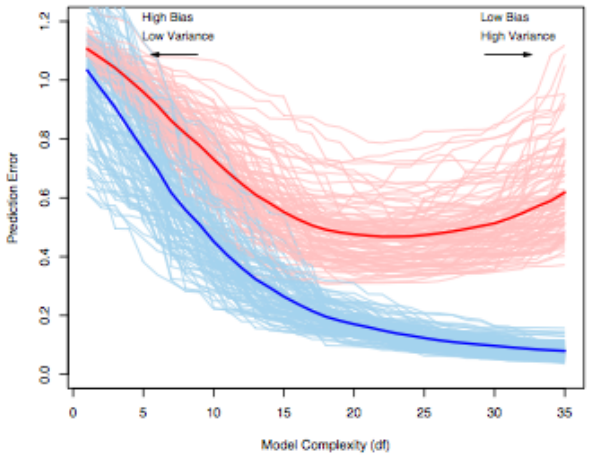


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$*, while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\mathrm{E}[\overline{\mathrm{err}}]$.

Model overfit$\rightsquigarrow$ in sample error is optimistic:

- Some model complexity captures systematic features of the data
- Characteristics found in both training and test set
- Reduces error in both training and test set
- Additional model complexity: idiosyncratic features of the training set

How Do We Build A Model?

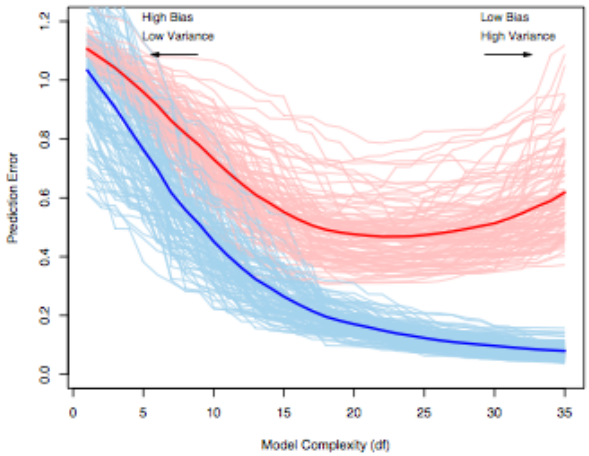


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\text{E}[\overline{\text{err}}]$.

Model overfit$\rightsquigarrow$ in sample error is optimistic:

- Some model complexity captures systematic features of the data
- Characteristics found in both training and test set
- Reduces error in both training and test set
- Additional model complexity: idiosyncratic features of the training set
- Reduces error in training set, increases error in test set

How Do We Choose Covariates?

Best model depends on task

- Causal inference observational study: make treatment assignment ignorable
- Prediction: improve predictive performance

Stepwise Regression

Suppose we have P covariates.
2^P potential models

Stepwise Regression

Suppose we have P covariates.
2^P potential models
Stepwise procedures

Stepwise Regression

Suppose we have P covariates.
2^P potential models
Stepwise procedures

1) Forward selection
    a) No variables in model.
    b) Check all variables p-value if include, include lowest p-value
    c) Repeat until included p-value is above some threshold

Stepwise Regression

Suppose we have P covariates.
2^P potential models
Stepwise procedures

1) Forward selection
   a) No variables in model.
   b) Check all variables p-value if include, include lowest p-value
   c) Repeat until included p-value is above some threshold
2) Backward elimination
   a) Fit model with all variables (if possible)
   b) Remove variable with largest p-value
   c) Repeat until potentially excluded p-value is below some threshold

Stepwise Regression

Suppose we have P covariates.
2^P potential models
Stepwise procedures

1) Forward selection
   a) No variables in model.
   b) Check all variables p-value if include, include lowest p-value
   c) Repeat until included p-value is above some threshold
2) Backward elimination
   a) Fit model with all variables (if possible)
   b) Remove variable with largest p-value
   c) Repeat until potentially excluded p-value is below some threshold

Problematic:

1) Not optimal model selection (path dependent)
2) P-value $\neq$ objective of model

Analytic Solutions

Approximate optimism and compensate in loss function.

Analytic Solutions

Approximate optimism and compensate in loss function.

Akaike Information Criterion (AIC) $\rightsquigarrow$ Minimize

Analytic Solutions

Approximate optimism and compensate in loss function.
Akaike Information Criterion (AIC) $\rightsquigarrow$ Minimize
As $N \to \infty$

Analytic Solutions

Approximate optimism and compensate in loss function.
Akaike Information Criterion (AIC) $\rightsquigarrow$ Minimize
As $N \to \infty$

$$-2\mathrm{E}[\log P_{\hat{\boldsymbol{\beta}}}(Y)] \quad = \quad -2\left[\mathrm{E}[\log \mathrm{L}(\hat{\boldsymbol{\beta}}|\boldsymbol{X}, \boldsymbol{Y})] - d\right]$$

Analytic Solutions

Approximate optimism and compensate in loss function.
Akaike Information Criterion (AIC) $\rightsquigarrow$ Minimize
As $N \to \infty$

$$
\begin{aligned}
-2\mathrm{E}[\log P_{\hat{\boldsymbol{\beta}}}(Y)] &= -2\left[\mathrm{E}[\log \mathrm{L}(\hat{\boldsymbol{\beta}}|\boldsymbol{X}, \boldsymbol{Y})] - d\right] \\
\mathrm{AIC} &= -2\left[\log \mathrm{L}(\hat{\boldsymbol{\beta}}|\boldsymbol{X}, \boldsymbol{Y}) - d\right]
\end{aligned}
$$

Analytic Solutions

Approximate optimism and compensate in loss function.
Akaike Information Criterion (AIC) $\rightsquigarrow$ Minimize
As $N \to \infty$

$$
\begin{aligned}
-2\mathrm{E}[\log P_{\hat{\boldsymbol{\beta}}}(Y)] &= -2\left[\mathrm{E}[\log \mathrm{L}(\hat{\boldsymbol{\beta}}|\boldsymbol{X}, \boldsymbol{Y})] - d\right] \\
\mathrm{AIC} &= -2\left[\log \mathrm{L}(\hat{\boldsymbol{\beta}}|\boldsymbol{X}, \boldsymbol{Y}) - d\right]
\end{aligned}
$$

where d are the number of parameters in the model

Analytic Solutions

Approximate optimism and compensate in loss function.
Akaike Information Criterion (AIC) $\rightsquigarrow$ Minimize
As $N \to \infty$

$$
\begin{aligned}
-2\mathrm{E}[\log P_{\hat{\boldsymbol{\beta}}}(Y)] &= -2\left[\mathrm{E}[\log \mathrm{L}(\hat{\boldsymbol{\beta}}|\boldsymbol{X}, \boldsymbol{Y})] - d\right] \\
\mathrm{AIC} &= -2\left[\log \mathrm{L}(\hat{\boldsymbol{\beta}}|\boldsymbol{X}, \boldsymbol{Y}) - d\right]
\end{aligned}
$$

where d are the number of parameters in the model

- Intuition: balances model fit with penalty for complexity

Analytic Solutions

Approximate optimism and compensate in loss function.
Akaike Information Criterion (AIC) $\rightsquigarrow$ Minimize
As $N \to \infty$

$$
\begin{aligned}
-2\mathrm{E}[\log P_{\hat{\boldsymbol{\beta}}}(Y)] &= -2\left[\mathrm{E}[\log \mathrm{L}(\hat{\boldsymbol{\beta}}|\boldsymbol{X}, \boldsymbol{Y})] - d\right] \\
\mathrm{AIC} &= -2\left[\log \mathrm{L}(\hat{\boldsymbol{\beta}}|\boldsymbol{X}, \boldsymbol{Y}) - d\right]
\end{aligned}
$$

where d are the number of parameters in the model

- Intuition: balances model fit with penalty for complexity
- Derived from method to estimate optimism in likelihood based models

Analytic Solutions

Approximate optimism and compensate in loss function.
Akaike Information Criterion (AIC) $\rightsquigarrow$ Minimize
As $N \to \infty$

$$
\begin{aligned}
-2\mathrm{E}[\log P_{\hat{\beta}}(Y)] &= -2\left[\mathrm{E}[\log \mathrm{L}(\hat{\beta}|\boldsymbol{X}, \boldsymbol{Y})] - d\right] \\
\mathrm{AIC} &= -2\left[\log \mathrm{L}(\hat{\beta}|\boldsymbol{X}, \boldsymbol{Y}) - d\right]
\end{aligned}
$$

where d are the number of parameters in the model

- Intuition: balances model fit with penalty for complexity
- Derived from method to estimate optimism in likelihood based models
- Derived from a method to compute similarity between estimated model and true model (under assumptions of course)

Analytic Solutions

Approximate optimism and compensate in loss function.
Akaike Information Criterion (AIC) $\rightsquigarrow$ Minimize
As $N \to \infty$

$$
\begin{aligned}
-2\mathrm{E}[\log P_{\hat{\beta}}(Y)] &= -2\left[\mathrm{E}[\log \mathrm{L}(\hat{\beta}|\boldsymbol{X}, \boldsymbol{Y})] - d\right] \\
\mathrm{AIC} &= -2\left[\log \mathrm{L}(\hat{\beta}|\boldsymbol{X}, \boldsymbol{Y}) - d\right]
\end{aligned}
$$

where d are the number of parameters in the model

- Intuition: balances model fit with penalty for complexity
- Derived from method to estimate optimism in likelihood based models
- Derived from a method to compute similarity between estimated model and true model (under assumptions of course)
- Can be extended to general models, though requires estimate of irresolvable error

Analytic Solutions

Bayesian Information Criterion (BIC) [Schwarz Criterion]

Analytic Solutions

Bayesian Information Criterion (BIC) [Schwarz Criterion]

$$\text{BIC} = -2 \log \text{L}(\widehat{\beta} | \boldsymbol{X}, \boldsymbol{Y}) + (\log N) d$$

Analytic Solutions

Bayesian Information Criterion (BIC) [Schwarz Criterion]

$$\text{BIC} \quad = \quad -2 \log \mathrm{L}(\widehat{\beta} | \boldsymbol{X}, \boldsymbol{Y}) + (\log N) d$$

where d is again the effective number of parameters

Analytic Solutions

Bayesian Information Criterion (BIC) [Schwarz Criterion]

$$\text{BIC} \quad = \quad -2 \log \mathrm{L}(\widehat{\beta} | \boldsymbol{X}, \boldsymbol{Y}) + (\log N) d$$

where d is again the effective number of parameters

- Intuition: balances model fit with penalty for complexity

Analytic Solutions

Bayesian Information Criterion (BIC) [Schwarz Criterion]

$$\text{BIC} = -2 \log \text{L}(\widehat{\beta} | \boldsymbol{X}, \boldsymbol{Y}) + (\log N) d$$

where d is again the effective number of parameters

- Intuition: balances model fit with penalty for complexity
- Derived from Bayesian approach to model selection

Analytic Solutions

Bayesian Information Criterion (BIC) [Schwarz Criterion]

$$\text{BIC} = -2 \log \text{L}(\widehat{\beta} | \boldsymbol{X}, \boldsymbol{Y}) + (\log N) d$$

where d is again the effective number of parameters

- Intuition: balances model fit with penalty for complexity
- Derived from Bayesian approach to model selection
- Approximation to Bayes' factor

Analytic Solutions

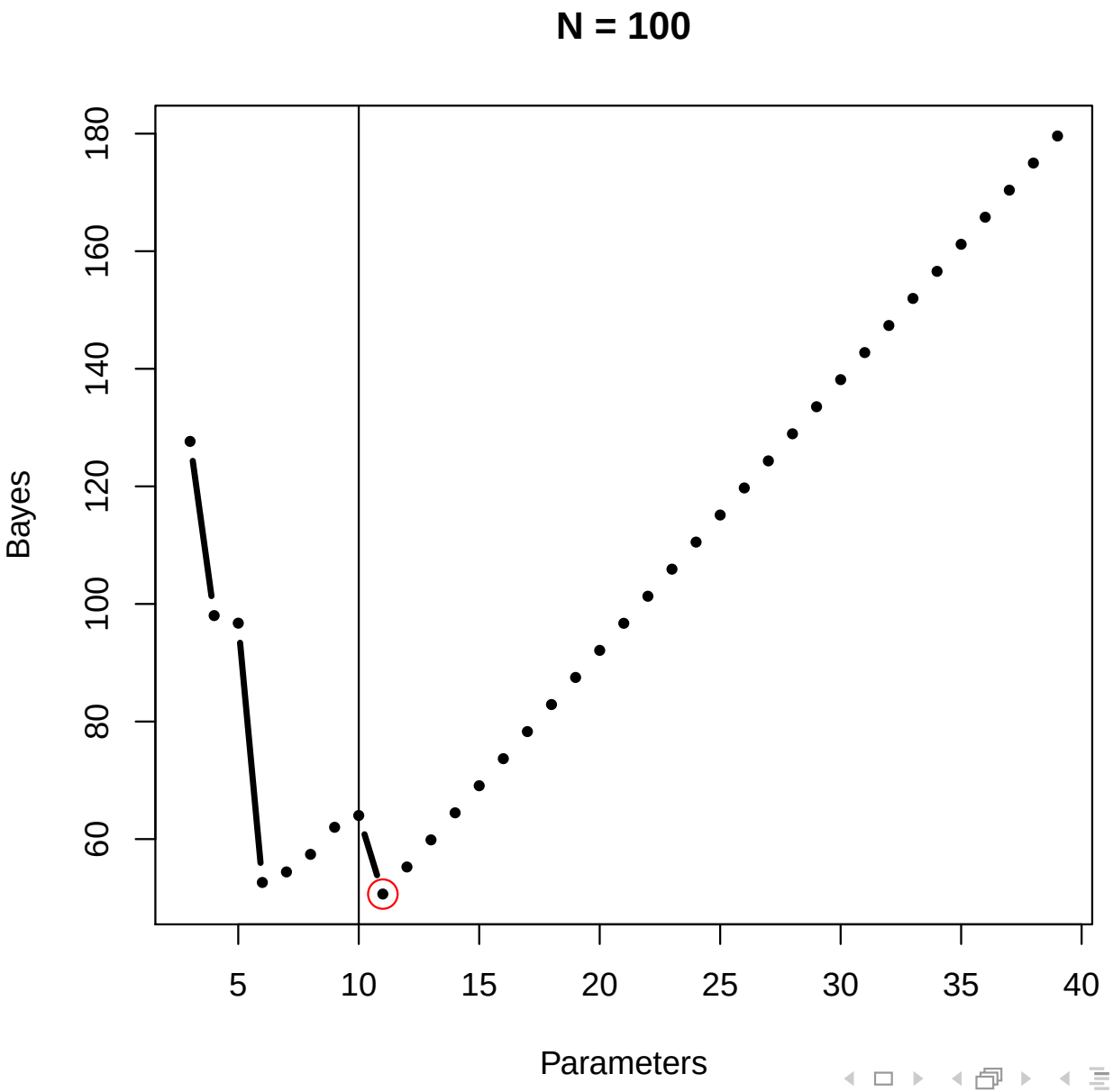
Bayesian Information Criterion (BIC) [Schwarz Criterion]

$$\text{BIC} = -2 \log \text{L}(\widehat{\beta} | \boldsymbol{X}, \boldsymbol{Y}) + (\log N) d$$

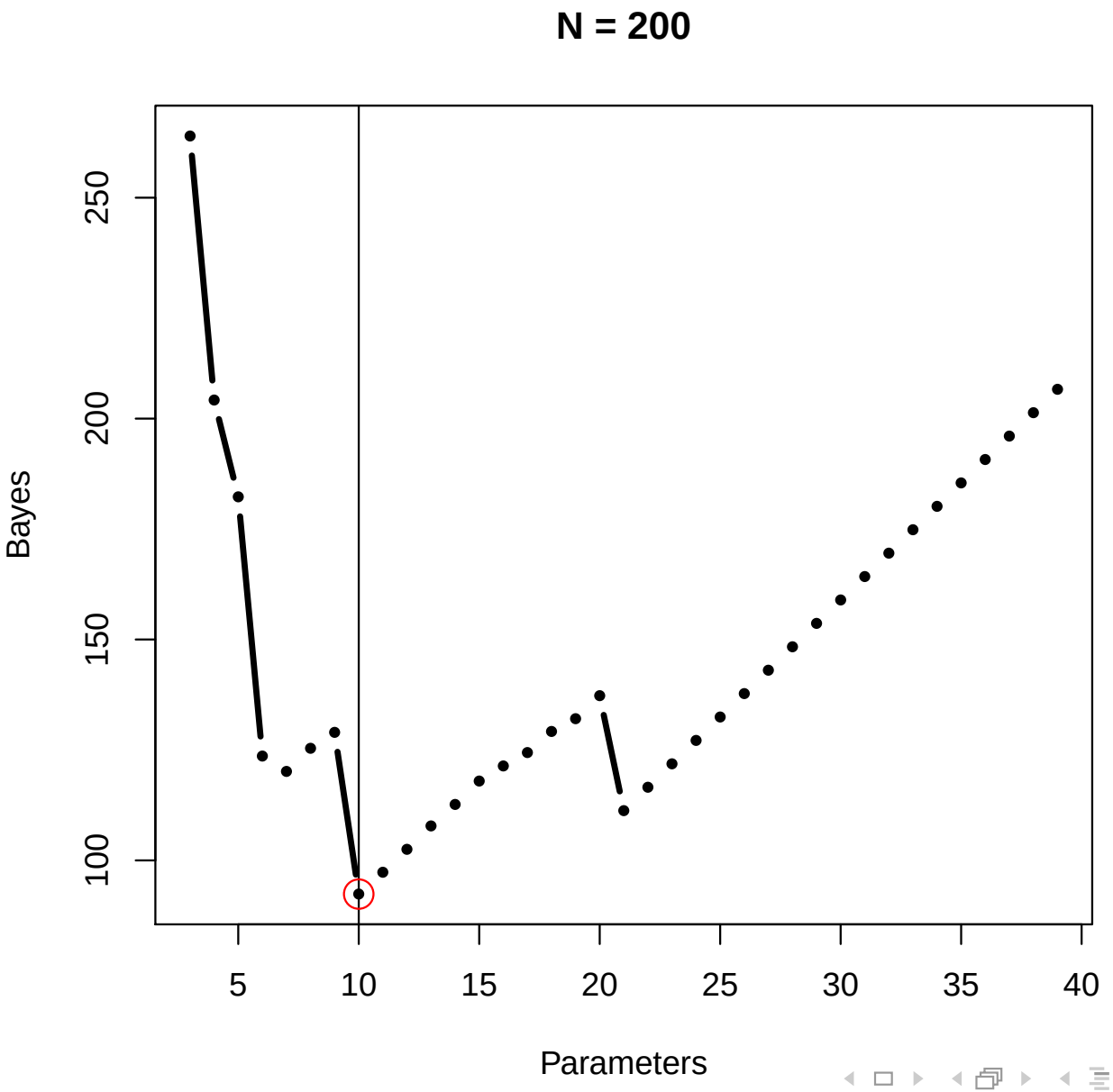
where d is again the effective number of parameters

- Intuition: balances model fit with penalty for complexity
- Derived from Bayesian approach to model selection
- Approximation to Bayes' factor
- Penalizes more heavily than AIC

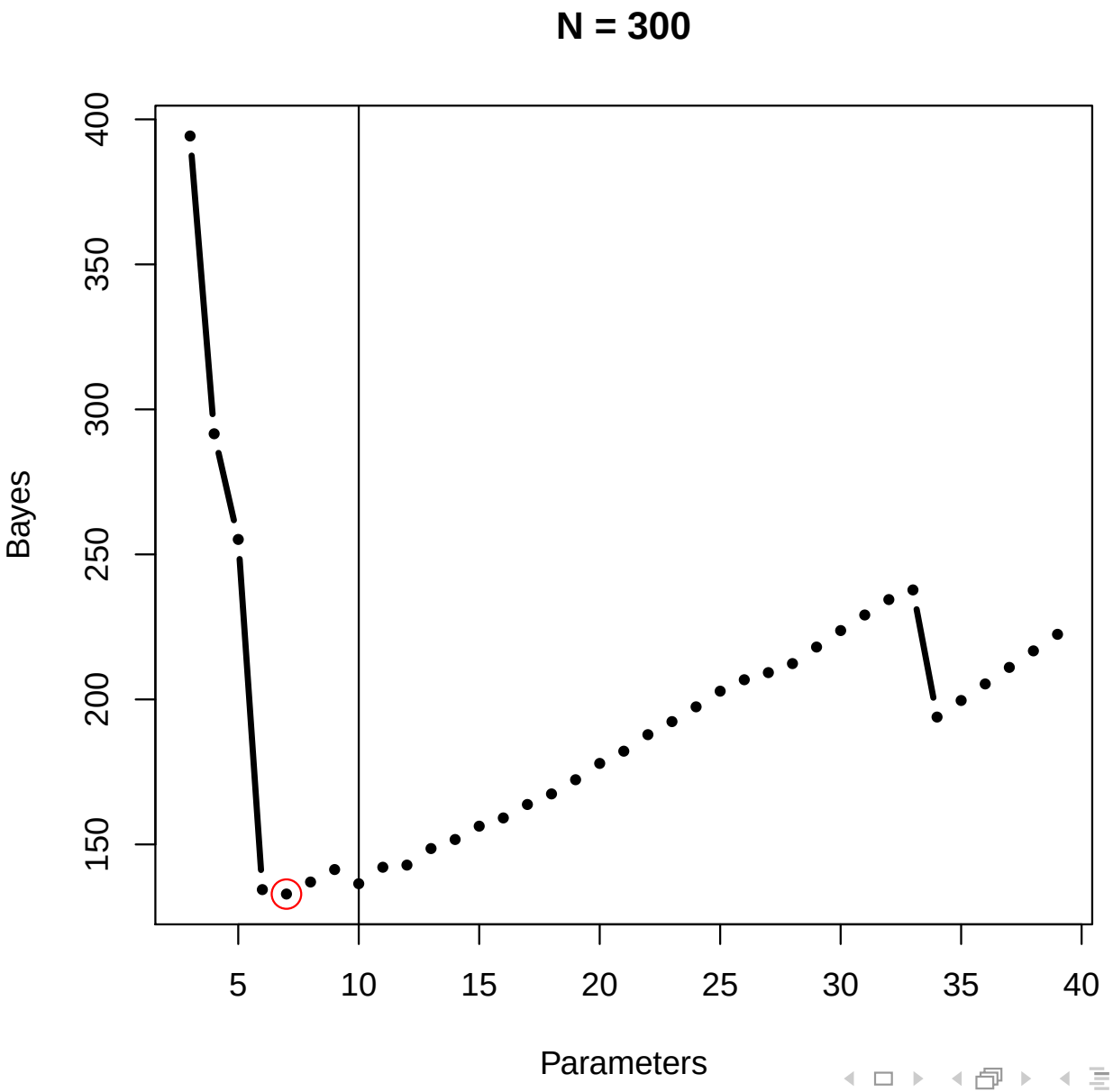
BIC or AIC?


N = 100
Bayes
60
80
100
120
140
160
180
5
10
15
20
25
30
35
40
Parameters

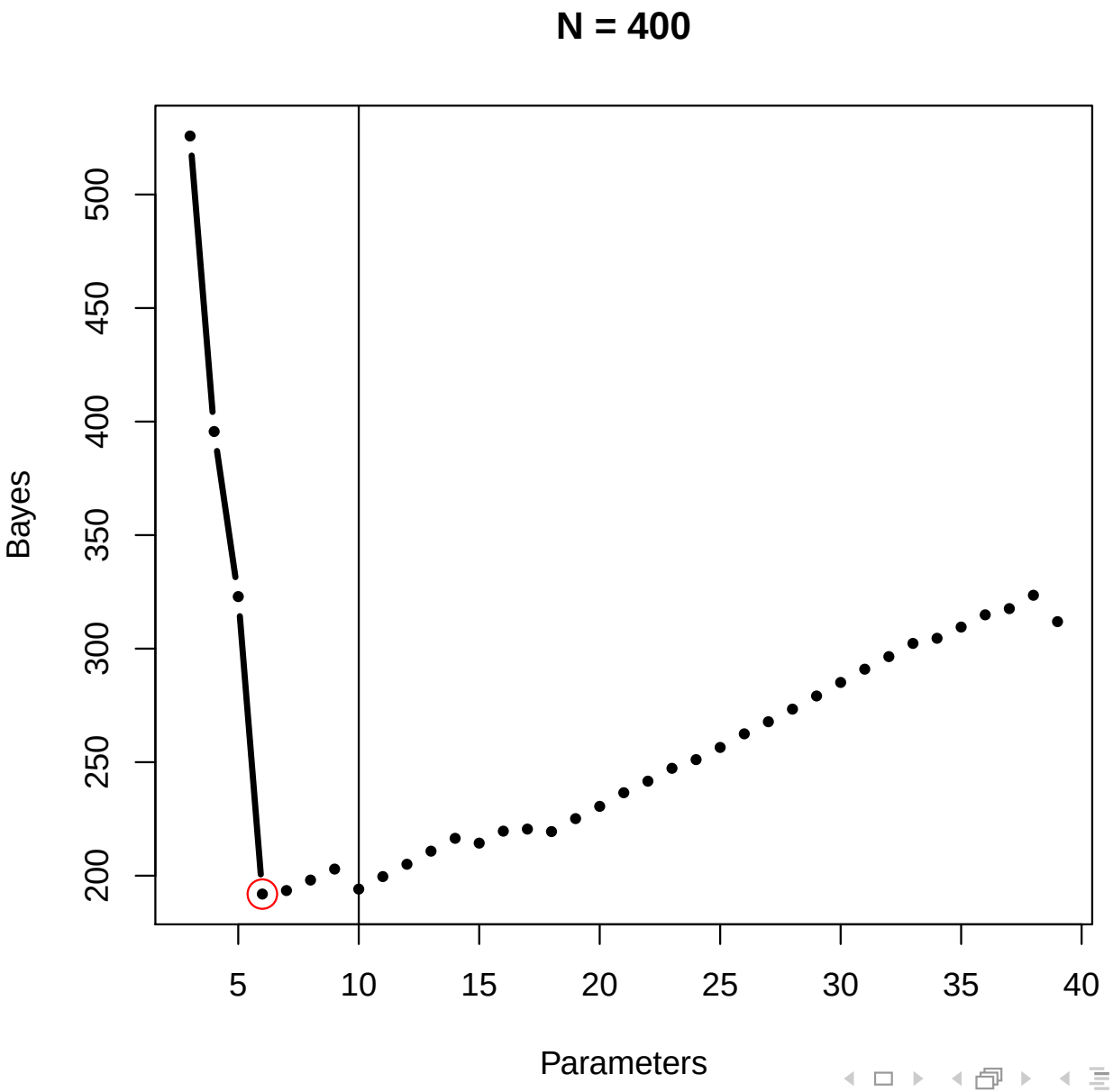
BIC or AIC?



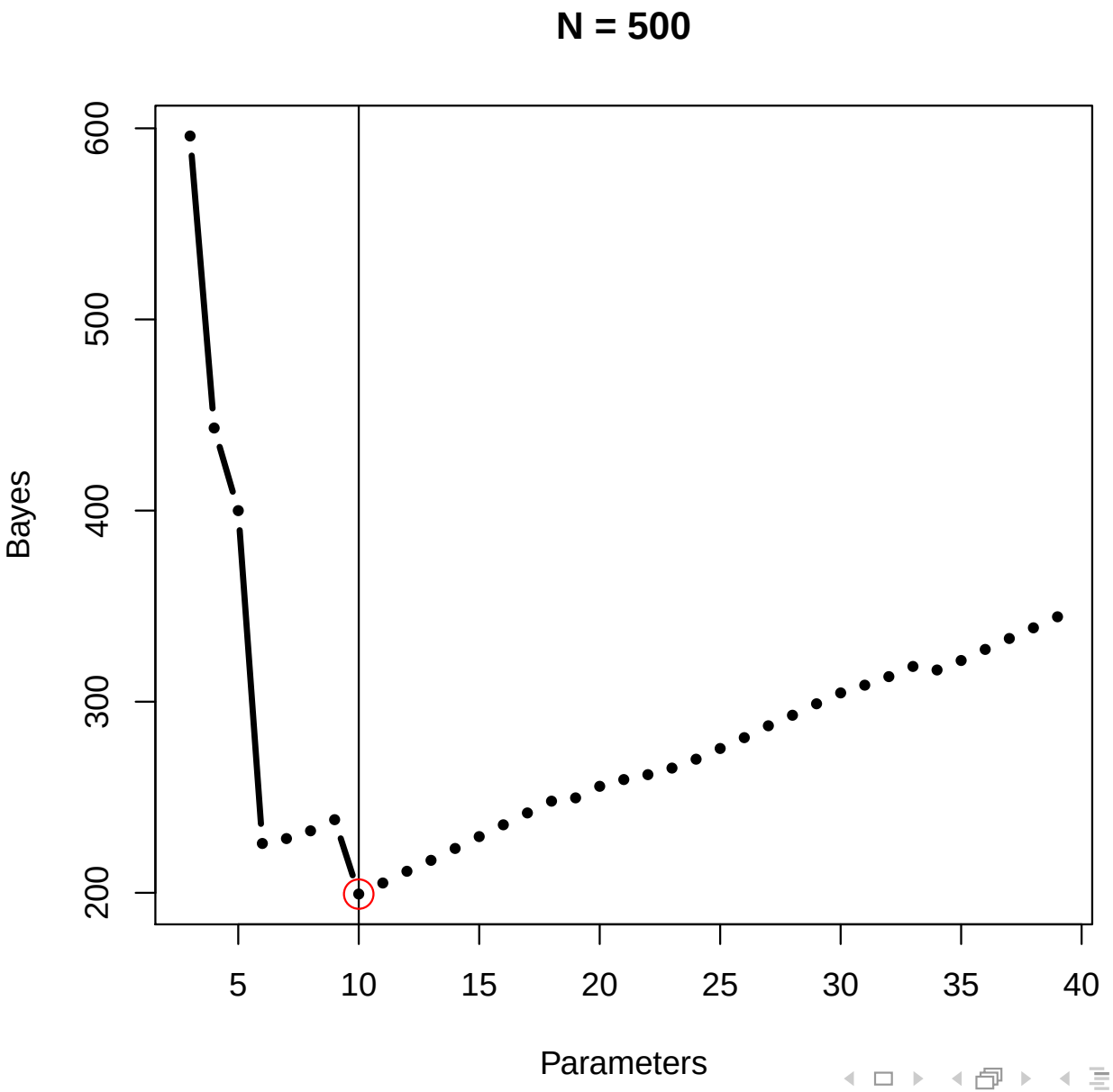
BIC or AIC?


N = 300
Bayes
400
350
300
250
200
150
5
10
15
20
25
30
35
40
Parameters

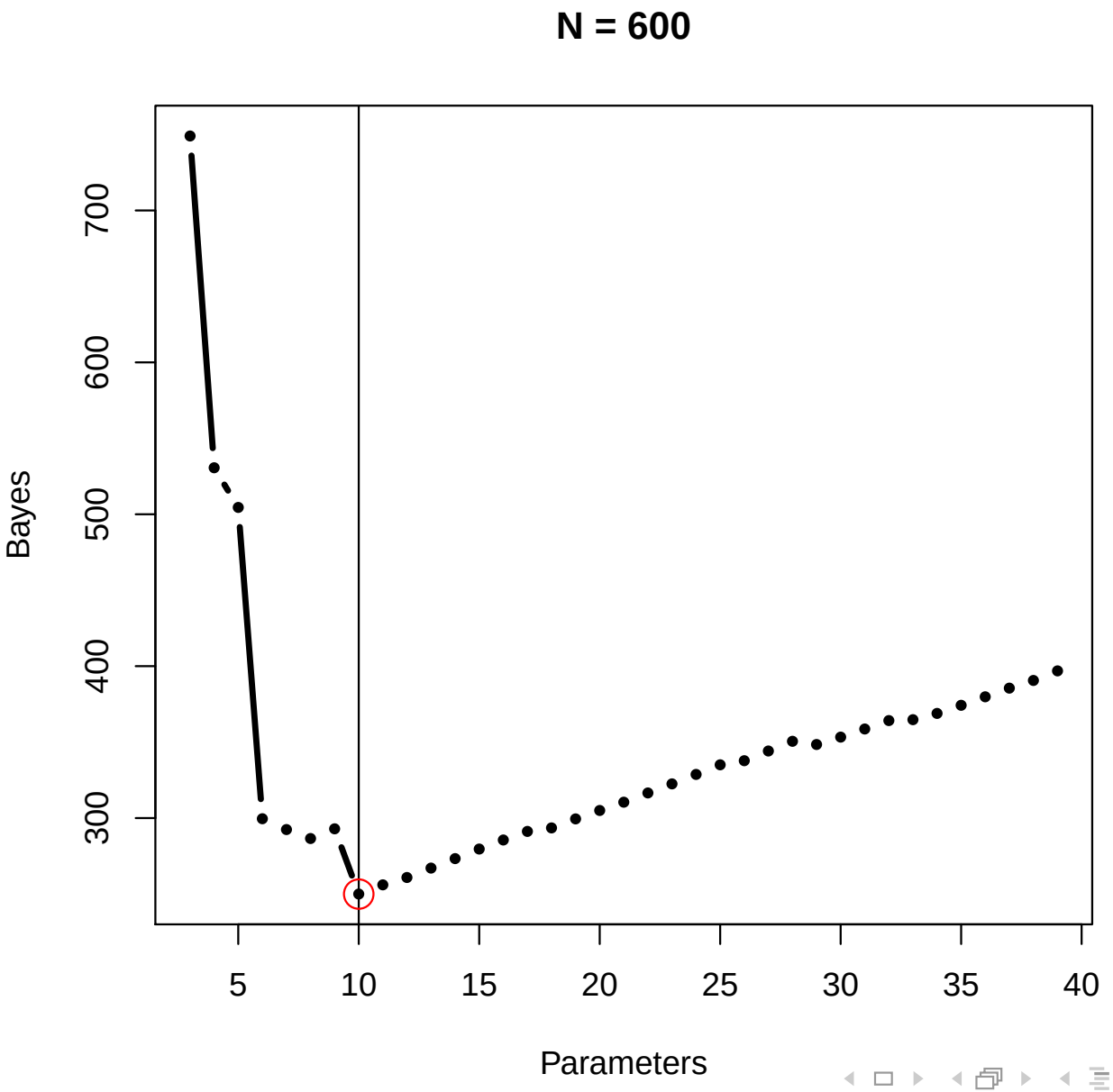
BIC or AIC?


N = 400
Bayes
200
250
300
350
400
450
500
5
10
15
20
25
30
35
40
Parameters

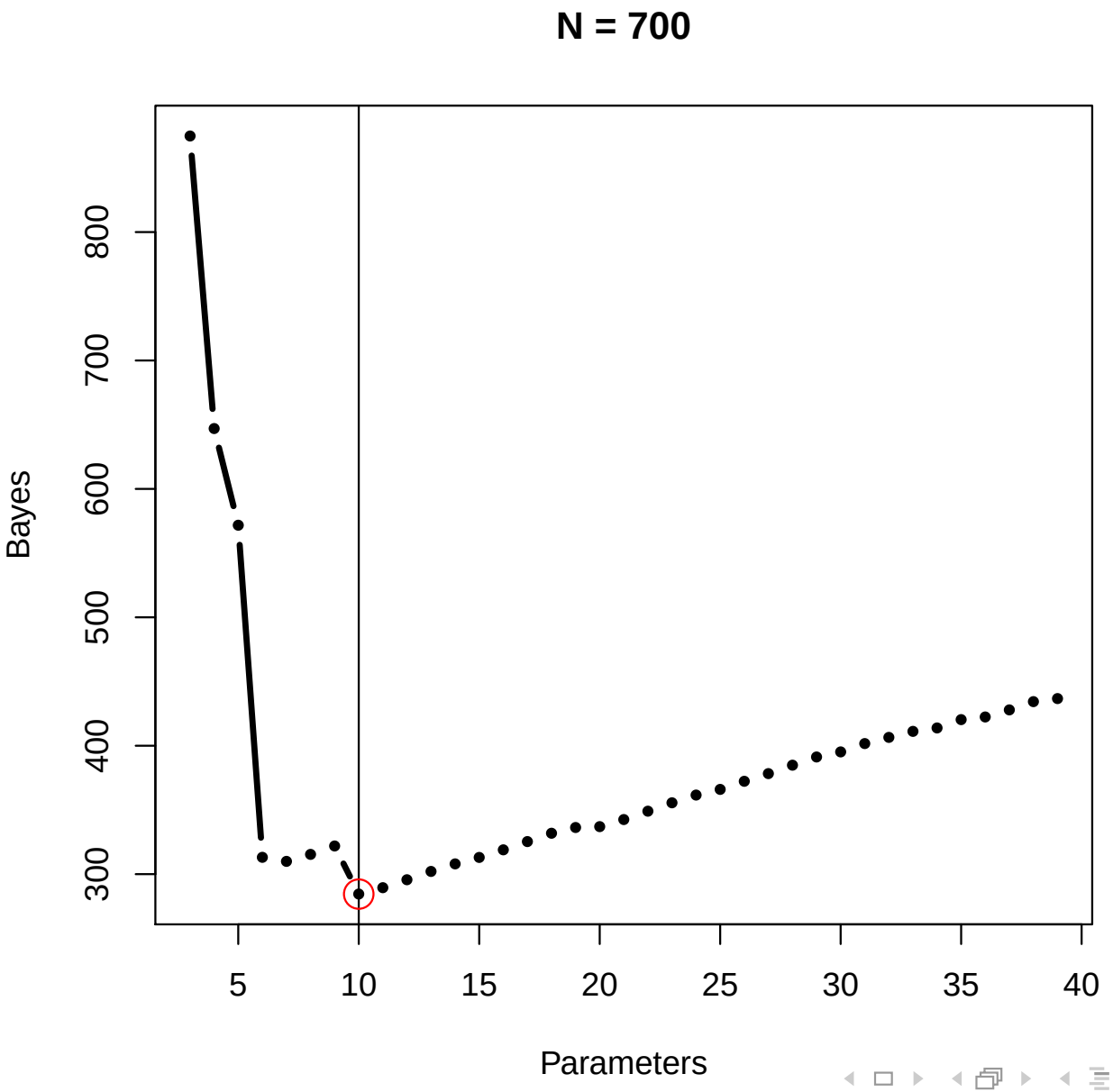
BIC or AIC?



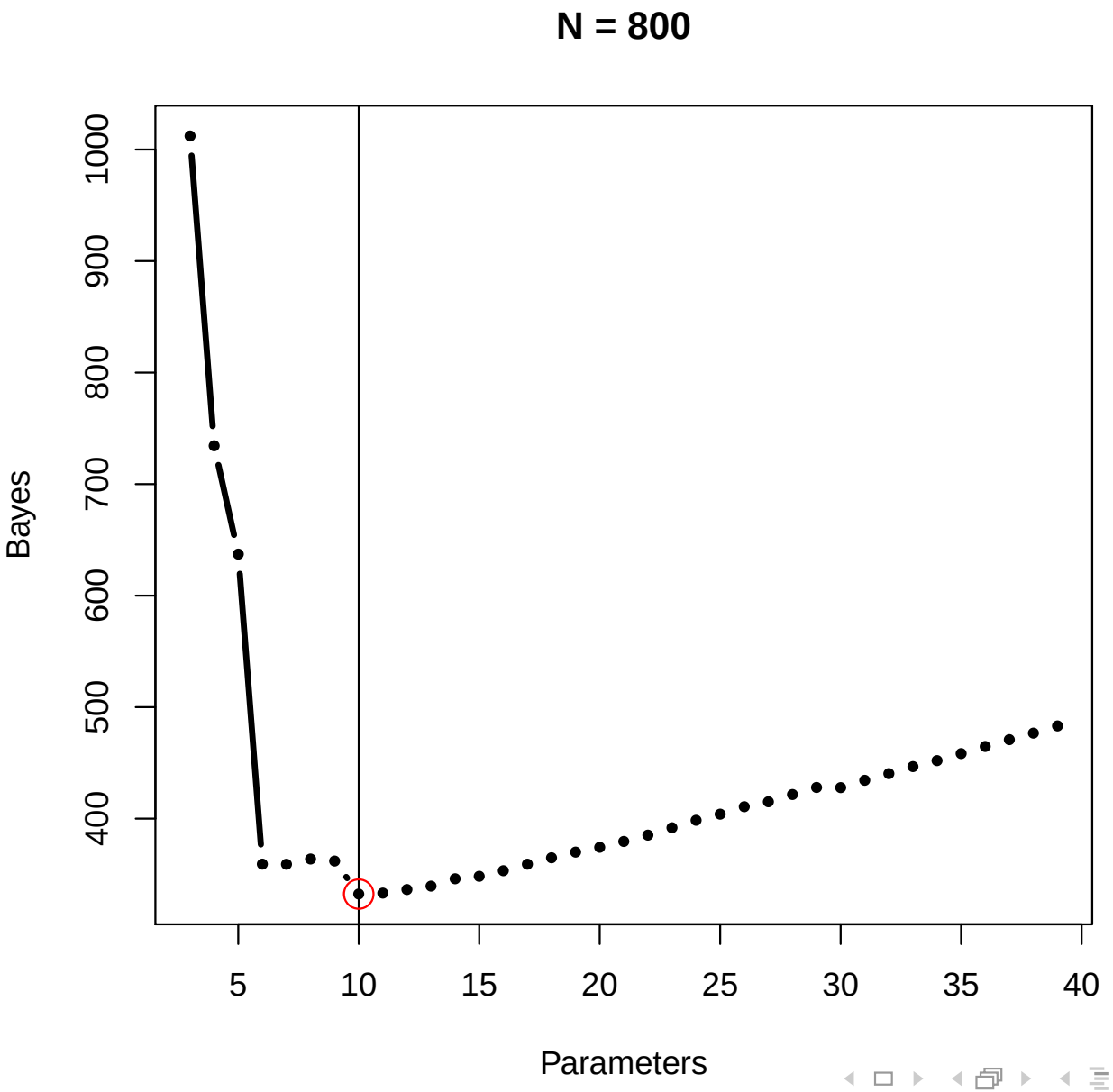
BIC or AIC?



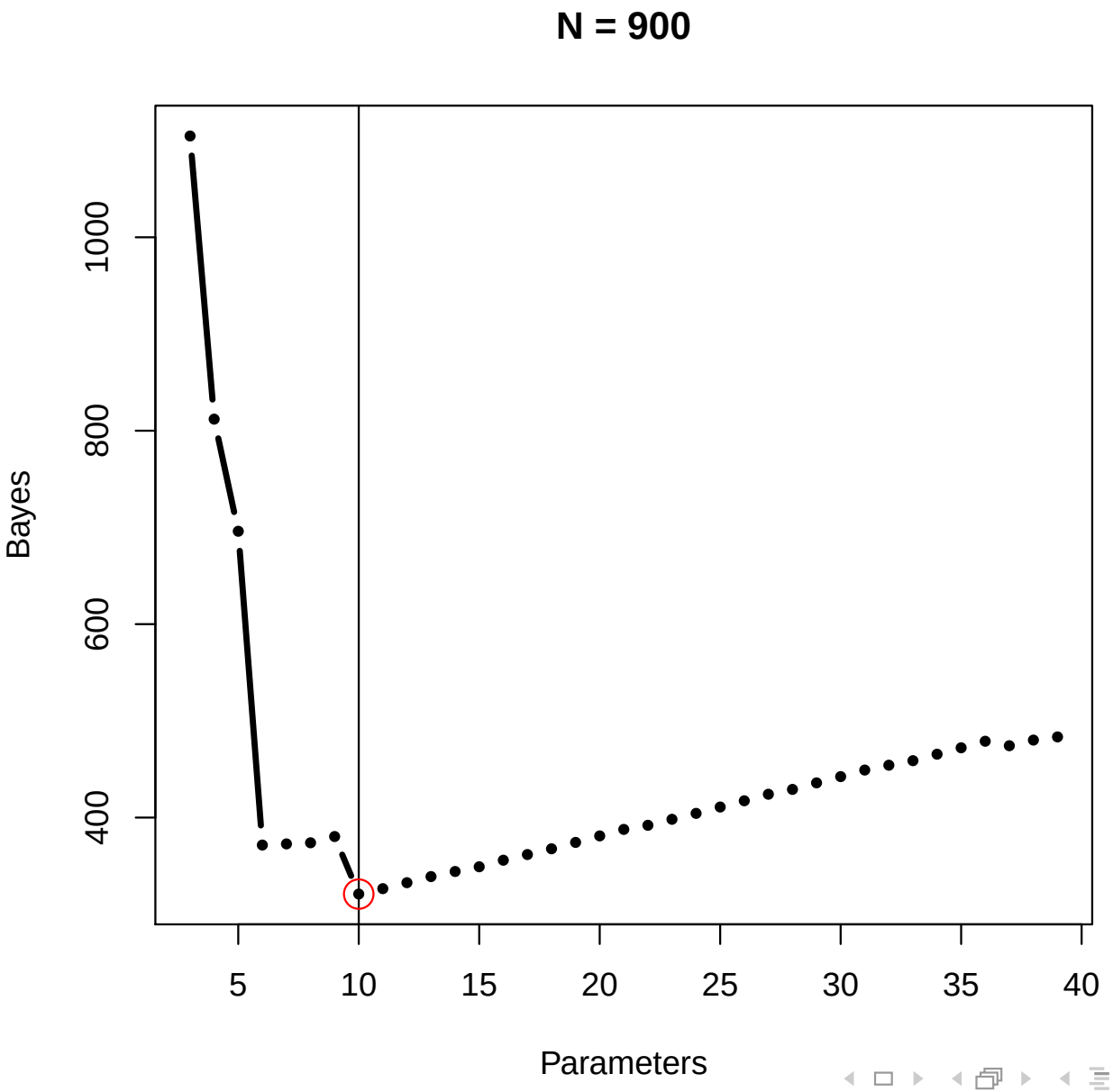
BIC or AIC?


N = 700
Bayes
300
400
500
600
700
800
5
10
15
20
25
30
35
40
Parameters

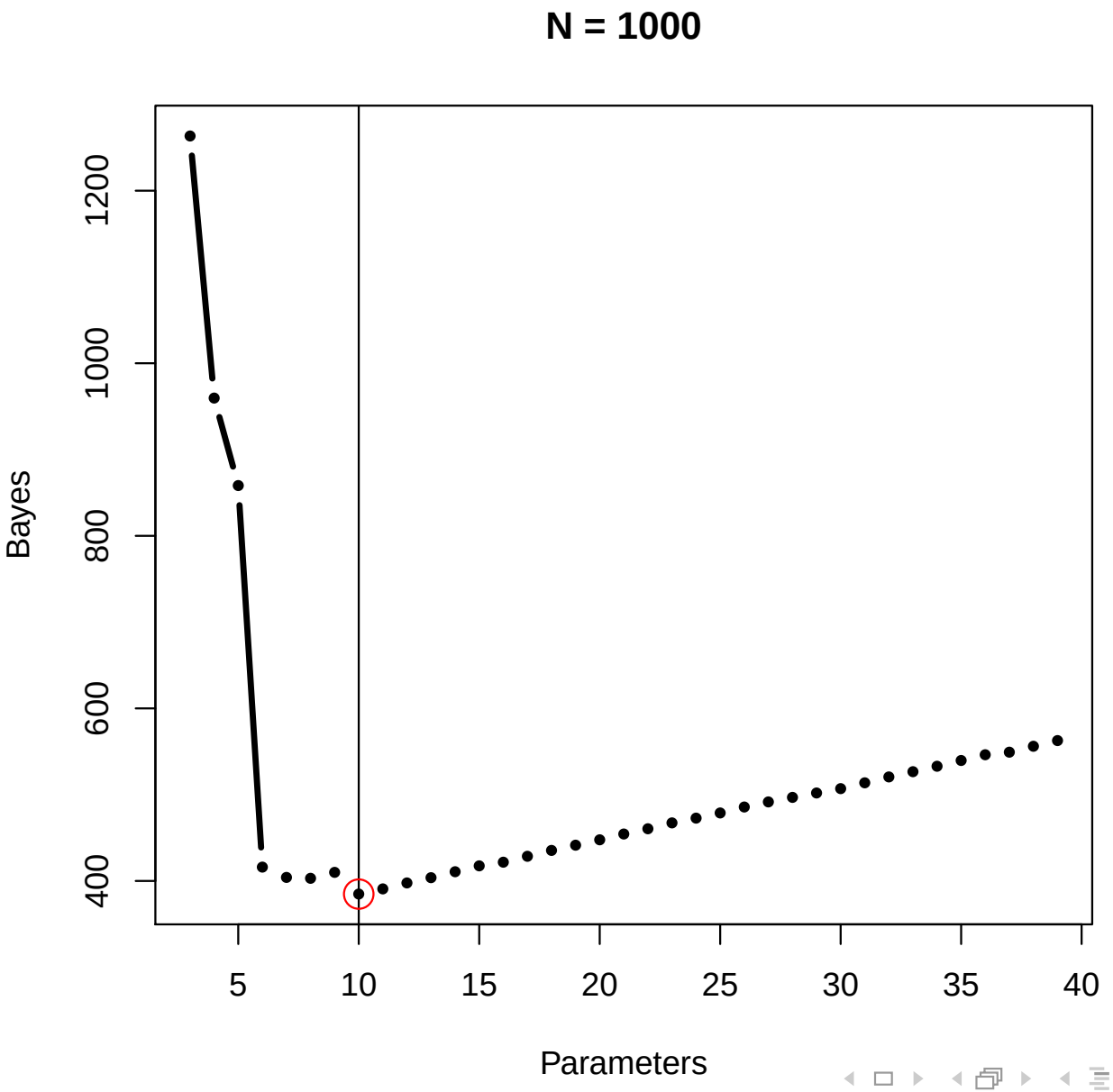
BIC or AIC?


N = 800
Bayes
400
500
600
700
800
900
1000
5
10
15
20
25
30
35
40
Parameters

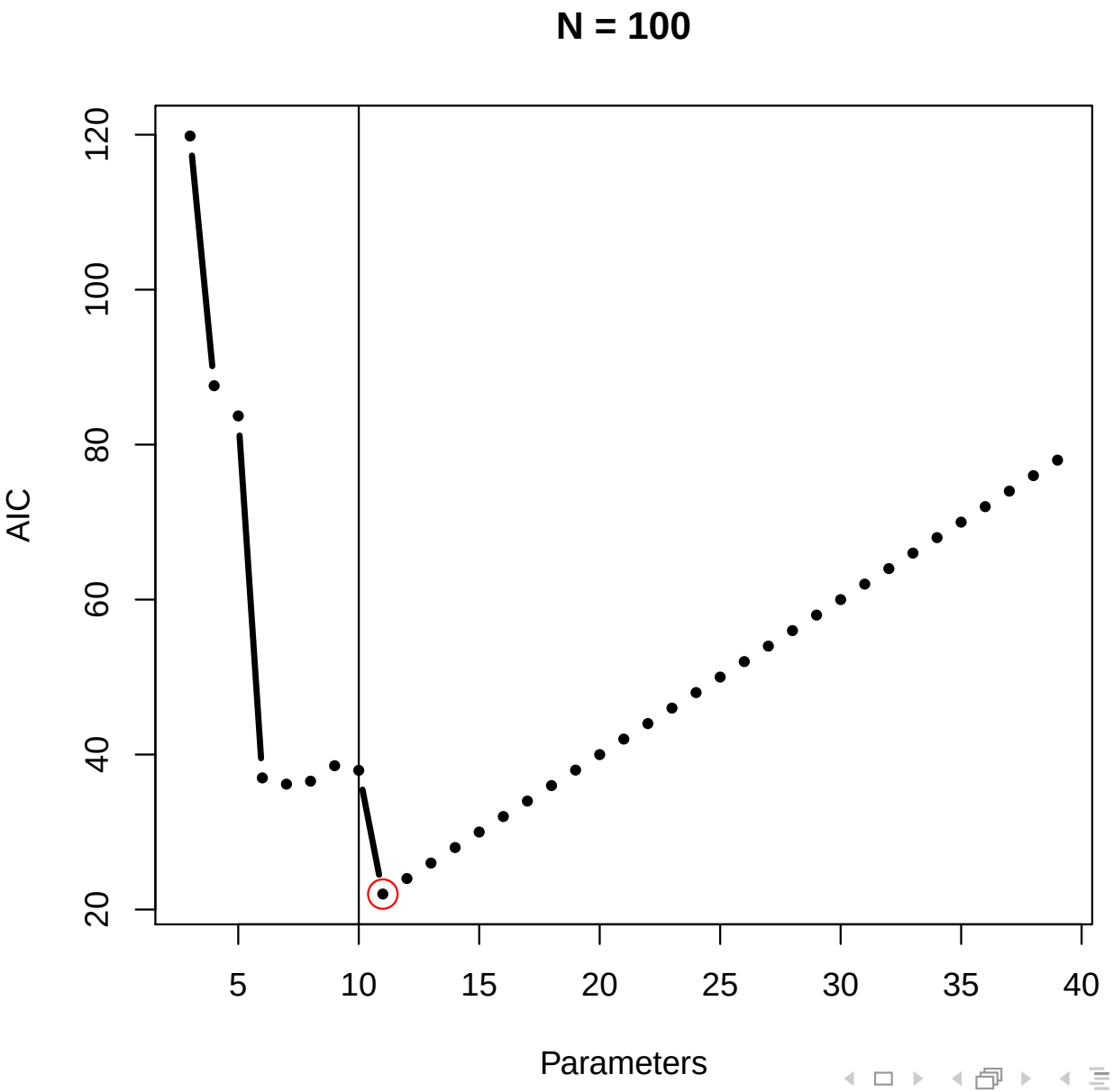
BIC or AIC?



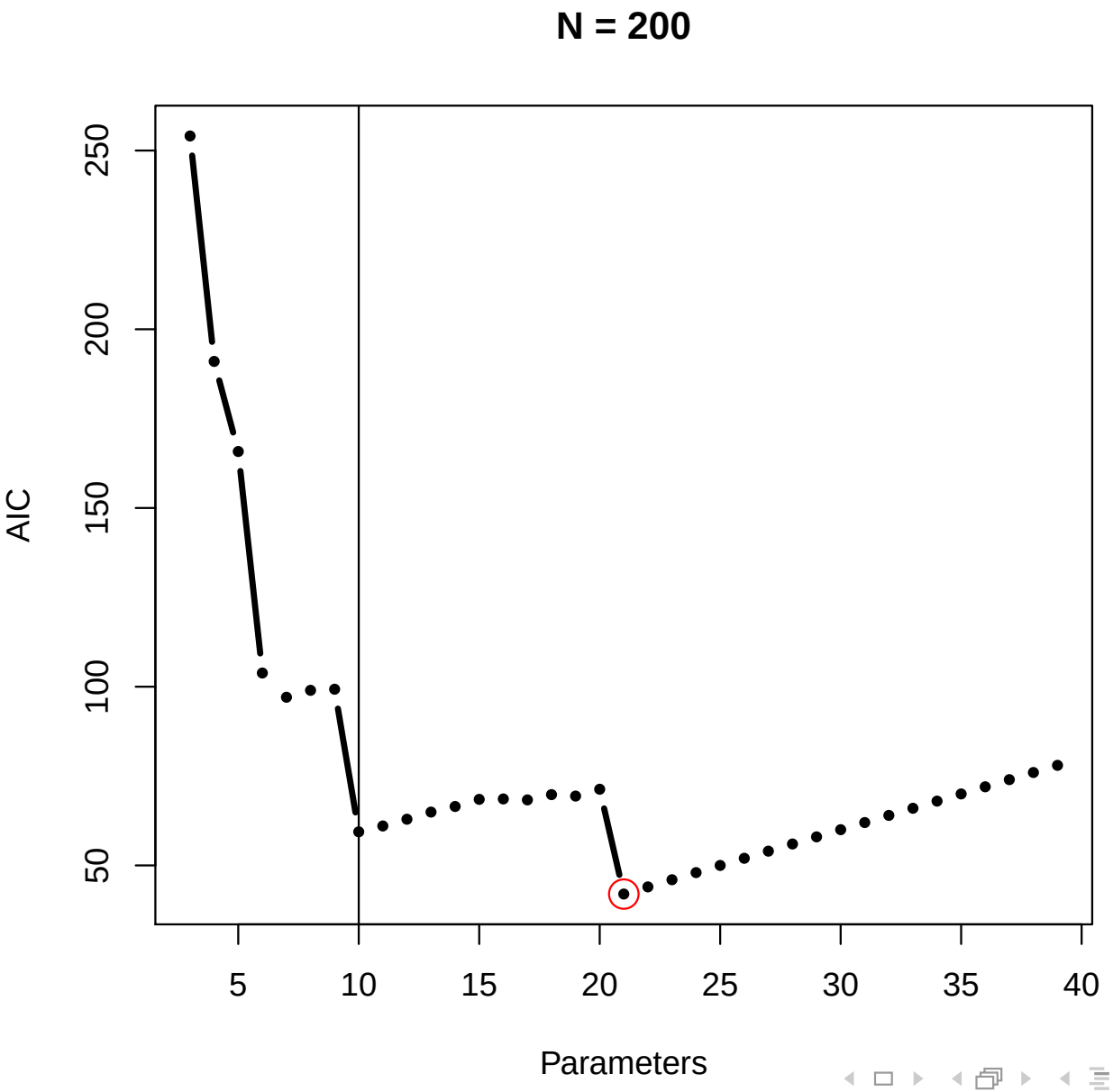
BIC or AIC?


N = 1000
Bayes
400
600
800
1000
1200
5
10
15
20
25
30
35
40
Parameters

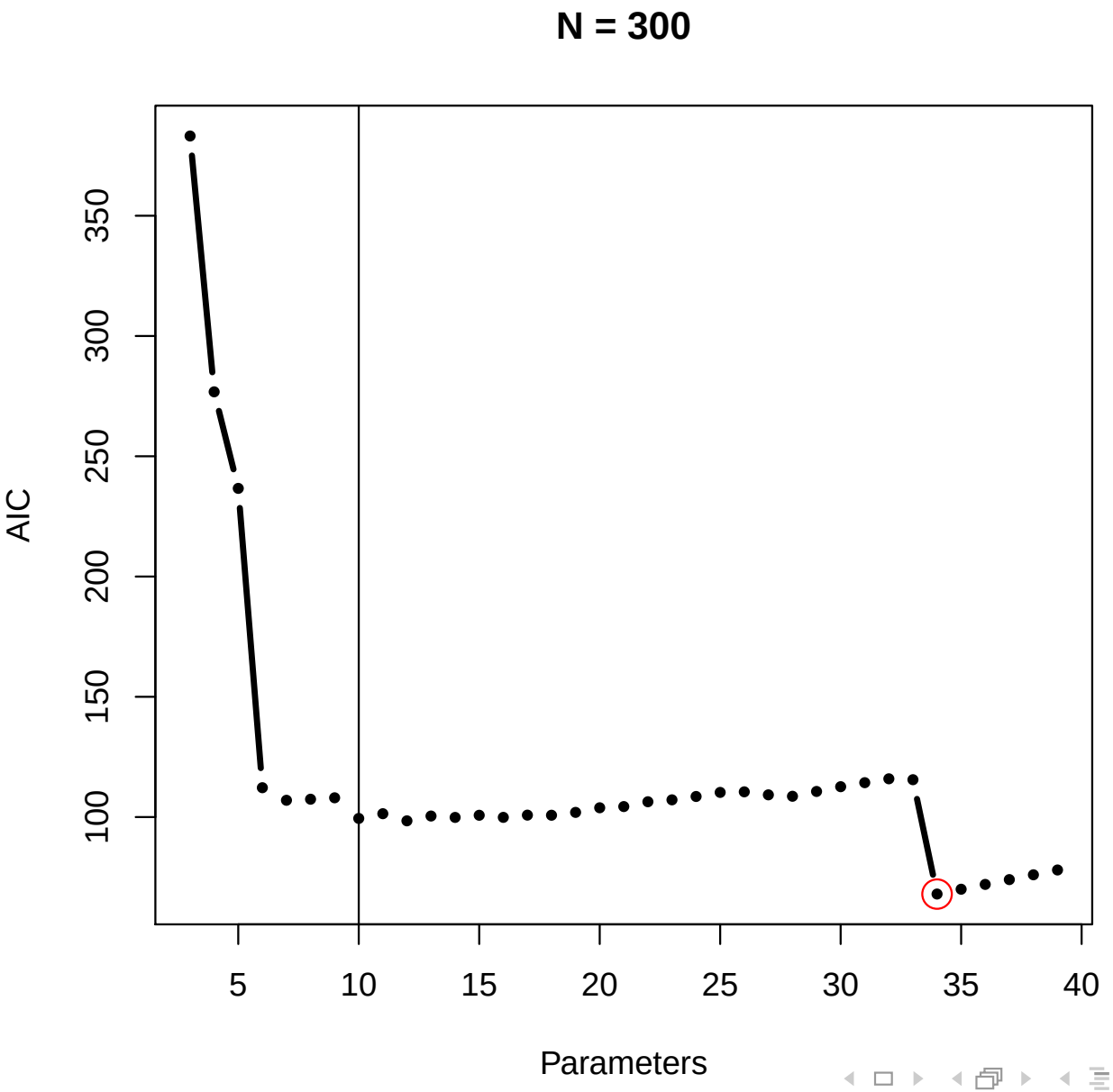
BIC or AIC?



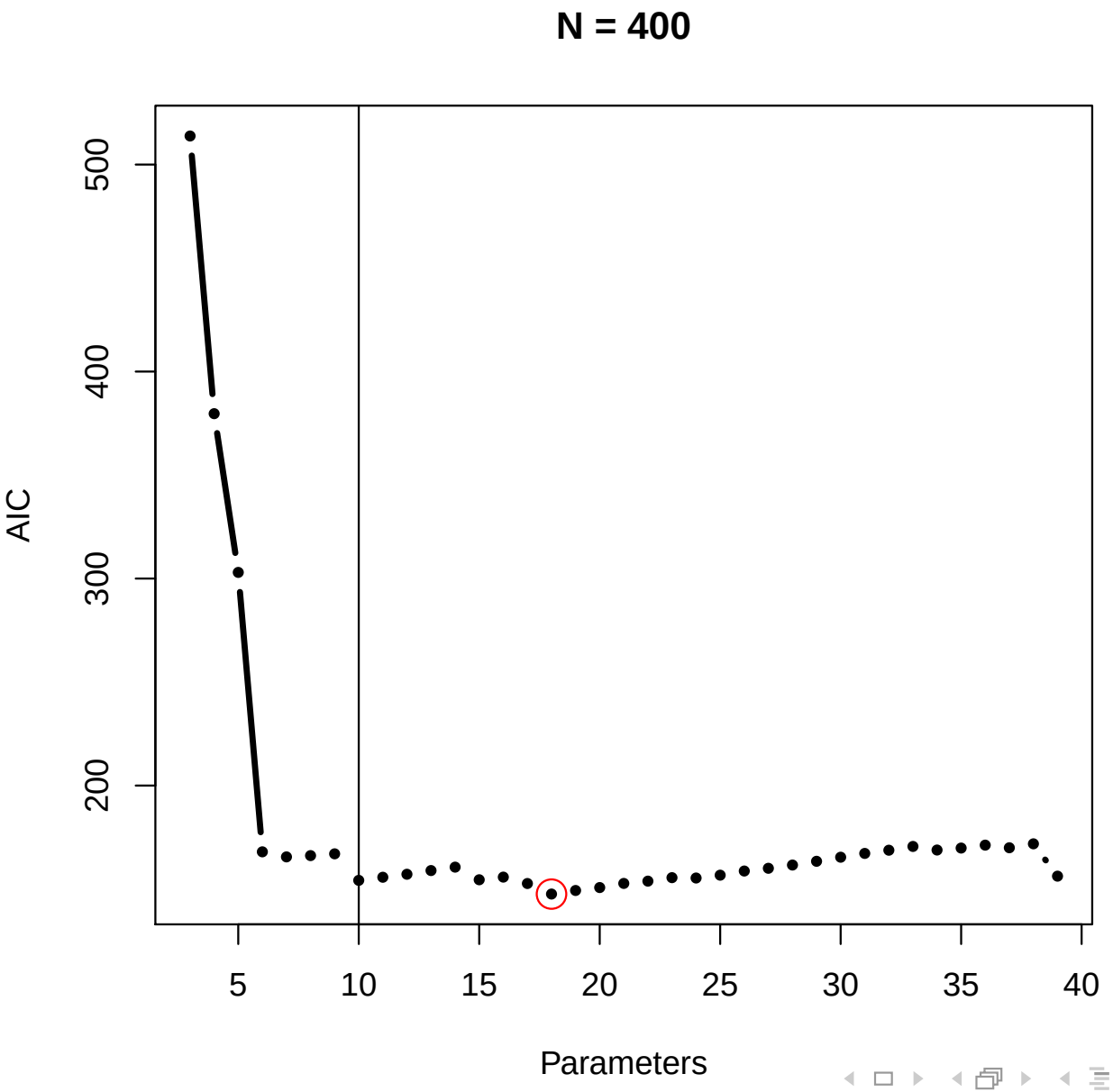
BIC or AIC?



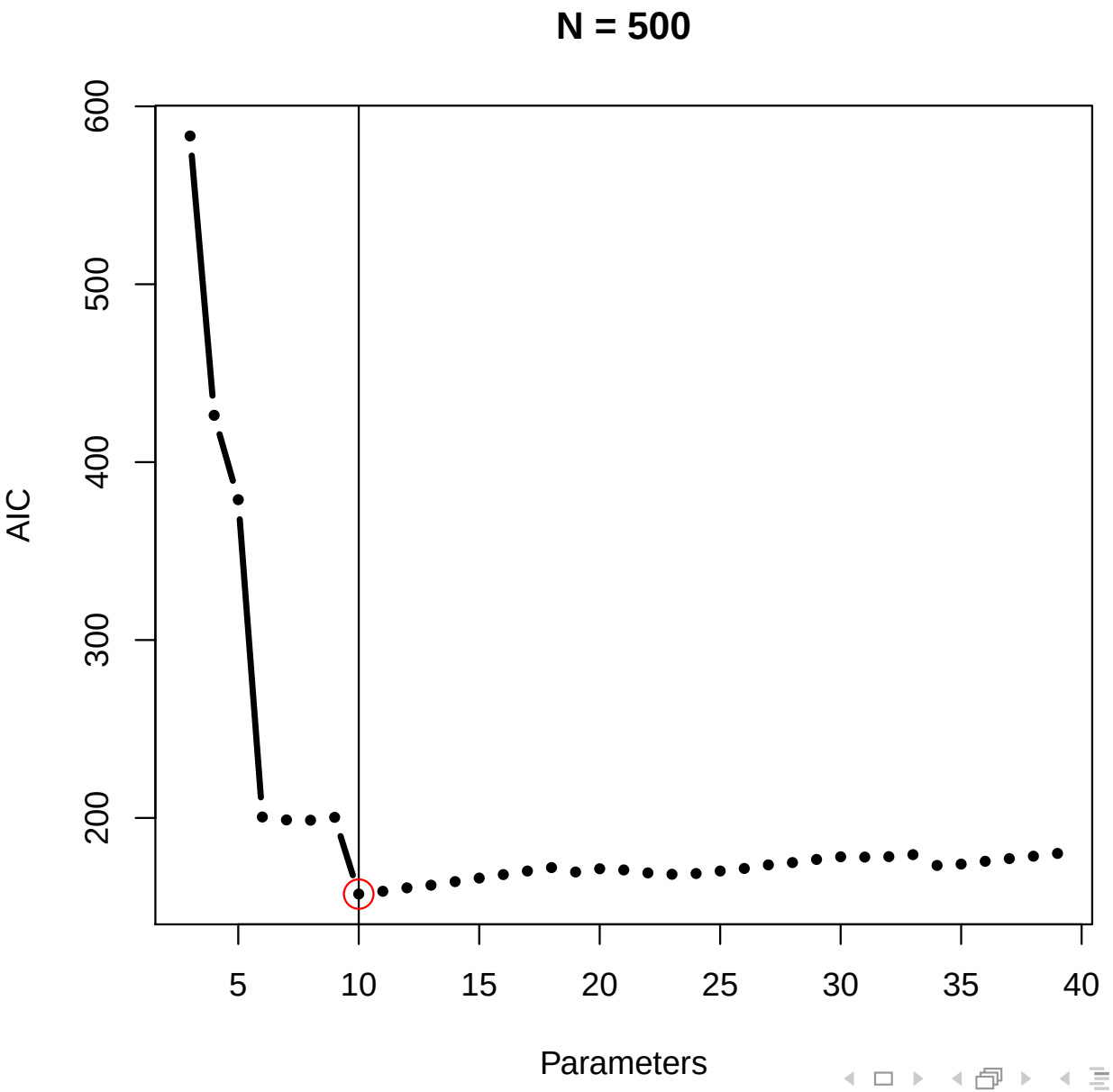
BIC or AIC?


N = 300
AIC
100
150
200
250
300
350
5
10
15
20
25
30
35
40
Parameters

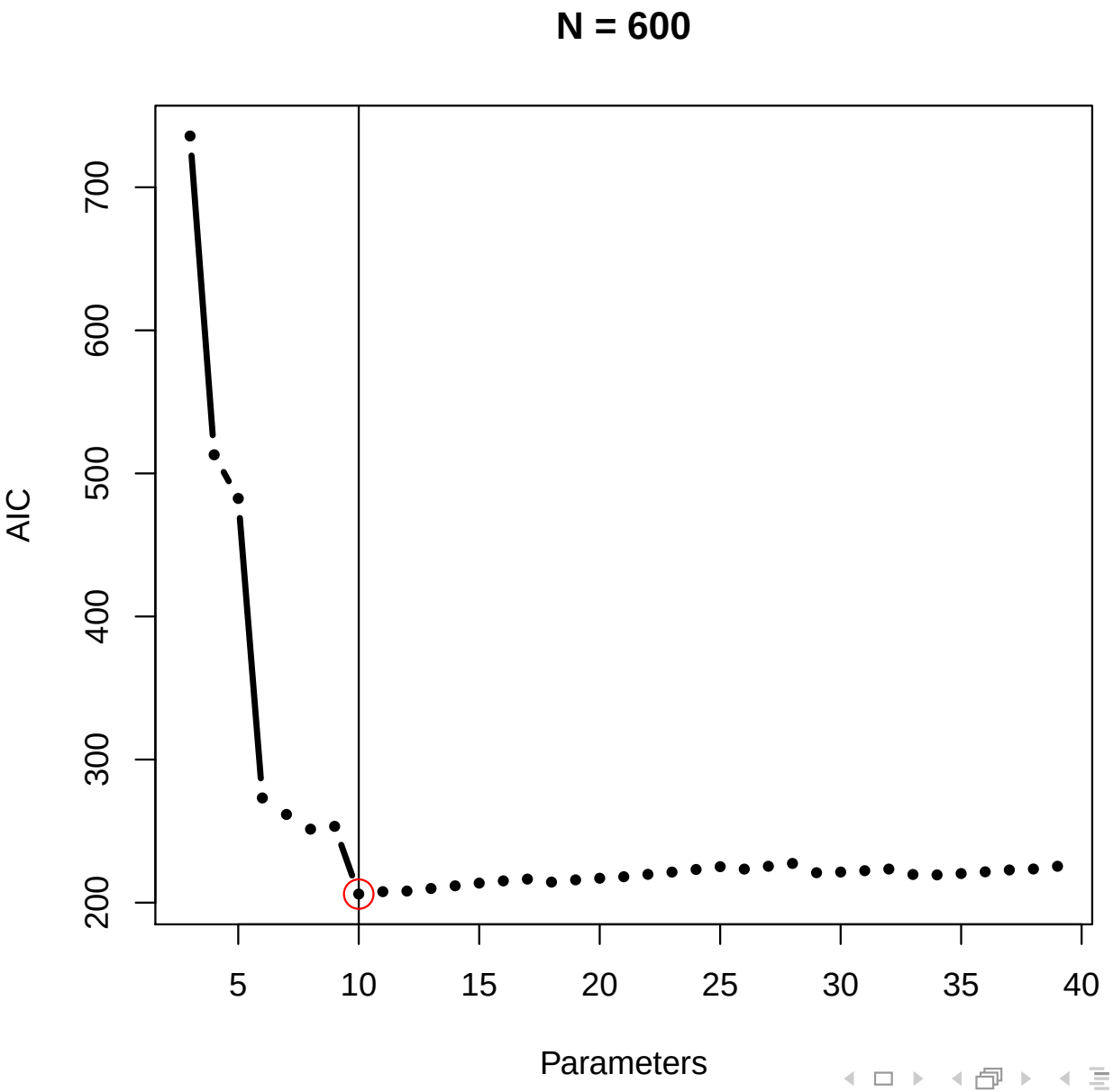
BIC or AIC?



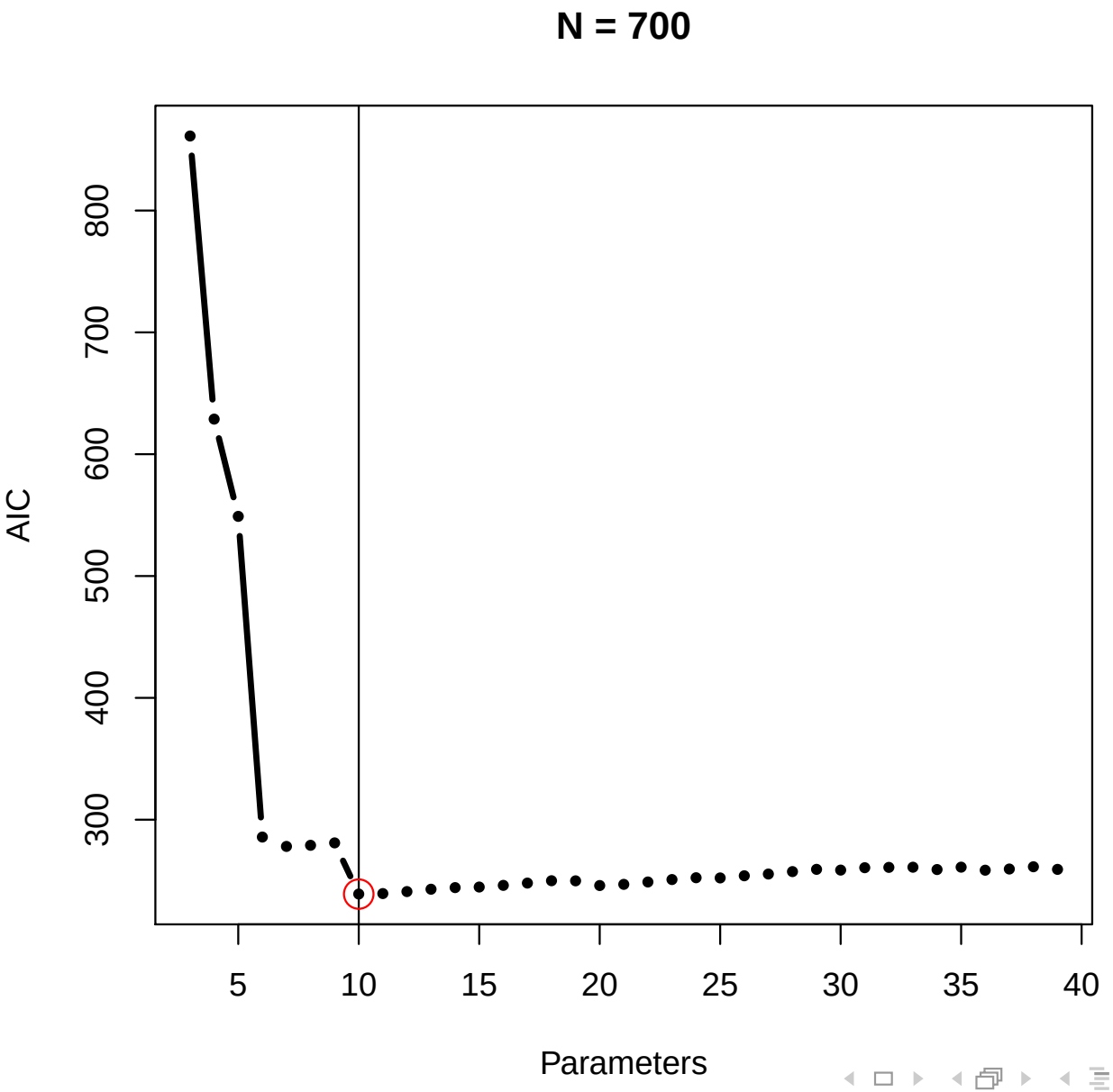
BIC or AIC?



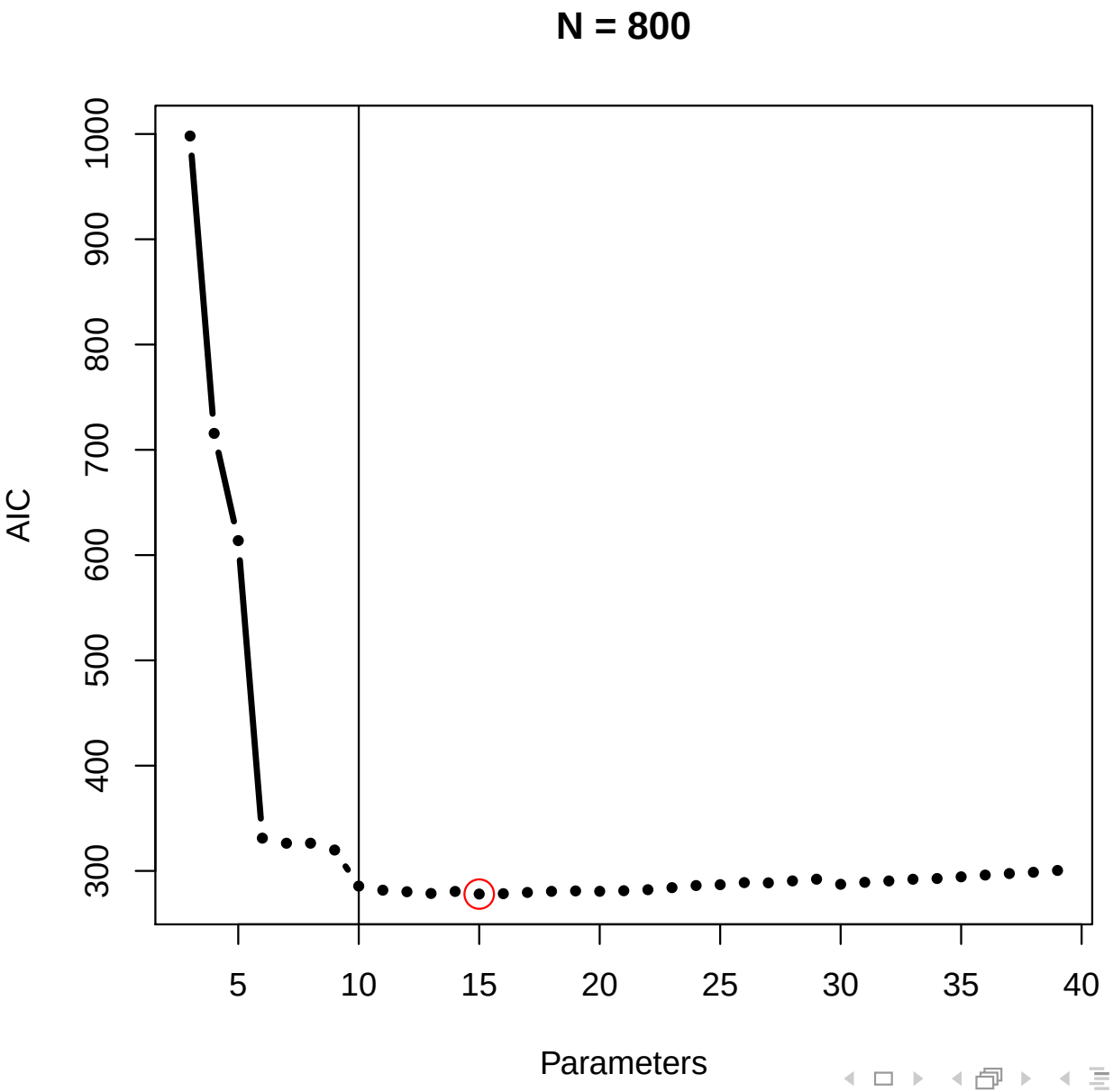
BIC or AIC?


N = 600
AIC
200
300
400
500
600
700
5
10
15
20
25
30
35
40
Parameters

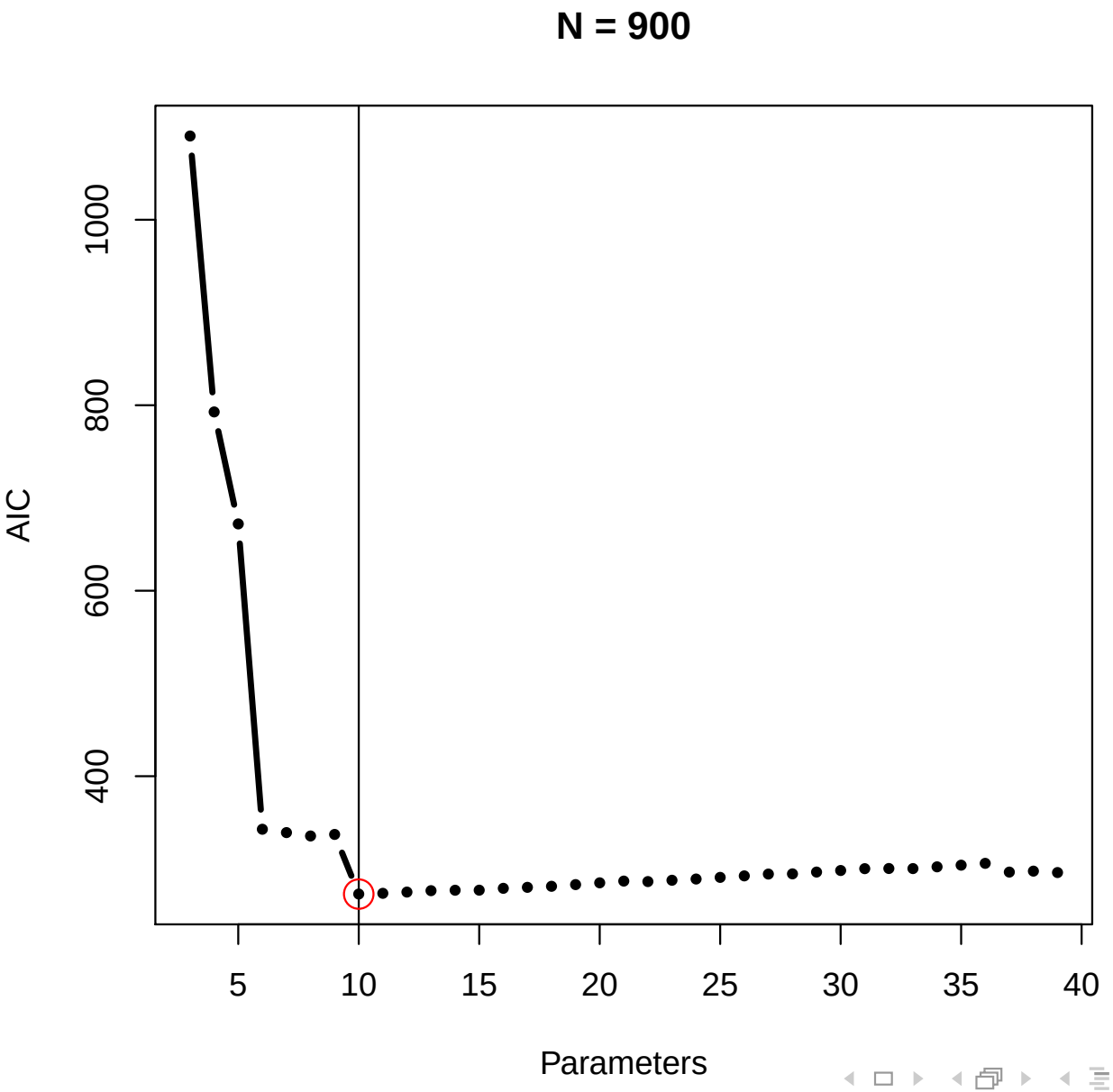
BIC or AIC?


N = 700
AIC
300
400
500
600
700
800
5
10
15
20
25
30
35
40
Parameters

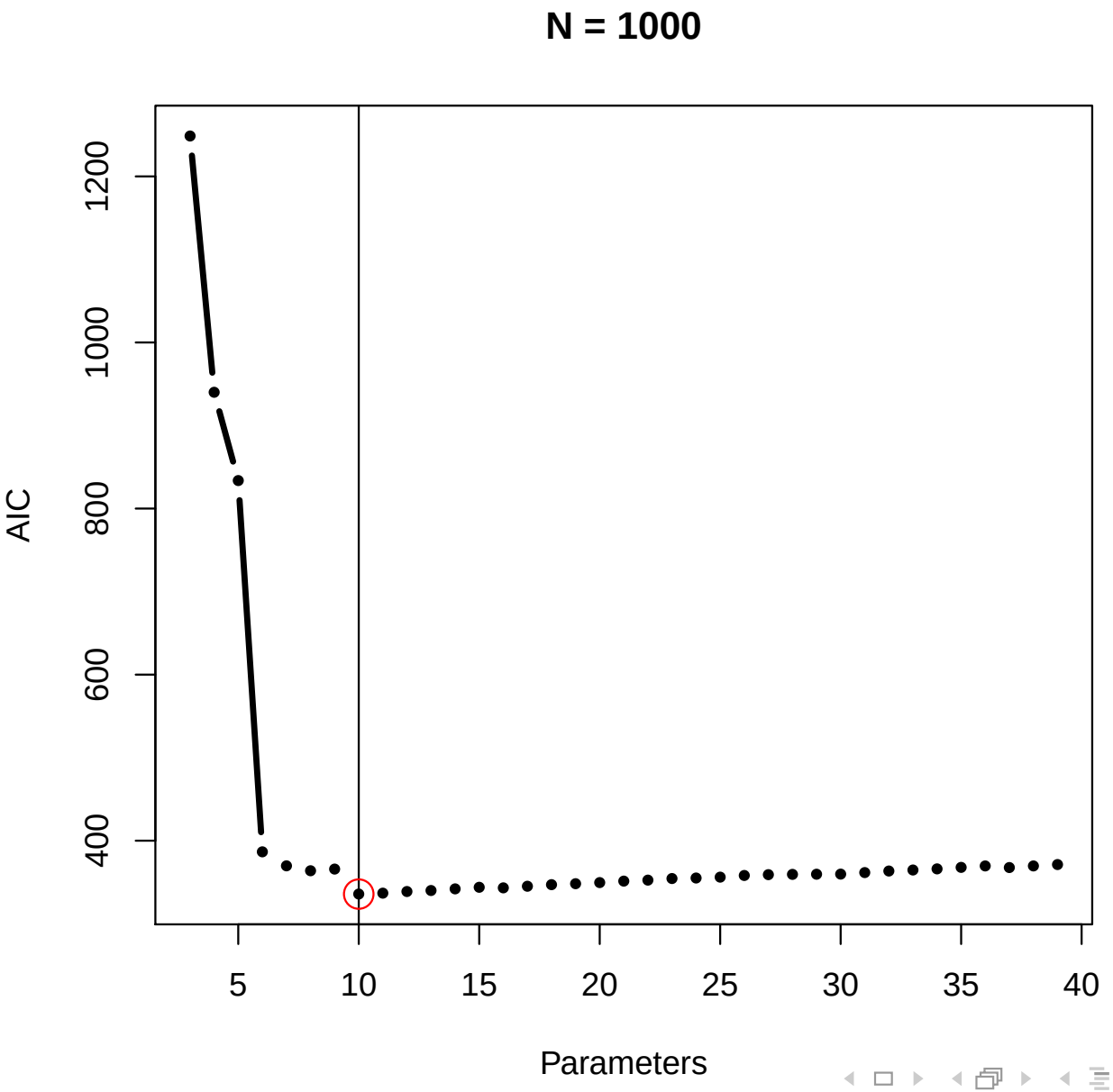
BIC or AIC?


N = 800
AIC
1000
900
800
700
600
500
400
300
5
10
15
20
25
30
35
40
Parameters

BIC or AIC?



BIC or AIC?


N = 1000
AIC
400
600
800
1000
1200
5
10
15
20
25
30
35
40
Parameters

BIC or AIC?

- BIC
  - Asymptotically consistent if true model is in choice set
  - As $N \to \infty$ will choose correct model with probability 1 (if available)
  - Small samples$\rightsquigarrow$ overpenalize
- AIC
  - No asymptotic guarantees $\rightsquigarrow$ derivation doesn't require truth in set. (KL-criteria)
  - In large samples$\rightsquigarrow$ favors complexity
  - Small samples$\rightsquigarrow$ avoids over penalization

How Do We Select A Model?

Analytic statistics for selection, include penalty for complexity

How Do We Select A Model?

Analytic statistics for selection, include penalty for complexity

- AIC : Akaka Information Criterion

How Do We Select A Model?

Analytic statistics for selection, include penalty for complexity

- AIC : Akaka Information Criterion
- BIC: Bayesian Information Criterion

How Do We Select A Model?

Analytic statistics for selection, include penalty for complexity

- AIC : Akaka Information Criterion
- BIC: Bayesian Information Criterion
- DIC: Deviance Information Criterion

How Do We Select A Model?

Analytic statistics for selection, include penalty for complexity

- AIC : Akaka Information Criterion
- BIC: Bayesian Information Criterion
- DIC: Deviance Information Criterion

Can work well, but...

How Do We Select A Model?

Analytic statistics for selection, include penalty for complexity

- AIC : Akaka Information Criterion
- BIC: Bayesian Information Criterion
- DIC: Deviance Information Criterion

Can work well, but...

- Rely on specific loss function

How Do We Select A Model?

Analytic statistics for selection, include penalty for complexity

- AIC : Akaka Information Criterion
- BIC: Bayesian Information Criterion
- DIC: Deviance Information Criterion

Can work well, but...

- Rely on specific loss function
- Rely on asymptotic argument

How Do We Select A Model?

Analytic statistics for selection, include penalty for complexity

- AIC : Akaka Information Criterion
- BIC: Bayesian Information Criterion
- DIC: Deviance Information Criterion

Can work well, but...

- Rely on specific loss function
- Rely on asymptotic argument
- Rely on estimate of number of parameters

How Do We Select A Model?

Analytic statistics for selection, include penalty for complexity

- AIC : Akaka Information Criterion
- BIC: Bayesian Information Criterion
- DIC: Deviance Information Criterion

Can work well, but...

- Rely on specific loss function
- Rely on asymptotic argument
- Rely on estimate of number of parameters
- Extremely model dependent

How Do We Select A Model?

Analytic statistics for selection, include penalty for complexity

- AIC : Akaka Information Criterion
- BIC: Bayesian Information Criterion
- DIC: Deviance Information Criterion

Can work well, but...

- Rely on specific loss function
- Rely on asymptotic argument
- Rely on estimate of number of parameters
- Extremely model dependent

Need: general tool for evaluating models, replicates decision problem

Cross-Validation: Some Intuition

Optimal division of data for prediction:

Cross-Validation: Some Intuition

Optimal division of data for prediction:

- Train: build model

Cross-Validation: Some Intuition

Optimal division of data for prediction:

- Train: build model
- Validation: assess model

Cross-Validation: Some Intuition

Optimal division of data for prediction:

- Train: build model
- Validation: assess model
- Test: predict remaining data

Cross-Validation: Some Intuition

Optimal division of data for prediction:

- Train: build model
- Validation: assess model
- Test: predict remaining data

K-fold Cross-validation idea: create many training and test sets.

Cross-Validation: Some Intuition

Optimal division of data for prediction:

- Train: build model
- Validation: assess model
- Test: predict remaining data

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets

Cross-Validation: Some Intuition

Optimal division of data for prediction:

- Train: build model
- Validation: assess model
- Test: predict remaining data

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets
- Each step: use held out data to evaluate performance

Cross-Validation: Some Intuition

Optimal division of data for prediction:

- Train: build model
- Validation: assess model
- Test: predict remaining data

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets
- Each step: use held out data to evaluate performance
- Avoid overfitting and have context specific penalty

Cross-Validation: Some Intuition

Optimal division of data for prediction:

- Train: build model
- Validation: assess model
- Test: predict remaining data

K-fold Cross-validation idea: create many training and test sets.

- Idea: use observations both in training and test sets
- Each step: use held out data to evaluate performance
- Avoid overfitting and have context specific penalty

Estimates:

$$\text{Error} = \mathrm{E}\left[\mathrm{E}[L(\boldsymbol{Y}, f(\hat{\boldsymbol{\beta}}, \boldsymbol{X}))|\mathcal{T}]\right]$$

Cross-Validation: A How To Guide

Process:

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, . . ., Group K)

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, . . ., Group K)
- Rotate through groups as follows

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, . . ., Group K)
- Rotate through groups as follows

Step	Training	Validation ("Test")

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.

  (Group 1, Group 2, Group3, ..., Group K)

- Rotate through groups as follows

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.

  (Group 1, Group 2, Group3, ..., Group K)

- Rotate through groups as follows

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.

  (Group 1, Group 2, Group3, ..., Group K)

- Rotate through groups as follows

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.

  (Group 1, Group 2, Group3, ..., Group K)

- Rotate through groups as follows

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮

Cross-Validation: A How To Guide

Process:

- Randomly partition data into K groups.
  (Group 1, Group 2, Group3, ..., Group K)
- Rotate through groups as follows

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K-1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{\beta}, \boldsymbol{X})$

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K-1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{\beta}, \boldsymbol{X})$
- Predict values for K^{th}

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K-1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{\beta}, \boldsymbol{X})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{\beta}, \boldsymbol{X}))$

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K-1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{\beta}, \boldsymbol{X})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{\beta}, \boldsymbol{X}))$
  - Mean square error, Absolute error, Prediction error, ...

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K-1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{\beta}, \boldsymbol{X})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{\beta}, \boldsymbol{X}))$
  - Mean square error, Absolute error, Prediction error, ...

CV(ind. classification) $= \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{Y}_i, f^{-k}(\boldsymbol{\beta}, \boldsymbol{X}_i))$

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K-1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{\beta}, \boldsymbol{X})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{\beta}, \boldsymbol{X}))$
  - Mean square error, Absolute error, Prediction error, ...

CV(ind. classification) $= \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{Y}_i, f^{-k}(\boldsymbol{\beta}, \boldsymbol{X}_i))$

CV(proportions) $=$
$\frac{1}{K}\sum_{j=1}^{K}$ Mean Square Error Proportions from Group j

Cross-Validation: A How To Guide

Step	Training	Validation ("Test")
1	Group2, Group3, Group 4, ..., Group K	Group 1
2	Group 1, Group3, Group 4, ..., Group K	Group 2
3	Group 1, Group 2, Group 4, ..., Group K	Group 3
⋮	⋮	⋮
K	Group 1, Group 2, Group 3, ..., Group K - 1	Group K

Strategy:

- Divide data into K groups
- Train data on $K-1$ groups. Estimate $\hat{f}^{-K}(\boldsymbol{\beta}, \boldsymbol{X})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{\beta}, \boldsymbol{X}))$
  - Mean square error, Absolute error, Prediction error, ...

  CV(ind. classification) $= \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{Y}_i, f^{-k}(\boldsymbol{\beta}, \boldsymbol{X}_i))$

  CV(proportions) $=$
  $\frac{1}{K}\sum_{j=1}^{K}$ Mean Square Error Proportions from Group j
- Final choice: model with highest CV score

How Do We Select K? (HTF, Section 7.10)

Common values of K

- $K = 5$: Five fold cross validation
- $K = 10$: Ten fold cross validation
- $K = N$: Leave one out cross validation

Considerations:

- How sensitive are inferences to number of coded documents? (HTF, pg 243-244)
- 200 labeled documents
  - $K = N \rightarrow 199$ documents to train,
  - $K = 10 \rightarrow 180$ documents to train
  - $K = 5 \rightarrow 160$ documents to train
- 50 labeled documents
  - $K = N \rightarrow 49$ documents to train,
  - $K = 10 \rightarrow 45$ documents to train
  - $K = 5 \rightarrow 40$ documents to train
- How long will it take to run models?
  - $K-$fold cross validation requires $K\times$ One model run
- What is the correct loss function?

If you cross validate, you really need to cross validate (Section 7.10.2, ESL)

- Use CV to estimate prediction error
- All supervised steps performed in cross-validation
- Underestimate prediction error
- Could lead to selecting lower performing model

Example from Facebook Data

What do people say to legislators? (Franco, Grimmer, and Lee 2017)

1) Example: estimating classification error
    a) Accuracy in legislator posts: 75%
    b) Accuracy in public posts: 66.25%

Credit Claiming (Back to Ridge/Lasso, Grimmer, Westwood, and Messing 2014)

```
library(glmnet)
set.seed(8675309) ##setting seed
folds<- sample(1:10, nrow(dtm), replace=T) ##assigning to fold
out_of_samp<- c() ##collecting the predictions
```

Credit Claiming (Back to Ridge/Lasso, Grimmer, Westwood, and Messing 2014)

```
for(z in 1:10){
train<- which(folds!=z) ##the observations we will use to train the model

test<- which(folds==z) ##the observations we will use to test the model
part1<- cv.glmnet(x = dtm[train,], y = credit[train], alpha = 1, family =
́binomial) ##fitting the LASSO model on the data.
## alpha = 1 -> LASSO
## alpha = 0 -> RIDGE
## 0<alpha<1 -> Elastic-Net
out_of_samp[test]<- predict(part1, newx= dtm[test,], s = part1$lambda.min,
type =́class) ##predicting the labels
print(z) ##printing the labels
}
conf_table<- table(out_of_samp, credit) ##calculating the confusion table
> round(sum(diag(conf_table))/len(credit), 3)
[1] 0.844
```

Generalized Cross Validation and Ridge Regression

In some special cases there are analytic solutions:

Generalized Cross Validation and Ridge Regression

In some special cases there are analytic solutions:

$$\beta^{\text{Ridge}} = \left(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_J\right)^{-1} \boldsymbol{X}'\boldsymbol{Y}$$

Generalized Cross Validation and Ridge Regression

In some special cases there are analytic solutions:

$$\begin{aligned} \beta^{\text{Ridge}} &= \left(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_J\right)^{-1} \boldsymbol{X}'\boldsymbol{Y} \\ \widehat{\boldsymbol{Y}} &= \boldsymbol{X}(\beta)^{\text{Ridge}} \end{aligned}$$

Generalized Cross Validation and Ridge Regression

In some special cases there are analytic solutions:

$$
\begin{aligned}
\boldsymbol{\beta}^{\text{Ridge}} &= \left(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_J\right)^{-1} \boldsymbol{X}'\boldsymbol{Y} \\
\widehat{\boldsymbol{Y}} &= \boldsymbol{X}(\boldsymbol{\beta})^{\text{Ridge}} \\
&= \underbrace{\boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_J\right)^{-1} \boldsymbol{X}'}_{\text{Hat Matrix}} \boldsymbol{Y}
\end{aligned}
$$

Generalized Cross Validation and Ridge Regression

In some special cases there are analytic solutions:

$$
\begin{aligned}
\beta^{\text{Ridge}} &= \left(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_J\right)^{-1} \boldsymbol{X}'\boldsymbol{Y} \\
\widehat{\boldsymbol{Y}} &= \boldsymbol{X}(\beta)^{\text{Ridge}} \\
&= \underbrace{\boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_J\right)^{-1} \boldsymbol{X}'}_{\text{Hat Matrix}} \boldsymbol{Y} \\
\widehat{\boldsymbol{Y}} &= \underbrace{\boldsymbol{H}}_{\text{Smoother Matrix}} \boldsymbol{Y}
\end{aligned}
$$

Generalized Cross Validation and Ridge Regression

In some special cases there are analytic solutions:

$$
\begin{aligned}
\beta^{\text{Ridge}} &= \left(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_J\right)^{-1} \boldsymbol{X}'\boldsymbol{Y} \\
\widehat{\boldsymbol{Y}} &= \boldsymbol{X}(\beta)^{\text{Ridge}} \\
&= \underbrace{\boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_J\right)^{-1} \boldsymbol{X}'}_{\text{Hat Matrix}} \boldsymbol{Y} \\
\widehat{\boldsymbol{Y}} &= \underbrace{\boldsymbol{H}}_{\text{Smoother Matrix}} \boldsymbol{Y}
\end{aligned}
$$

Generalized Cross Validation and Ridge Regression

Why do we care?

Generalized Cross Validation and Ridge Regression

Why do we care?
Leave one out cross validation

Generalized Cross Validation and Ridge Regression

Why do we care?
Leave one out cross validation

$$\text{Cross Validation}(1) \quad = \quad \frac{1}{N}\sum_{i=1}^{N}(Y_i - f(\boldsymbol{X}_{-i}, \boldsymbol{Y}_{-i}, \lambda, \hat{\beta}))^2$$

Generalized Cross Validation and Ridge Regression

Why do we care?
Leave one out cross validation

$$\begin{aligned}
\text{Cross Validation}(1) &= \frac{1}{N}\sum_{i=1}^{N}(Y_i - f(\boldsymbol{X}_{-i}, \boldsymbol{Y}_{-i}, \lambda, \hat{\boldsymbol{\beta}}))^2 \\
&= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{Y_i - f(\boldsymbol{X}, \boldsymbol{Y}, \lambda, \hat{\boldsymbol{\beta}})}{1 - H_{ii}}\right)^2
\end{aligned}$$

Generalized Cross Validation and Ridge Regression

Calculating $\boldsymbol{H}$ can be computationally expensive

Generalized Cross Validation and Ridge Regression

Calculating $\boldsymbol{H}$ can be computationally expensive

- $\text{Trace}(\boldsymbol{H}) \equiv \text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} H_{ii}$

Generalized Cross Validation and Ridge Regression

Calculating $\boldsymbol{H}$ can be computationally expensive

- $\text{Trace}(\boldsymbol{H}) \equiv \text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} H_{ii}$
- $\text{Tr}(\boldsymbol{H})$ = Effective number of parameters (class regression = number of independent variables + 1)

Generalized Cross Validation and Ridge Regression

Calculating $\boldsymbol{H}$ can be computationally expensive

- $\text{Trace}(\boldsymbol{H}) \equiv \text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} H_{ii}$
- $\text{Tr}(\boldsymbol{H})$ = Effective number of parameters (class regression = number of independent variables + 1)
- For Ridge regression:

Generalized Cross Validation and Ridge Regression

Calculating $\boldsymbol{H}$ can be computationally expensive

- $\text{Trace}(\boldsymbol{H}) \equiv \text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} H_{ii}$
- $\text{Tr}(\boldsymbol{H})$ = Effective number of parameters (class regression = number of independent variables + 1)
- For Ridge regression:

$$\text{Tr}(\boldsymbol{H}) \quad = \quad \sum_{j=1}^{J} \frac{\lambda_j}{\lambda_j + \underbrace{\lambda}_{\text{Penalty}}}$$

Generalized Cross Validation and Ridge Regression

Calculating $\boldsymbol{H}$ can be computationally expensive

- $\text{Trace}(\boldsymbol{H}) \equiv \text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} H_{ii}$
- $\text{Tr}(\boldsymbol{H})$ = Effective number of parameters (class regression = number of independent variables + 1)
- For Ridge regression:

$$\text{Tr}(\boldsymbol{H}) = \sum_{j=1}^{J} \frac{\lambda_j}{\lambda_j + \underbrace{\lambda}_{\text{Penalty}}}$$

where λ_j is the jth Eigenvalue from $\boldsymbol{\Sigma} = \boldsymbol{X}'\boldsymbol{X}$

Generalized Cross Validation and Ridge Regression

Calculating $\boldsymbol{H}$ can be computationally expensive

- $\text{Trace}(\boldsymbol{H}) \equiv \text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} H_{ii}$
- $\text{Tr}(\boldsymbol{H})$ = Effective number of parameters (class regression = number of independent variables + 1)
- For Ridge regression:

$$\text{Tr}(\boldsymbol{H}) \quad = \quad \sum_{j=1}^{J} \frac{\lambda_j}{\lambda_j + \underbrace{\lambda}_{\text{Penalty}}}$$

where λ_j is the jth Eigenvalue from $\boldsymbol{\Sigma} = \boldsymbol{X}'\boldsymbol{X}$ (!!!!!)

Generalized Cross Validation and Ridge Regression

Calculating $\boldsymbol{H}$ can be computationally expensive

- $\text{Trace}(\boldsymbol{H}) \equiv \text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} H_{ii}$
- $\text{Tr}(\boldsymbol{H})$ = Effective number of parameters (class regression = number of independent variables + 1)
- For Ridge regression:

$$\text{Tr}(\boldsymbol{H}) \quad = \quad \sum_{j=1}^{J} \frac{\lambda_j}{\lambda_j + \underbrace{\lambda}_{\text{Penalty}}}$$

where λ_j is the jth Eigenvalue from $\boldsymbol{\Sigma} = \boldsymbol{X}'\boldsymbol{X}$ (!!!!!)

Define generalized cross validation:

Generalized Cross Validation and Ridge Regression

Calculating $\boldsymbol{H}$ can be computationally expensive

- $\text{Trace}(\boldsymbol{H}) \equiv \text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} H_{ii}$
- $\text{Tr}(\boldsymbol{H})$ = Effective number of parameters (class regression = number of independent variables + 1)
- For Ridge regression:

$$\text{Tr}(\boldsymbol{H}) = \sum_{j=1}^{J} \frac{\lambda_j}{\lambda_j + \underbrace{\lambda}_{\text{Penalty}}}$$

where λ_j is the jth Eigenvalue from $\boldsymbol{\Sigma} = \boldsymbol{X}'\boldsymbol{X}$ (!!!!!)

Define generalized cross validation:

$$\text{GCV} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{Y_i - \hat{Y}_i}{1 - \frac{\text{Tr}(\boldsymbol{H})}{N}} \right)^2$$

Generalized Cross Validation and Ridge Regression

Calculating $\boldsymbol{H}$ can be computationally expensive

- $\text{Trace}(\boldsymbol{H}) \equiv \text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} H_{ii}$
- $\text{Tr}(\boldsymbol{H})$ = Effective number of parameters (class regression = number of independent variables + 1)
- For Ridge regression:

$$\text{Tr}(\boldsymbol{H}) = \sum_{j=1}^{J} \frac{\lambda_j}{\lambda_j + \underbrace{\lambda}_{\text{Penalty}}}$$

where λ_j is the jth Eigenvalue from $\boldsymbol{\Sigma} = \boldsymbol{X}'\boldsymbol{X}$ (!!!!!)

Define generalized cross validation:

$$\text{GCV} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{Y_i - \hat{Y}_i}{1 - \frac{\text{Tr}(\boldsymbol{H})}{N}} \right)^2$$

Applicable in any setting where we can write Smoother matrix

Generalized Cross Validation and Ridge Regression

Calculating $\boldsymbol{H}$ can be computationally expensive

- $\text{Trace}(\boldsymbol{H}) \equiv \text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} H_{ii}$
- $\text{Tr}(\boldsymbol{H})$ = Effective number of parameters (class regression = number of independent variables + 1)
- For Ridge regression:

$$\text{Tr}(\boldsymbol{H}) = \sum_{j=1}^{J} \frac{\lambda_j}{\lambda_j + \underbrace{\lambda}_{\text{Penalty}}}$$

where λ_j is the jth Eigenvalue from $\boldsymbol{\Sigma} = \boldsymbol{X}'\boldsymbol{X}$ (!!!!!)

Define generalized cross validation:

$$\text{GCV} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{Y_i - \hat{Y}_i}{1 - \frac{\text{Tr}(\boldsymbol{H})}{N}} \right)^2$$

Applicable in any setting where we can write Smoother matrix

Ensemble Learning: Intuition

Heuristic (upon which we'll improve):

Ensemble Learning: Intuition

Heuristic (upon which we'll improve): if regressions are accurate and diverse$\rightarrow$ ensemble methods improve

Ensemble Learning: Intuition

Heuristic (upon which we'll improve): if regressions are accurate and diverse$\rightarrow$ ensemble methods improve

Intuition:

Ensemble Learning: Intuition

Heuristic (upon which we'll improve): if regressions are accurate and diverse$\rightarrow$ ensemble methods improve

Intuition:

- Classify observations into two categories (Category 1, Category 2).

Ensemble Learning: Intuition

Heuristic (upon which we'll improve): if regressions are accurate and diverse$\rightarrow$ ensemble methods improve

Intuition:

- Classify observations into two categories (Category 1, Category 2).
- True labels: evenly distributed across two categories

Ensemble Learning: Intuition

Heuristic (upon which we'll improve): if regressions are accurate and diverse$\rightarrow$ ensemble methods improve

Intuition:

- Classify observations into two categories (Category 1, Category 2).
- True labels: evenly distributed across two categories
- Three classifiers with 75% accuracy, but independent

Ensemble Learning: Intuition

Heuristic (upon which we'll improve): if regressions are accurate and diverse→ ensemble methods improve

Intuition:

- Classify observations into two categories (Category 1, Category 2).
- True labels: evenly distributed across two categories
- Three classifiers with 75% accuracy, but independent
- Implement majority voting rule

Ensemble Learning: Intuition

Heuristic (upon which we'll improve): if regressions are accurate and diverse$\rightarrow$ ensemble methods improve

Intuition:

- Classify observations into two categories (Category 1, Category 2).
- True labels: evenly distributed across two categories
- Three classifiers with 75% accuracy, but independent
- Implement majority voting rule

$$\Pr(\text{Correct Guess}|\text{Votes})$$

Ensemble Learning: Intuition

Heuristic (upon which we'll improve): if regressions are accurate and diverse$\rightarrow$ ensemble methods improve

Intuition:

- Classify observations into two categories (Category 1, Category 2).
- True labels: evenly distributed across two categories
- Three classifiers with 75% accuracy, but independent
- Implement majority voting rule

$$\Pr(\text{Correct Guess}|\text{Votes}) = \Pr(3 \text{ correct}) + \Pr(2 \text{ correct})$$

Ensemble Learning: Intuition

Heuristic (upon which we'll improve): if regressions are accurate and diverse$\rightarrow$ ensemble methods improve

Intuition:

- Classify observations into two categories (Category 1, Category 2).
- True labels: evenly distributed across two categories
- Three classifiers with 75% accuracy, but independent
- Implement majority voting rule

$$\begin{aligned} \Pr(\text{Correct Guess}|\text{Votes}) &= \Pr(3 \text{ correct}) + \Pr(2 \text{ correct}) \\ &= 0.75^3 + 3 \times (0.75^2 \times 0.25) \end{aligned}$$

Ensemble Learning: Intuition

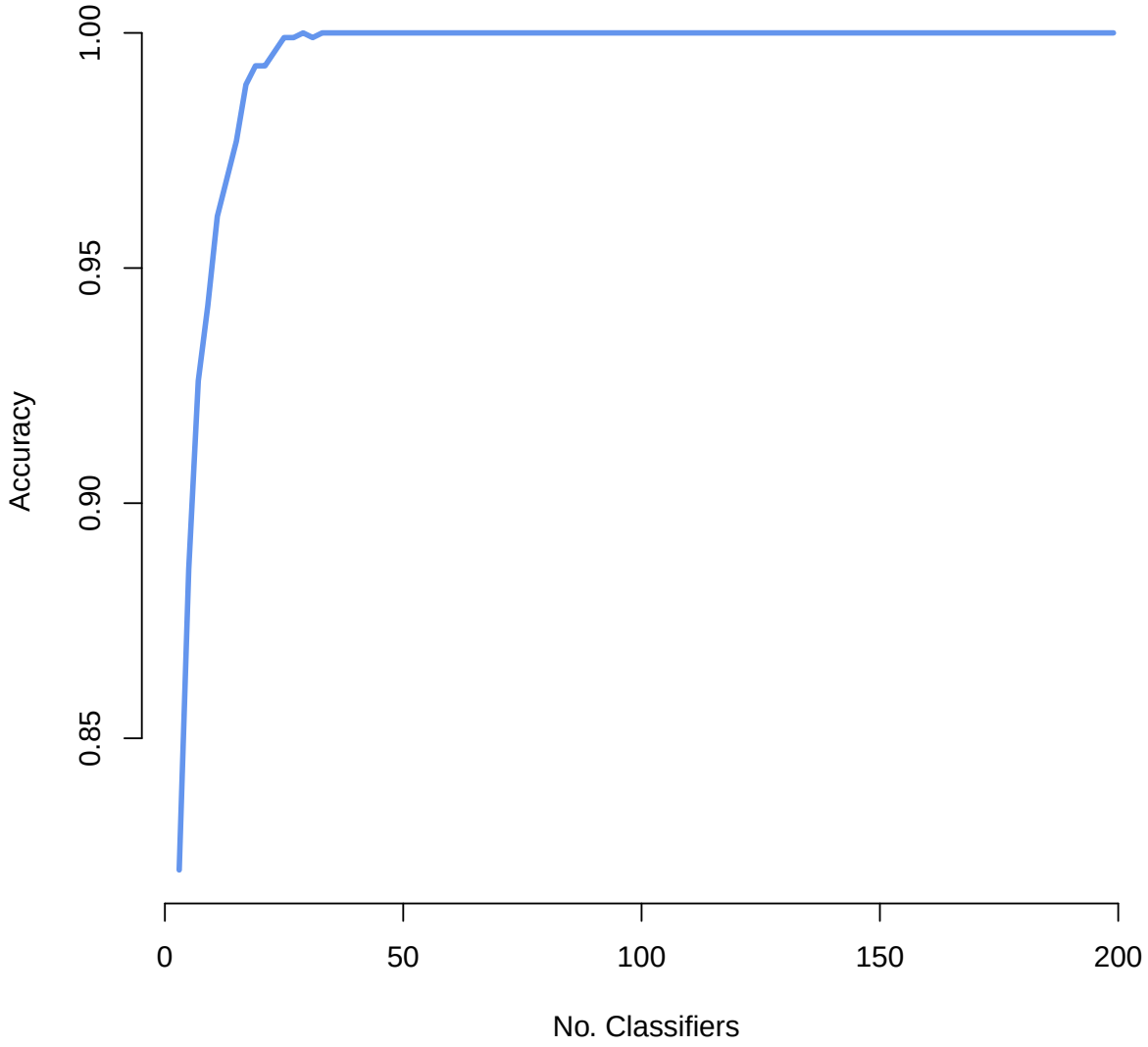
Heuristic (upon which we'll improve): if regressions are accurate and diverse$\rightarrow$ ensemble methods improve

Intuition:

- Classify observations into two categories (Category 1, Category 2).
- True labels: evenly distributed across two categories
- Three classifiers with 75% accuracy, but independent
- Implement majority voting rule

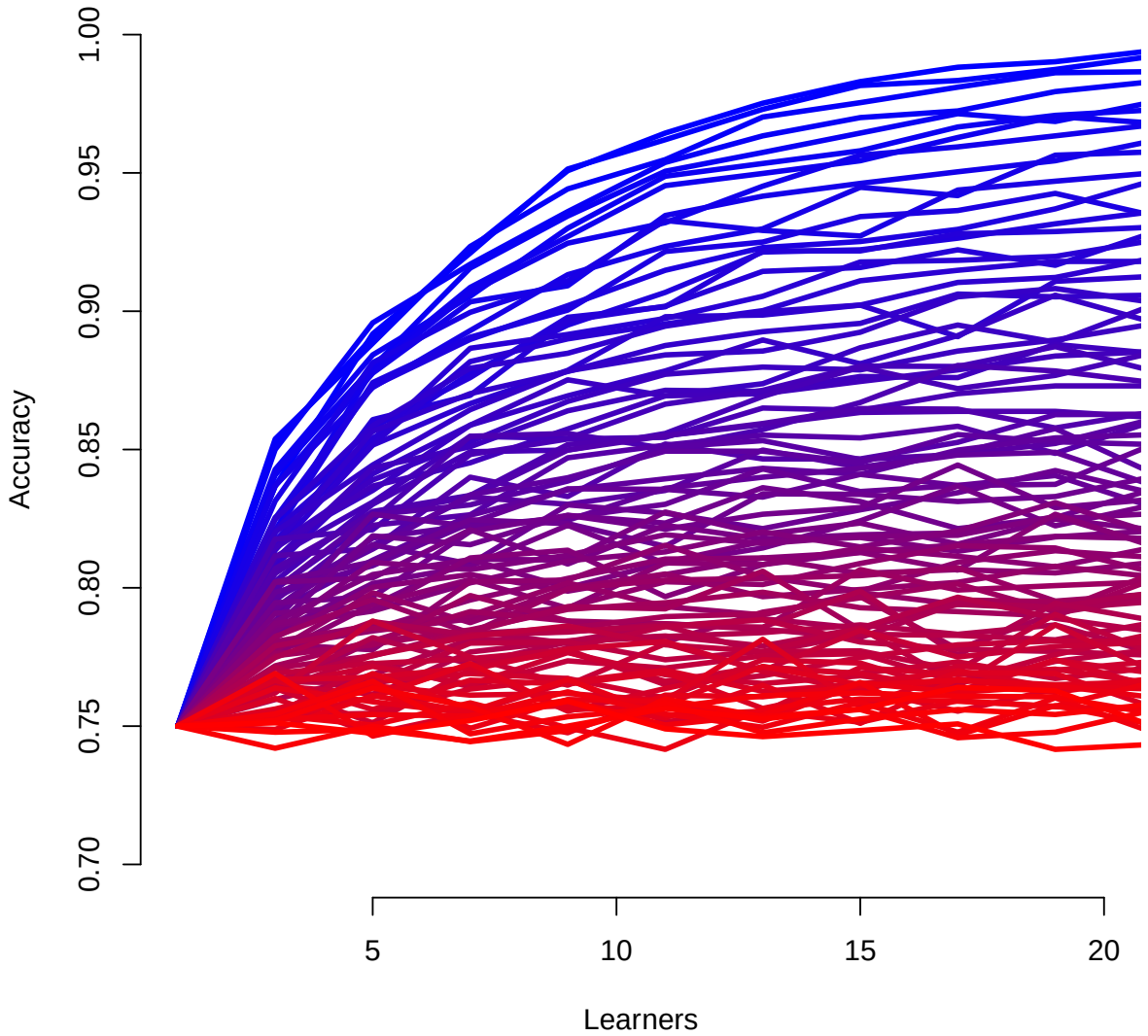
$$\begin{aligned}
\Pr(\text{Correct Guess}|\text{Votes}) &= \Pr(3 \text{ correct}) + \Pr(2 \text{ correct}) \\
&= 0.75^3 + 3 \times (0.75^2 \times 0.25) \\
&= 0.844
\end{aligned}$$

Ensemble Learning: Intuition



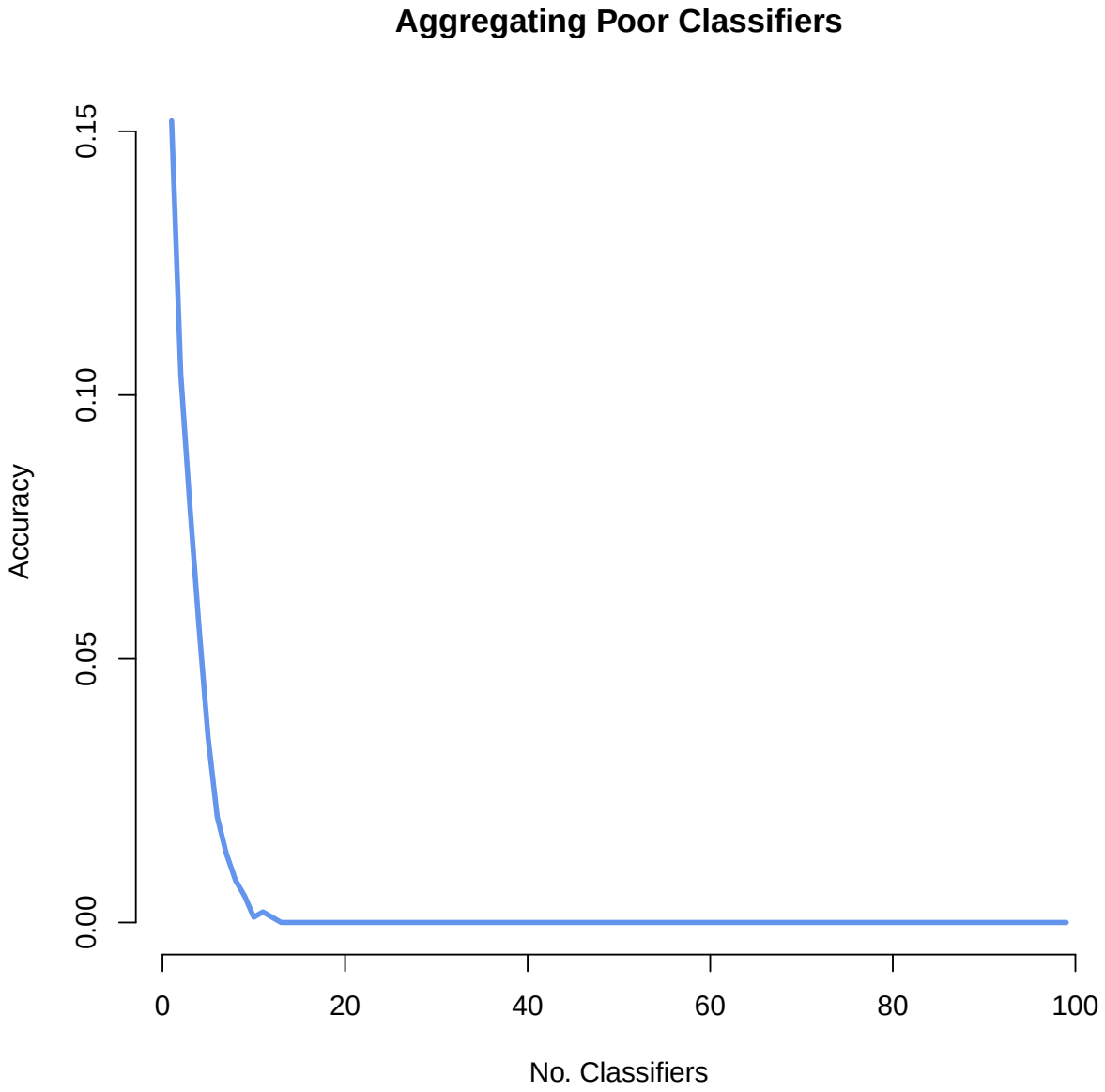
Ensemble Learning: Intuition

Diverse and Accurate matter.



Ensemble Learning: Intuition

Diverse and Accurate matter.



Wisdom of the Crowds:

Goal: estimate an observation's category $\rightsquigarrow$ $Y \in \{0, 1\}$

Wisdom of the Crowds:

Goal: estimate an observation's category $\rightsquigarrow$ $Y \in \{0, 1\}$
Classifiers: (suppose) a sequence of identically distributed (not necessarily independent) random variables.

Wisdom of the Crowds:

Goal: estimate an observation's category $\rightsquigarrow$ $Y \in \{0, 1\}$
Classifiers: (suppose) a sequence of identically distributed (not necessarily independent) random variables.
Suppose $Y = 1$

Wisdom of the Crowds:

Goal: estimate an observation's category $\rightsquigarrow$ $Y \in \{0, 1\}$
Classifiers: (suppose) a sequence of identically distributed (not necessarily independent) random variables.
Suppose $Y = 1$
Guess from classifer m is B_m with $\Pr(B_i = 1) = p > 0.5$.

Wisdom of the Crowds:

Goal: estimate an observation's category $\rightsquigarrow$ $Y \in \{0, 1\}$
Classifiers: (suppose) a sequence of identically distributed (not necessarily independent) random variables.
Suppose $Y = 1$
Guess from classifer m is B_m with $\Pr(B_i = 1) = p > 0.5$.

$$\bar{B} = \sum_{m=1}^{M} \frac{B_m}{M}$$

Wisdom of the Crowds:

Goal: estimate an observation's category $\rightsquigarrow$ $Y \in \{0, 1\}$
Classifiers: (suppose) a sequence of identically distributed (not necessarily independent) random variables.
Suppose $Y = 1$
Guess from classifer m is B_m with $\Pr(B_i = 1) = p > 0.5$.

$$\bar{B} = \sum_{m=1}^{M} \frac{B_m}{M}$$

Wisdom of crows (Condorcet Jury Theorem)

Wisdom of the Crowds:

Goal: estimate an observation's category $\rightsquigarrow$ $Y \in \{0, 1\}$
Classifiers: (suppose) a sequence of identically distributed (not necessarily independent) random variables.
Suppose $Y = 1$
Guess from classifer m is B_m with $\Pr(B_i = 1) = p > 0.5$.

$$\bar{B} = \sum_{m=1}^{M} \frac{B_m}{M}$$

Wisdom of crows (Condorcet Jury Theorem)

$$\lim_{M \to \infty} P(\bar{B} > 0.5) = 1$$

Wisdom of the Crowds

Suppose B_m have variance σ^2 and pairwise correlation ρ.

Wisdom of the Crowds

Suppose B_m have variance σ^2 and pairwise correlation ρ.
Then,

Wisdom of the Crowds

Suppose B_m have variance σ^2 and pairwise correlation ρ. Then,

$$\text{var}(\bar{B})$$

Wisdom of the Crowds

Suppose B_m have variance σ^2 and pairwise correlation ρ. Then,

$$\text{var}(\bar{B}) = \text{var}\left(\sum_{i=1}^{M} \frac{B_i}{M}\right)$$

Wisdom of the Crowds

Suppose B_m have variance σ^2 and pairwise correlation ρ. Then,

$$\begin{aligned}
\text{var}(\bar{B}) &= \text{var}\left(\sum_{i=1}^{M} \frac{B_i}{M}\right) \\
&= \frac{1}{M^2}\sum_{i=1}^{M} \text{var}\left(B_i\right) + \frac{2}{M^2}\sum_{i<j} \text{cov}(B_i, B_j)
\end{aligned}$$

Wisdom of the Crowds

Suppose B_m have variance σ^2 and pairwise correlation ρ. Then,

$$\begin{aligned}
\text{var}(\bar{B}) &= \text{var}\left(\sum_{i=1}^{M} \frac{B_i}{M}\right) \\
&= \frac{1}{M^2}\sum_{i=1}^{M} \text{var}\left(B_i\right) + \frac{2}{M^2}\sum_{i<j} \text{cov}(B_i, B_j) \\
&= \frac{M\sigma^2}{M^2} + \frac{2}{M^2}\rho\sigma^2\binom{M}{2}
\end{aligned}$$

Wisdom of the Crowds

Suppose B_m have variance σ^2 and pairwise correlation ρ. Then,

$$
\begin{aligned}
\text{var}(\bar{B}) &= \text{var}\left(\sum_{i=1}^{M} \frac{B_i}{M}\right) \\
&= \frac{1}{M^2}\sum_{i=1}^{M} \text{var}\left(B_i\right) + \frac{2}{M^2}\sum_{i<j} \text{cov}(B_i, B_j) \\
&= \frac{M\sigma^2}{M^2} + \frac{2}{M^2}\rho\sigma^2\binom{M}{2} \\
&= \underbrace{\rho\sigma^2}_{\text{Resolve with independence}} + \underbrace{\frac{1-\rho}{M}\sigma^2}_{\text{Resolve with }\uparrow\text{classifiers}}
\end{aligned}
$$

Bagging: bootstrap aggregation

Creating Weak Classifiers with resampling:

Bagging: bootstrap aggregation

Creating Weak Classifiers with resampling:

- Suppose we have dependent variables $\boldsymbol{Y}$ and data $\boldsymbol{X}$

Bagging: bootstrap aggregation

Creating Weak Classifiers with resampling:

- Suppose we have dependent variables $\boldsymbol{Y}$ and data $\boldsymbol{X}$
- For each bootstrap step m, $(m = 1, 2, \ldots, M)$ draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m$, $\tilde{\boldsymbol{X}}_m$.

Bagging: bootstrap aggregation

Creating Weak Classifiers with resampling:

- Suppose we have dependent variables $\boldsymbol{Y}$ and data $\boldsymbol{X}$
- For each bootstrap step m, $(m = 1, 2, \ldots, M)$ draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m$, $\tilde{\boldsymbol{X}}_m$.
- Train classifier on bootstrapped data,

Bagging: bootstrap aggregation

Creating Weak Classifiers with resampling:

- Suppose we have dependent variables $\boldsymbol{Y}$ and data $\boldsymbol{X}$
- For each bootstrap step m, $(m = 1, 2, \ldots, M)$ draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m$, $\tilde{\boldsymbol{X}}_m$.
- Train classifier on bootstrapped data,

$$\tilde{\boldsymbol{Y}}_m = f^m(\tilde{\boldsymbol{X}}_m, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda})$$

Bagging: bootstrap aggregation

Creating Weak Classifiers with resampling:

- Suppose we have dependent variables $\boldsymbol{Y}$ and data $\boldsymbol{X}$
- For each bootstrap step m, $(m = 1, 2, \ldots, M)$ draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m$, $\tilde{\boldsymbol{X}}_m$.
- Train classifier on bootstrapped data,

$$\begin{aligned} \tilde{\boldsymbol{Y}}_m &= f^m(\tilde{\boldsymbol{X}}_m, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda}) \\ \hat{f}^m(\boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda}) &= \text{Classifier from m}^{\text{th}} \text{ iteration at } \boldsymbol{x}_i \end{aligned}$$

Bagging: bootstrap aggregation

Creating Weak Classifiers with resampling:

- Suppose we have dependent variables $\boldsymbol{Y}$ and data $\boldsymbol{X}$
- For each bootstrap step m, $(m = 1, 2, \ldots, M)$ draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m$, $\tilde{\boldsymbol{X}}_m$.
- Train classifier on bootstrapped data,

$$\begin{aligned} \tilde{\boldsymbol{Y}}_m &= f^m(\tilde{\boldsymbol{X}}_m, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda}) \\ \hat{f}^m(\boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda}) &= \text{Classifier from m}^{\text{th}} \text{ iteration at } \boldsymbol{x}_i \end{aligned}$$

- Aggregating across classifiers,

Bagging: bootstrap aggregation

Creating Weak Classifiers with resampling:

- Suppose we have dependent variables $\boldsymbol{Y}$ and data $\boldsymbol{X}$
- For each bootstrap step m, $(m = 1, 2, \ldots, M)$ draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m$, $\tilde{\boldsymbol{X}}_m$.
- Train classifier on bootstrapped data,

$$\begin{aligned} \tilde{\boldsymbol{Y}}_m &= f^m(\tilde{\boldsymbol{X}}_m, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda}) \\ \hat{f}^m(\boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda}) &= \text{Classifier from m}^{\text{th}} \text{ iteration at } \boldsymbol{x}_i \end{aligned}$$

- Aggregating across classifiers,

$$f_{\text{bag}}(\boldsymbol{x}_i) = \frac{1}{M} \sum_{m=1}^{M} \hat{f}^m(\boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda})$$

Bagging: bootstrap aggregation

Creating Weak Classifiers with resampling:

- Suppose we have dependent variables $\boldsymbol{Y}$ and data $\boldsymbol{X}$
- For each bootstrap step m, $(m = 1, 2, \ldots, M)$ draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m$, $\tilde{\boldsymbol{X}}_m$.
- Train classifier on bootstrapped data,

$$\begin{aligned} \tilde{\boldsymbol{Y}}_m &= f^m(\tilde{\boldsymbol{X}}_m, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda}) \\ \hat{f}^m(\boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda}) &= \text{Classifier from m}^{\text{th}} \text{ iteration at } \boldsymbol{x}_i \end{aligned}$$

- Aggregating across classifiers,

$$f_{\text{bag}}(\boldsymbol{x}_i) = \frac{1}{M} \sum_{m=1}^{M} \hat{f}^m(\boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda})$$

- Only leads to a difference in estimate if classifiers are non-linear.

Bagging: bootstrap aggregation

Creating Weak Classifiers with resampling:

- Suppose we have dependent variables $\boldsymbol{Y}$ and data $\boldsymbol{X}$
- For each bootstrap step m, $(m = 1, 2, \ldots, M)$ draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m$, $\tilde{\boldsymbol{X}}_m$.
- Train classifier on bootstrapped data,

$$\begin{aligned} \tilde{\boldsymbol{Y}}_m &= f^m(\tilde{\boldsymbol{X}}_m, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda}) \\ \hat{f}^m(\boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda}) &= \text{Classifier from m}^{\text{th}} \text{ iteration at } \boldsymbol{x}_i \end{aligned}$$

- Aggregating across classifiers,

$$f_{\text{bag}}(\boldsymbol{x}_i) = \frac{1}{M} \sum_{m=1}^{M} \hat{f}^m(\boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}, \boldsymbol{\lambda})$$

- Only leads to a difference in estimate if classifiers are non-linear.
- Strong Correlation between classifiers (recall optimal division from previous slide)

Classification and Regression Trees (CART): Intuition

Consider regression $E[Y|\boldsymbol{x}_i]$.

Classification and Regression Trees (CART): Intuition

Consider regression $E[Y|\boldsymbol{x}_i]$.

With no assumptions, stratify $\rightsquigarrow$ different mean for unique values of $\boldsymbol{x}_i$

Classification and Regression Trees (CART): Intuition

Consider regression $E[Y|\boldsymbol{x}_i]$.

With no assumptions, stratify $\rightsquigarrow$ different mean for unique values of $\boldsymbol{x}_i$

- Within each strata p, compute average Y

Classification and Regression Trees (CART): Intuition

Consider regression $E[Y|\boldsymbol{x}_i]$.

With no assumptions, stratify $\rightsquigarrow$ different mean for unique values of $\boldsymbol{x}_i$

- Within each strata p, compute average Y

$$\bar{Y}|\boldsymbol{x}_p = \sum_{i=1}^{N} \frac{I(\boldsymbol{x}_i = \boldsymbol{x}_p)Y_i}{\sum_{t=1}^{N} I(\boldsymbol{x}_t = \boldsymbol{x}_p)}$$

Classification and Regression Trees (CART): Intuition

Consider regression $E[Y|\boldsymbol{x}_i]$.

With no assumptions, stratify $\rightsquigarrow$ different mean for unique values of $\boldsymbol{x}_i$

- Within each strata p, compute average Y

$$\bar{Y}|\boldsymbol{x}_p = \sum_{i=1}^{N} \frac{I(\boldsymbol{x}_i = \boldsymbol{x}_p)Y_i}{\sum_{t=1}^{N} I(\boldsymbol{x}_t = \boldsymbol{x}_p)}$$

Implies that for test data we would fit:

Classification and Regression Trees (CART): Intuition

Consider regression $E[Y|\boldsymbol{x}_i]$.

With no assumptions, stratify $\rightsquigarrow$ different mean for unique values of $\boldsymbol{x}_i$

- Within each strata p, compute average Y

$$\bar{Y}|\boldsymbol{x}_p = \sum_{i=1}^{N} \frac{I(\boldsymbol{x}_i = \boldsymbol{x}_p)Y_i}{\sum_{t=1}^{N} I(\boldsymbol{x}_t = \boldsymbol{x}_p)}$$

Implies that for test data we would fit:

$$\hat{f}(\boldsymbol{x}_i) = \sum_{p=1}^{P} \bar{Y}|\boldsymbol{x}_p I(\boldsymbol{x}_i = \boldsymbol{x}_p)$$

Classification and Regression Trees (CART): Intuition

Consider regression $E[Y|\boldsymbol{x}_i]$.

With no assumptions, stratify $\rightsquigarrow$ different mean for unique values of $\boldsymbol{x}_i$

- Within each strata p, compute average Y

$$\bar{Y}|\boldsymbol{x}_p = \sum_{i=1}^{N} \frac{I(\boldsymbol{x}_i = \boldsymbol{x}_p)Y_i}{\sum_{t=1}^{N} I(\boldsymbol{x}_t = \boldsymbol{x}_p)}$$

Implies that for test data we would fit:

$$\begin{aligned}\hat{f}(\boldsymbol{x}_i) &= \sum_{p=1}^{P} \bar{Y}|\boldsymbol{x}_p I(\boldsymbol{x}_i = \boldsymbol{x}_p) \\ &= \sum_{p=1}^{P} c_p I(\boldsymbol{x}_i = \boldsymbol{x}_p)\end{aligned}$$

Classification and Regression Trees (CART): Intuition

Consider regression $E[Y|\boldsymbol{x}_i]$.

With no assumptions, stratify $\rightsquigarrow$ different mean for unique values of $\boldsymbol{x}_i$

- Within each strata p, compute average Y

$$\bar{Y}|\boldsymbol{x}_p = \sum_{i=1}^{N} \frac{I(\boldsymbol{x}_i = \boldsymbol{x}_p)Y_i}{\sum_{t=1}^{N} I(\boldsymbol{x}_t = \boldsymbol{x}_p)}$$

Implies that for test data we would fit:

$$\begin{aligned}\hat{f}(\boldsymbol{x}_i) &= \sum_{p=1}^{P} \bar{Y}|\boldsymbol{x}_p I(\boldsymbol{x}_i = \boldsymbol{x}_p) \\ &= \sum_{p=1}^{P} c_p I(\boldsymbol{x}_i = \boldsymbol{x}_p)\end{aligned}$$

Curse of dimensionality(!!!)

Classification and Regression Trees (CART): Intuition

Consider regression $E[Y|\boldsymbol{x}_i]$.

With no assumptions, stratify$\rightsquigarrow$ different mean for unique values of $\boldsymbol{x}_i$

- Within each strata p, compute average Y

$$\bar{Y}|\boldsymbol{x}_p = \sum_{i=1}^{N} \frac{I(\boldsymbol{x}_i = \boldsymbol{x}_p)Y_i}{\sum_{t=1}^{N} I(\boldsymbol{x}_t = \boldsymbol{x}_p)}$$

Implies that for test data we would fit:

$$\begin{aligned}\hat{f}(\boldsymbol{x}_i) &= \sum_{p=1}^{P} \bar{Y}|\boldsymbol{x}_p I(\boldsymbol{x}_i = \boldsymbol{x}_p) \\ &= \sum_{p=1}^{P} c_p I(\boldsymbol{x}_i = \boldsymbol{x}_p)\end{aligned}$$

Curse of dimensionality(!!!)

Approximate with regions$\rightsquigarrow$ search for splits of data to approximate stratification

Classification and Regression Trees (CART): Objective function

Labels $\boldsymbol{Y}_i$ and documents $\boldsymbol{x}_i$

$$
\begin{aligned}
E[Y|\boldsymbol{x}_i] &= \widehat{f}(\boldsymbol{x}_i) \\
&= \sum_{p=1}^{P} c_p I(\boldsymbol{x}_i \in R_p)
\end{aligned}
$$

where:

- R_p describes a region $\rightsquigarrow$ node
- c_p describes values of Y_i for document in R_p

Classification and Regression Trees (CART): Optimization function

Suppose we want to minimize sum of squared residuals with each node

Classification and Regression Trees (CART): Optimization function

Suppose we want to minimize sum of squared residuals with each node
Then c_p = Average Y for documents assigned to R_p

Classification and Regression Trees (CART): Optimization function

Suppose we want to minimize sum of squared residuals with each node
Then c_p = Average Y for documents assigned to R_p

$$\widehat{c}_p = \sum_{i=1}^{N} \frac{Y_i I(\boldsymbol{x}_i \in R_p)}{\sum_{j=1}^{N} I(\boldsymbol{x}_j \in R_p)}$$

Classification and Regression Trees (CART): Optimization function

Suppose we want to minimize sum of squared residuals with each node
Then c_p = Average Y for documents assigned to R_p

$$\widehat{c}_p = \sum_{i=1}^{N} \frac{Y_i I(\boldsymbol{x}_i \in R_p)}{\sum_{j=1}^{N} I(\boldsymbol{x}_j \in R_p)}$$

Determining an optimal partition $\rightsquigarrow$ NP-Hard.

Classification and Regression Trees (CART): Optimization function

Suppose we want to minimize sum of squared residuals with each node
Then c_p = Average Y for documents assigned to R_p

$$\widehat{c}_p = \sum_{i=1}^{N} \frac{Y_i I(\boldsymbol{x}_i \in R_p)}{\sum_{j=1}^{N} I(\boldsymbol{x}_j \in R_p)}$$

Determining an optimal partition $\rightsquigarrow$ NP-Hard.
Suppose we are in some node (perhaps at the start).

Classification and Regression Trees (CART): Optimization function

Suppose we want to minimize sum of squared residuals with each node
Then c_p = Average Y for documents assigned to R_p

$$\widehat{c}_p = \sum_{i=1}^{N} \frac{Y_i I(\boldsymbol{x}_i \in R_p)}{\sum_{j=1}^{N} I(\boldsymbol{x}_j \in R_p)}$$

Determining an optimal partition $\rightsquigarrow$ NP-Hard.
Suppose we are in some node (perhaps at the start).
Greedy algorithm:

Classification and Regression Trees (CART): Optimization function

Suppose we want to minimize sum of squared residuals with each node
Then c_p = Average Y for documents assigned to R_p

$$\widehat{c}_p = \sum_{i=1}^{N} \frac{Y_i I(\boldsymbol{x}_i \in R_p)}{\sum_{j=1}^{N} I(\boldsymbol{x}_j \in R_p)}$$

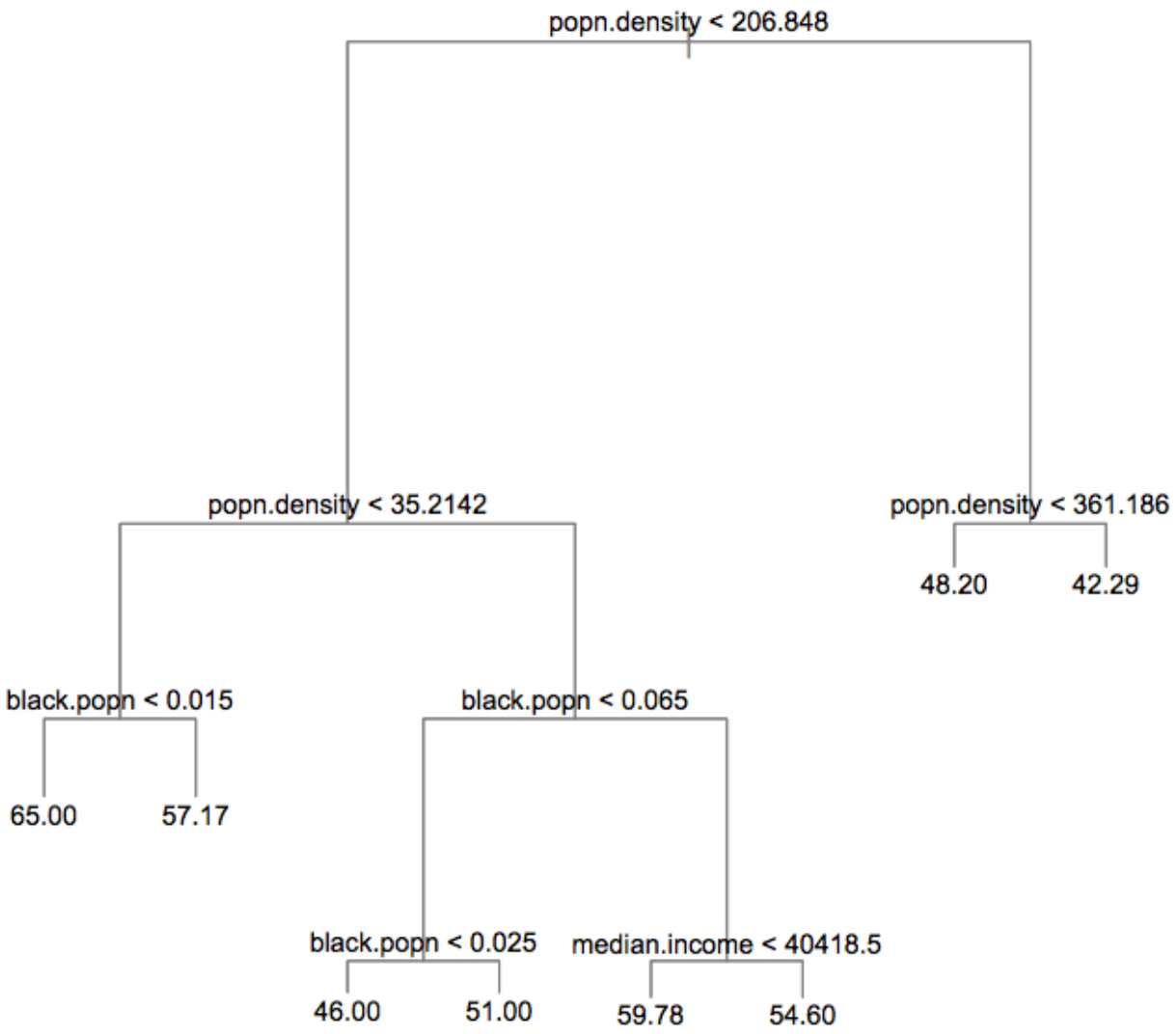
Determining an optimal partition $\rightsquigarrow$ NP-Hard.
Suppose we are in some node (perhaps at the start).
Greedy algorithm:

$$(j^*, s^*) = \arg\min_{j,s} \left[\underbrace{\min_{c_1} \sum_{i=1}^{N} I(x_{ij} < s)(Y_i - c_1)^2}_{\text{“cost” group 1}} + \underbrace{\min_{c_2} \sum_{i=1}^{N} I(x_{ij} > s)(Y_i - c_2)^2}_{\text{“cost” group 2}} \right]$$

Classification and Regression Trees (CART): Algorithm

- Start in Node
- Partition according to Greedy algorithm
- Continue until some stopping rule: number of observations per node

CART Picture (Spirling 2008)



Forests and Trees

Recall: accurate (unbiased) and uncorrelated classifiers

Forests and Trees

Recall: accurate (unbiased) and uncorrelated classifiers

- Grow trees deeply $\rightsquigarrow$ unbiased classifers, though high variance

Forests and Trees

Recall: accurate (unbiased) and uncorrelated classifiers

- Grow trees deeply $\rightsquigarrow$ unbiased classifers, though high variance
- Average $\rightsquigarrow$ reduces variance, but will be correlated

Forests and Trees

Recall: accurate (unbiased) and uncorrelated classifiers

- Grow trees deeply $\rightsquigarrow$ unbiased classifers, though high variance
- Average $\rightsquigarrow$ reduces variance, but will be correlated
- Random forest $\rightsquigarrow$ introduce additional sampling to induce independence $\rightsquigarrow$ Only split on subset of variables

Random Forest Algorithm (ESL, 588)

Random Forest Algorithm (ESL, 588)

1) For m bootstrap samples ($m = 1, \ldots, M$), draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m, \tilde{\boldsymbol{X}}_m$

Random Forest Algorithm (ESL, 588)

1) For m bootstrap samples ($m = 1, \ldots, M$), draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m, \tilde{\boldsymbol{X}}_m$
2) Until a minimum node size is reached:

Random Forest Algorithm (ESL, 588)

1) For m bootstrap samples ($m = 1, \ldots, M$), draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m, \tilde{\boldsymbol{X}}_m$
2) Until a minimum node size is reached:
   i) Select z of the J variables $\rightsquigarrow$ introduces independences across the trees

Random Forest Algorithm (ESL, 588)

1) For m bootstrap samples ($m = 1, \ldots, M$), draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m, \tilde{\boldsymbol{X}}_m$
2) Until a minimum node size is reached:
   i) Select z of the J variables $\rightsquigarrow$ introduces independences across the trees
   ii) Among those z, select the best split node

Random Forest Algorithm (ESL, 588)

1) For m bootstrap samples ($m = 1, \ldots, M$), draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m, \tilde{\boldsymbol{X}}_m$
2) Until a minimum node size is reached:
   i) Select z of the J variables $\rightsquigarrow$ introduces independences across the trees
   ii) Among those z, select the best split node
   iii) Split into daughter nodes

Random Forest Algorithm (ESL, 588)

1) For m bootstrap samples ($m = 1, \ldots, M$), draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m, \tilde{\boldsymbol{X}}_m$
2) Until a minimum node size is reached:
   i) Select z of the J variables $\rightsquigarrow$ introduces independences across the trees
   ii) Among those z, select the best split node
   iii) Split into daughter nodes
3) The result is an ensemble (forest) of trees $\boldsymbol{T} = (T_1, T_2, \ldots, T_M)$,

Random Forest Algorithm (ESL, 588)

1) For m bootstrap samples ($m = 1, \ldots, M$), draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m, \tilde{\boldsymbol{X}}_m$
2) Until a minimum node size is reached:
   i) Select z of the J variables $\rightsquigarrow$ introduces independences across the trees
   ii) Among those z, select the best split node
   iii) Split into daughter nodes
3) The result is an ensemble (forest) of trees $\boldsymbol{T} = (T_1, T_2, \ldots, T_M)$,

$$\hat{f}(\boldsymbol{x}_i) = \frac{1}{M} \sum_{m=1}^{M} T_m(\boldsymbol{x}_i)$$

Random Forest Algorithm (ESL, 588)

1) For m bootstrap samples ($m = 1, \ldots, M$), draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m, \tilde{\boldsymbol{X}}_m$
2) Until a minimum node size is reached:
   i) Select z of the J variables $\rightsquigarrow$ introduces independences across the trees
   ii) Among those z, select the best split node
   iii) Split into daughter nodes
3) The result is an ensemble (forest) of trees $\boldsymbol{T} = (T_1, T_2, \ldots, T_M)$,

$$\hat{f}(\boldsymbol{x}_i) = \frac{1}{M} \sum_{m=1}^{M} T_m(\boldsymbol{x}_i)$$

`RandomForest` $\rightsquigarrow$ Not a silver bullet!

Random Forest Algorithm (ESL, 588)

1) For m bootstrap samples ($m = 1, \ldots, M$), draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m, \tilde{\boldsymbol{X}}_m$
2) Until a minimum node size is reached:
   i) Select z of the J variables $\rightsquigarrow$ introduces independences across the trees
   ii) Among those z, select the best split node
   iii) Split into daughter nodes
3) The result is an ensemble (forest) of trees $\boldsymbol{T} = (T_1, T_2, \ldots, T_M)$,

$$\hat{f}(\boldsymbol{x}_i) = \frac{1}{M} \sum_{m=1}^{M} T_m(\boldsymbol{x}_i)$$

`RandomForest` $\rightsquigarrow$ Not a silver bullet!

- With many poor predictors $\rightsquigarrow$ the p selected may be meaningless

Random Forest Algorithm (ESL, 588)

1) For m bootstrap samples ($m = 1, \ldots, M$), draw N observations with replacement, $\tilde{\boldsymbol{Y}}_m, \tilde{\boldsymbol{X}}_m$
2) Until a minimum node size is reached:
   i) Select z of the J variables $\rightsquigarrow$ introduces independences across the trees
   ii) Among those z, select the best split node
   iii) Split into daughter nodes
3) The result is an ensemble (forest) of trees $\boldsymbol{T} = (T_1, T_2, \ldots, T_M)$,

$$\hat{f}(\boldsymbol{x}_i) = \frac{1}{M} \sum_{m=1}^{M} T_m(\boldsymbol{x}_i)$$

`RandomForest` $\rightsquigarrow$ Not a silver bullet!

- With many poor predictors $\rightsquigarrow$ the p selected may be meaningless
- Wager and Athey (2015): Random Forest for estimating heterogeneous effects

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
$Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words
     $\mathbf{x}_{i,\text{train}} = (X_{i1}, X_{i2}, \ldots, X_{iK})$

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words
     $\mathbf{x}_{i,\text{train}} = (X_{i1}, X_{i2}, \ldots, X_{iK})$

$$\Pr(Y_i = C_k | \mathbf{x}_i)$$

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words
     $\mathbf{x}_{i,\text{train}} = (X_{i1}, X_{i2}, \ldots, X_{iK})$

$$\Pr(Y_i = C_k | \mathbf{x}_i)_{\text{train}}$$

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words
     $\mathbf{x}_{i,\text{train}} = (X_{i1}, X_{i2}, \ldots, X_{iK})$

$$\Pr(Y_i = C_k | \mathbf{x}_i)_{\text{train}} \quad = \quad \widehat{g}(\mathbf{x}_i)_{\text{train}}$$

- Identify systematic relationship between words, labels

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words
     $\mathbf{x}_{i,\text{train}} = (X_{i1}, X_{i2}, \ldots, X_{iK})$

$$\Pr(Y_i = C_k | \mathbf{x}_i)_{\text{train}} \quad = \quad \widehat{g}(\mathbf{x}_i)_{\text{train}}$$

- Identify systematic relationship between words, labels $\rightsquigarrow$ Data and assumptions

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words
     $\mathbf{x}_{i,\text{train}} = (X_{i1}, X_{i2}, \ldots, X_{iK})$

$$\Pr(Y_i = C_k | \mathbf{x}_i)_{\text{train}} \quad = \quad \widehat{g}(\mathbf{x}_i)_{\text{train}}$$

- Identify systematic relationship between words, labels $\rightsquigarrow$ Data and assumptions
  - LASSO (Tibshirani 1996): sparsity

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words
     $\mathbf{x}_{i,\text{train}} = (X_{i1}, X_{i2}, \ldots, X_{iK})$

$$\Pr(Y_i = C_k | \mathbf{x}_i)_{\text{train}} \quad = \quad \widehat{g}(\mathbf{x}_i)_{\text{train}}$$

- Identify systematic relationship between words, labels $\rightsquigarrow$ Data and assumptions
  - LASSO (Tibshirani 1996): sparsity
  - KRLS (Hainmueller and Hazlett 2013): dense, flexible surface

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words
     $\mathbf{x}_{i,\text{train}} = (X_{i1}, X_{i2}, \ldots, X_{iK})$

$$\Pr(Y_i = C_k | \mathbf{x}_i)_{\text{train}} \quad = \quad \widehat{g}(\mathbf{x}_i)_{\text{train}}$$

- Identify systematic relationship between words, labels $\rightsquigarrow$ Data and assumptions
  - LASSO (Tibshirani 1996): sparsity
  - KRLS (Hainmueller and Hazlett 2013): dense, flexible surface
  - Ridge, Elastic-Net, SVM, Random Forests, BART, ...

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words
     $\mathbf{x}_{i,\text{train}} = (X_{i1}, X_{i2}, \ldots, X_{iK})$

$$\Pr(Y_i = C_k | \mathbf{x}_i)_{\text{train}} \quad = \quad \widehat{g}(\mathbf{x}_i)_{\text{train}}$$

- Identify systematic relationship between words, labels $\rightsquigarrow$ Data and assumptions
  - LASSO (Tibshirani 1996): sparsity
  - KRLS (Hainmueller and Hazlett 2013): dense, flexible surface
  - Ridge, Elastic-Net, SVM, Random Forests, BART, ...
- Which model? Difficult to know before hand

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words
     $\mathbf{x}_{i,\text{train}} = (X_{i1}, X_{i2}, \ldots, X_{iK})$

$$\Pr(Y_i = C_k | \mathbf{x}_i)_{\text{train}} \quad = \quad \widehat{g}(\mathbf{x}_i)_{\text{train}}$$

- Identify systematic relationship between words, labels $\rightsquigarrow$ Data and assumptions
  - LASSO (Tibshirani 1996): sparsity
  - KRLS (Hainmueller and Hazlett 2013): dense, flexible surface
  - Ridge, Elastic-Net, SVM, Random Forests, BART, ...
- Which model? Difficult to know before hand
- Assess out of sample performance with cross validation

Super Learning

1) Set of hand labeled documents. For each i, $(i = 1, \ldots, N_{\text{train}})$
   $Y_{i,\text{train}} \in \{C_1 C_2, \ldots C_K\}$
2) Estimate relationship between labels and words
   - Each document i is a count vector of K words
     $\mathbf{x}_{i,\text{train}} = (X_{i1}, X_{i2}, \ldots, X_{iK})$

$$\Pr(Y_i = C_k | \mathbf{x}_i)_{\text{train}} \quad = \quad \widehat{g}(\mathbf{x}_i)_{\text{train}}$$

- Identify systematic relationship between words, labels $\rightsquigarrow$ Data and assumptions
  - LASSO (Tibshirani 1996): sparsity
  - KRLS (Hainmueller and Hazlett 2013): dense, flexible surface
  - Ridge, Elastic-Net, SVM, Random Forests, BART, ...
- Which model? Difficult to know before hand
- Assess out of sample performance with cross validation

Weighted ensemble: weights determined by (unique) out of sample predictive performance

Committee Methods:
Fit many methods, average with equal weights

- Voting (classification)
- Averaging (predictions)

Problem: many poor methods may overwhelm high quality fit (remember earlier figures)
Solution: learn weights via cross validation

Weighted Ensemble to Classify Documents

- Suppose we have M $(m = 1, \ldots, M)$ models.

Weighted Ensemble to Classify Documents

- Suppose we have M $(m = 1, \ldots, M)$ models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} \quad = \quad \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

Weighted Ensemble to Classify Documents

- Suppose we have M $(m = 1, \ldots, M)$ models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} = \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

Weighted Ensemble to Classify Documents

- Suppose we have M $(m = 1, \ldots, M)$ models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} = \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

Weighted Ensemble to Classify Documents

- Suppose we have M ($m = 1, \ldots, M$) models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} \quad = \quad \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

- Estimate weights ($\widehat{\pi}_m$)

Weighted Ensemble to Classify Documents

- Suppose we have M ($m = 1, \ldots, M$) models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} \quad = \quad \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

- Estimate weights ($\widehat{\pi}_m$)
  - K-fold cross validation: generate M out of sample predictions for each document in training set

Weighted Ensemble to Classify Documents

- Suppose we have M ($m = 1, \ldots, M$) models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} \quad = \quad \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

- Estimate weights ($\widehat{\pi}_m$)
  - K-fold cross validation: generate M out of sample predictions for each document in training set
    $\widehat{\mathbf{Y}}_i = (\widehat{Y}_{i1}, \widehat{Y}_{i2}, \ldots, \widehat{Y}_{iM})$

Weighted Ensemble to Classify Documents

- Suppose we have M ($m = 1, \ldots, M$) models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} \quad = \quad \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

- Estimate weights ($\widehat{\pi}_m$)
  - K-fold cross validation: generate M out of sample predictions for each document in training set
    $\widehat{\mathbf{Y}}_i = (\widehat{Y}_{i1}, \widehat{Y}_{i2}, \ldots, \widehat{Y}_{iM})$
  - Estimate weights with constrained regression:

Weighted Ensemble to Classify Documents

- Suppose we have M ($m = 1, \ldots, M$) models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} = \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

- Estimate weights ($\widehat{\pi}_m$)
  - K-fold cross validation: generate M out of sample predictions for each document in training set
    $\widehat{\mathbf{Y}}_i = (\widehat{Y}_{i1}, \widehat{Y}_{i2}, \ldots, \widehat{Y}_{iM})$
  - Estimate weights with constrained regression:

$$Y_i = \sum_{m=1}^{M} \pi_m \hat{Y}_{im} + \epsilon_i$$

Weighted Ensemble to Classify Documents

- Suppose we have M ($m = 1, \ldots, M$) models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} \quad = \quad \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

- Estimate weights ($\widehat{\pi}_m$)
  - K-fold cross validation: generate M out of sample predictions for each document in training set
    $\widehat{\mathbf{Y}}_i = (\widehat{Y}_{i1}, \widehat{Y}_{i2}, \ldots, \widehat{Y}_{iM})$
  - Estimate weights with constrained regression:

$$Y_i \quad = \quad \sum_{m=1}^{M} \pi_m \hat{Y}_{im} + \epsilon_i$$

where we impose constraints: $\pi_m \geq 0$ and $\sum_{m=1}^{M} \pi_m = 1$.

Weighted Ensemble to Classify Documents

- Suppose we have M ($m = 1, \ldots, M$) models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} \quad = \quad \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

- Estimate weights ($\widehat{\pi}_m$)
  - K-fold cross validation: generate M out of sample predictions for each document in training set
    $\widehat{\mathbf{Y}}_i = (\widehat{Y}_{i1}, \widehat{Y}_{i2}, \ldots, \widehat{Y}_{iM})$
  - Estimate weights with constrained regression:

$$Y_i \quad = \quad \sum_{m=1}^{M} \pi_m \hat{Y}_{im} + \epsilon_i$$

    where we impose constraints: $\pi_m \geq 0$ and $\sum_{m=1}^{M} \pi_m = 1$.
  - Result $\widehat{\pi}_m$ for each method

Weighted Ensemble to Classify Documents

- Suppose we have M ($m = 1, \ldots, M$) models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} \quad = \quad \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

- Estimate weights ($\widehat{\pi}_m$)
- Estimate $\widehat{g}_m(\mathbf{x}_i) \rightsquigarrow$ Apply all M models to entire training set

Weighted Ensemble to Classify Documents

- Suppose we have M ($m = 1, \ldots, M$) models.

$$\Pr(Y_i = \mathrm{C}_1 | \mathbf{x}_i)_{\text{train}} \quad = \quad \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_i)$$

- Estimate weights ($\widehat{\pi}_m$)
- Estimate $\widehat{g}_m(\mathbf{x}_i) \rightsquigarrow$ Apply all M models to entire training set

3) For each document i in test set, $\mathbf{x}_{i,\text{test}}$

Weighted Ensemble to Classify Documents

- Suppose we have M ($m = 1, \ldots, M$) models.

$$\Pr(Y_i = C_1 | \mathbf{x}_{i,\text{test}})_{\text{train}} = \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_{i,\text{test}})$$

- Estimate weights ($\widehat{\pi}_m$)
- Estimate $\widehat{g}_m(\mathbf{x}_i) \rightsquigarrow$ Apply all M models to entire training set

3) For each document i in test set, $\mathbf{x}_{i,\text{test}}$

Weighted Ensemble to Classify Documents

- Suppose we have M ($m = 1, \ldots, M$) models.

$$\Pr(Y_i = C_1 | \mathbf{x}_{i,\text{test}})_{\text{train}} = \sum_{m=1}^{M} \widehat{\pi}_m \widehat{g}_m(\mathbf{x}_{i,\text{test}})$$

- Estimate weights ($\widehat{\pi}_m$)
- Estimate $\widehat{g}_m(\mathbf{x}_i) \rightsquigarrow$ Apply all M models to entire training set

3) For each document i in test set, $\mathbf{x}_{i,\text{test}}$
(Classify if above threshold)

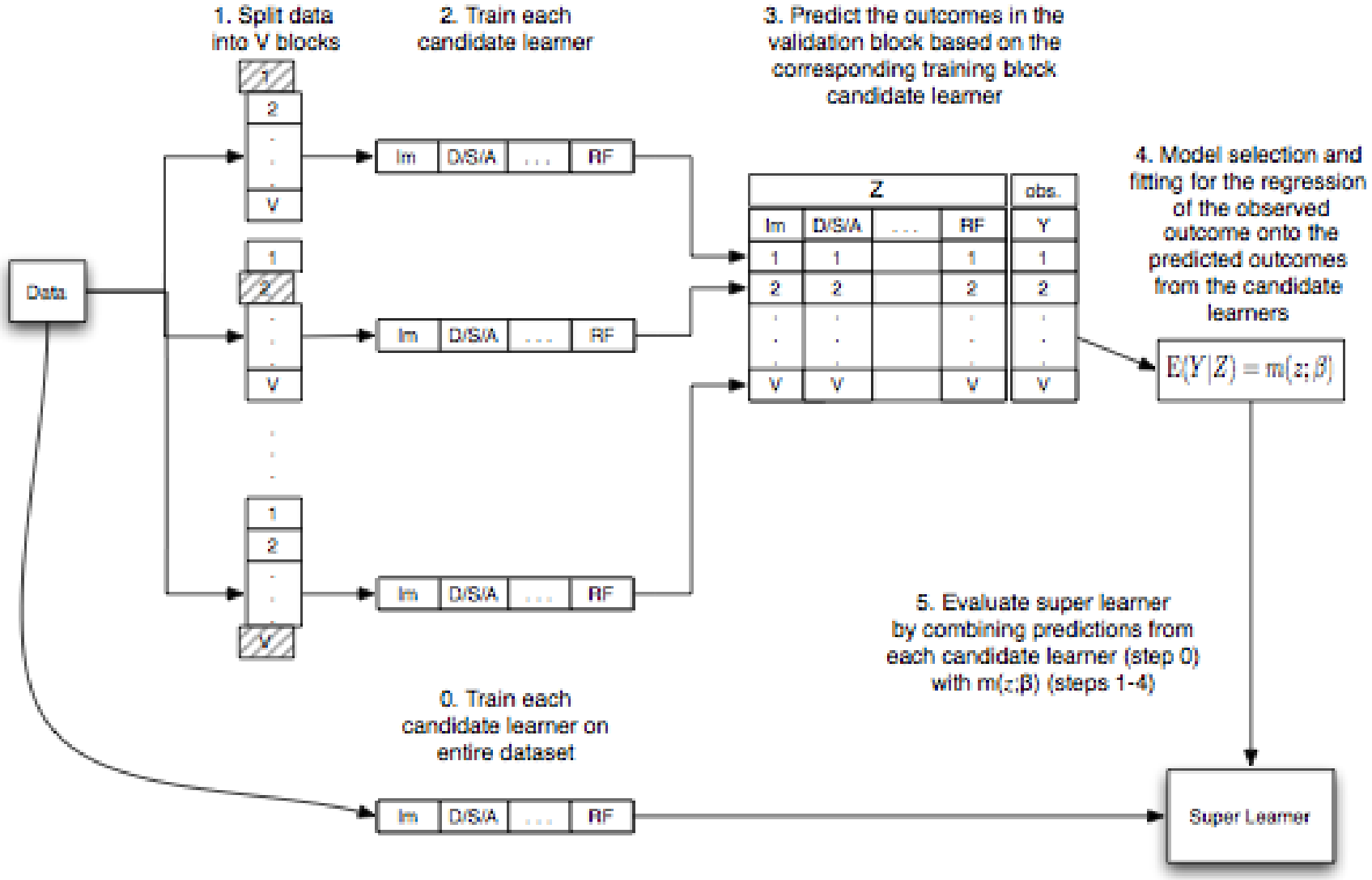


Figure 1: Flow Diagram for Super Learner

Why Super Learn?

van der Laan et al (2007) prove:

- Asymptotically: super learners will perform as well the best candidates for data
- Oracle: performs like the best possible method among candidate methods
    - Asymptotically outperforms constituent methods
    - Performs as well as optimal combinations of those methods

Practical questions:

- Final regression:
    - Logistic
    - Linear
    - Could super learn again!
- How Many Folds?
    - van der Laan et al's proofs rely on growing folds with N (but slowly)
    - Use 10-fold cross validation for simulations

Impression of Influence

Estimate: $Y_i \in \{\text{Credit}, \text{Not Credit}\}$

Impression of Influence

Estimate: $Y_i \in \{\text{Credit}, \text{Not Credit}\}$

- Triple hand code 800 press releases

Impression of Influence

Estimate: $Y_i \in \{\text{Credit}, \text{Not Credit}\}$

- Triple hand code 800 press releases
- Resolve disagreement with voting $\rightsquigarrow$ few disagreements

Impression of Influence

Estimate: $Y_i \in \{\text{Credit}, \text{Not Credit}\}$

- Triple hand code 800 press releases
- Resolve disagreement with voting $\rightsquigarrow$ few disagreements

Use five classifiers to form Ensemble (cross validating within each to tune parameters)

Impression of Influence

Estimate: $Y_i \in \{\text{Credit}, \text{Not Credit}\}$

- Triple hand code 800 press releases
- Resolve disagreement with voting $\rightsquigarrow$ few disagreements

Use five classifiers to form Ensemble (cross validating within each to tune parameters)

- LASSO

Impression of Influence

Estimate: $Y_i \in \{\text{Credit}, \text{Not Credit}\}$

- Triple hand code 800 press releases
- Resolve disagreement with voting $\rightsquigarrow$ few disagreements

Use five classifiers to form Ensemble (cross validating within each to tune parameters)

- LASSO
- Elastic-Net

Impression of Influence

Estimate: $Y_i \in \{\text{Credit}, \text{Not Credit}\}$

- Triple hand code 800 press releases
- Resolve disagreement with voting $\rightsquigarrow$ few disagreements

Use five classifiers to form Ensemble (cross validating within each to tune parameters)

- LASSO
- Elastic-Net
- Random Forest

Impression of Influence

Estimate: $Y_i \in \{\text{Credit}, \text{Not Credit}\}$

- Triple hand code 800 press releases
- Resolve disagreement with voting $\rightsquigarrow$ few disagreements

Use five classifiers to form Ensemble (cross validating within each to tune parameters)

- LASSO
- Elastic-Net
- Random Forest
- A Support Vector Machine

Impression of Influence

Estimate: $Y_i \in \{\text{Credit}, \text{Not Credit}\}$

- Triple hand code 800 press releases
- Resolve disagreement with voting $\rightsquigarrow$ few disagreements

Use five classifiers to form Ensemble (cross validating within each to tune parameters)

- LASSO
- Elastic-Net
- Random Forest
- A Support Vector Machine
- Kernel Regularized Least Squares (KRLS, Hainmueller and Hazlett 2014)

Impression of Influence

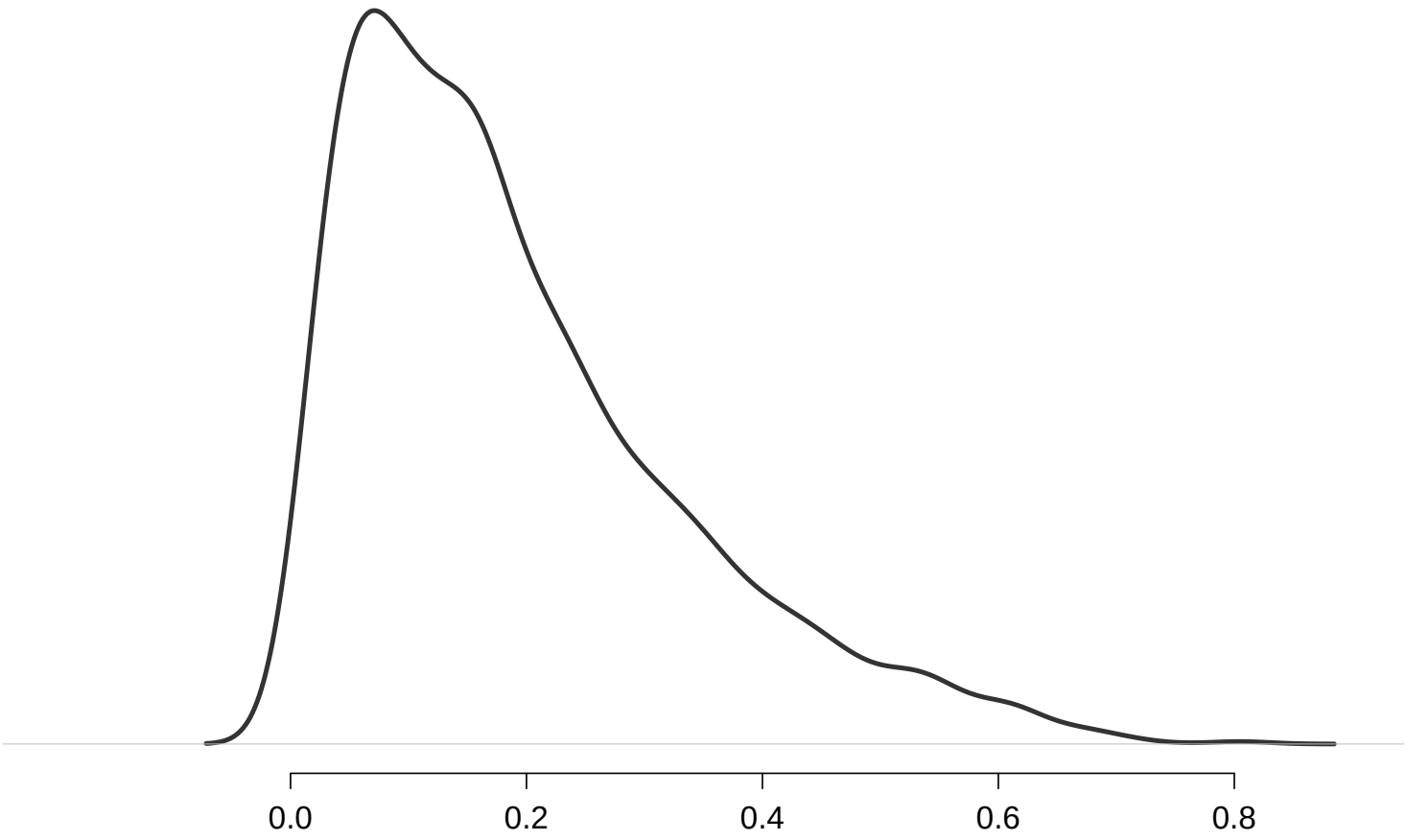
Estimate: $Y_i \in \{\text{Credit}, \text{Not Credit}\}$

- Triple hand code 800 press releases
- Resolve disagreement with voting$\rightsquigarrow$ few disagreements

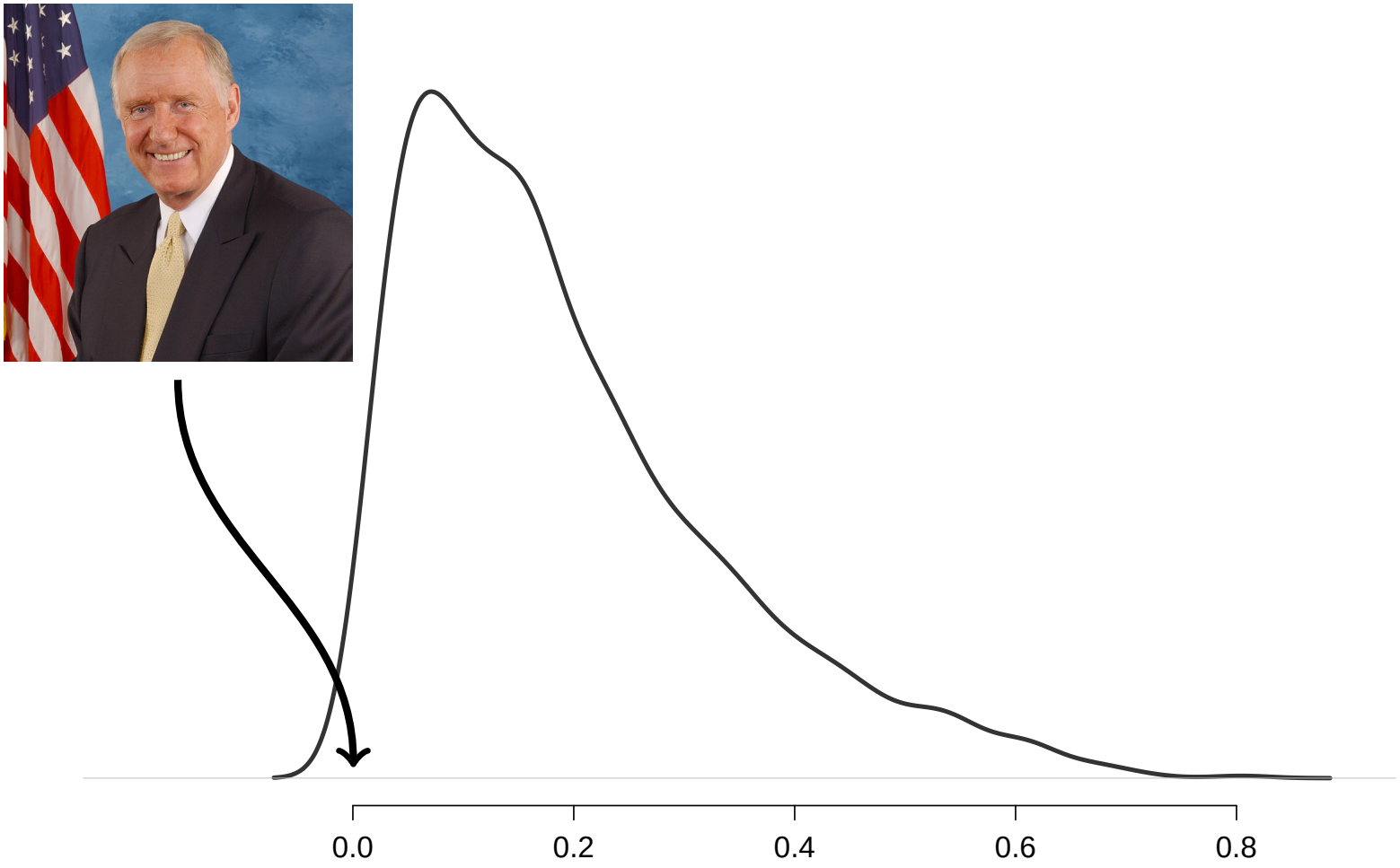
Use five classifiers to form Ensemble (cross validating within each to tune parameters)

- LASSO 0
- Elastic-Net 23%
- Random Forest 61%
- A Support Vector Machine16%
- Kernel Regularized Least Squares (KRLS, Hainmueller and Hazlett 2014) 0

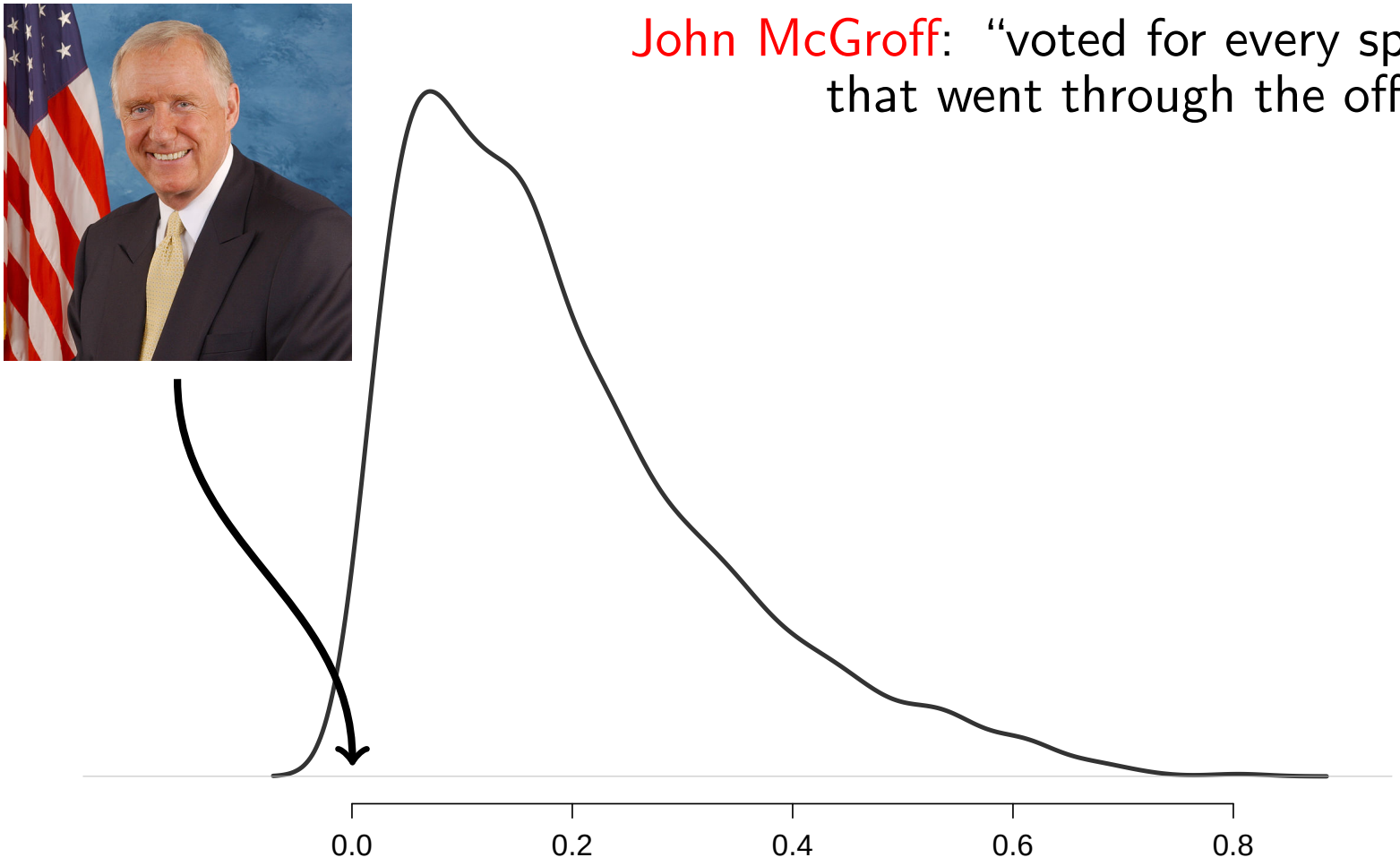
Strategic Credit Claiming to Build a Personal Vote



Strategic Credit Claiming to Build a Personal Vote

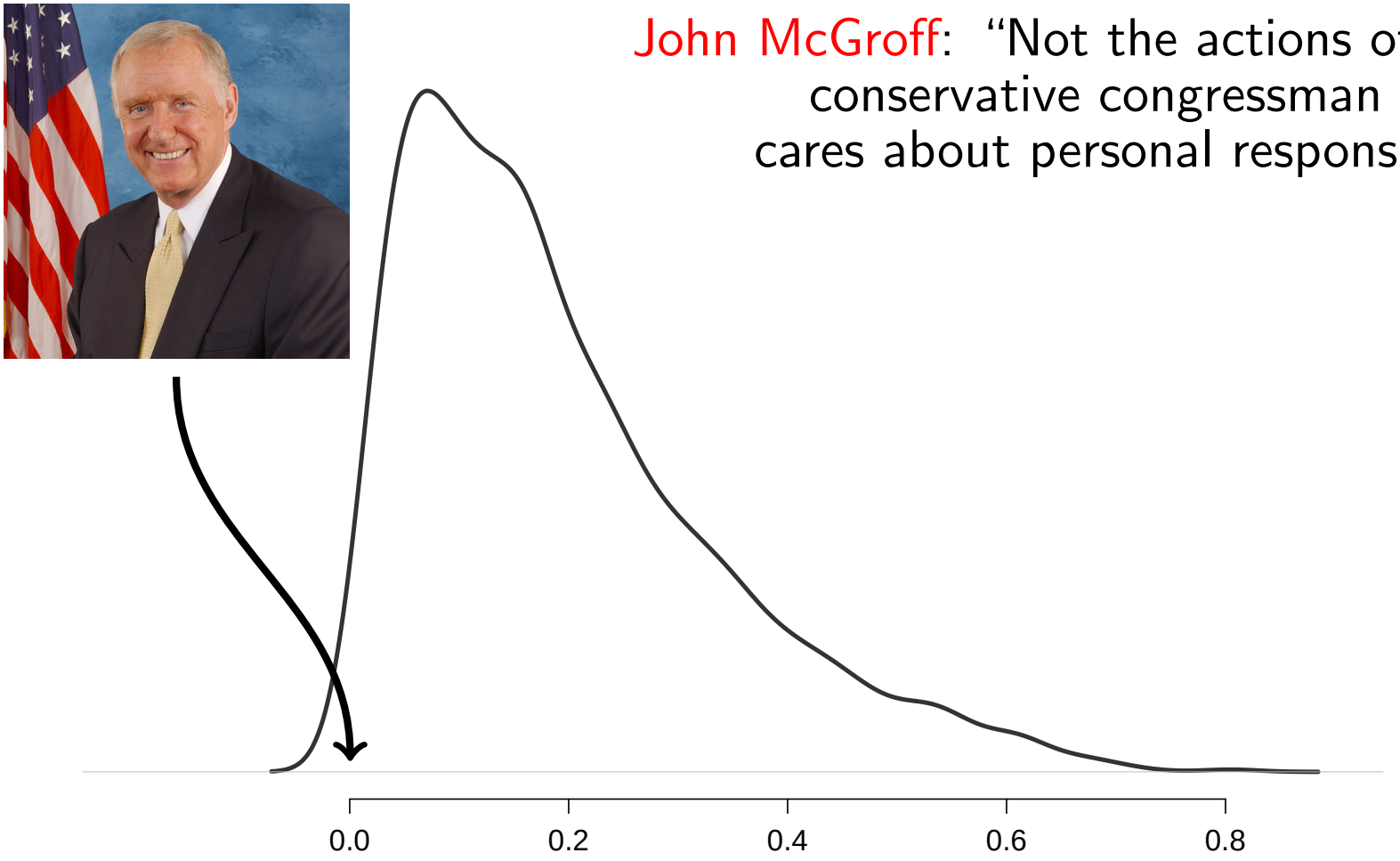


Strategic Credit Claiming to Build a Personal Vote



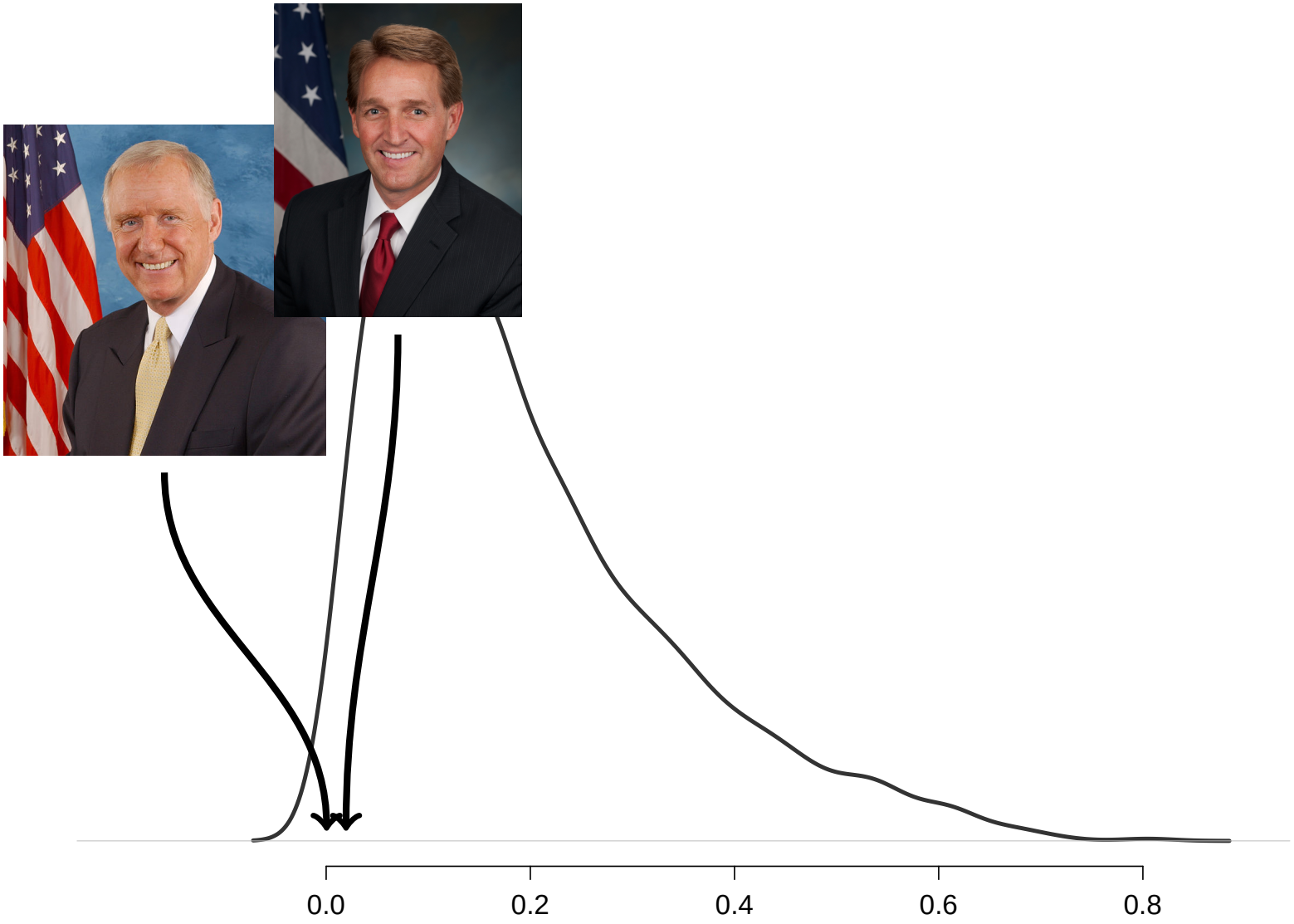
John McGroff: "voted for every spending bill " that went through the office"

Strategic Credit Claiming to Build a Personal Vote

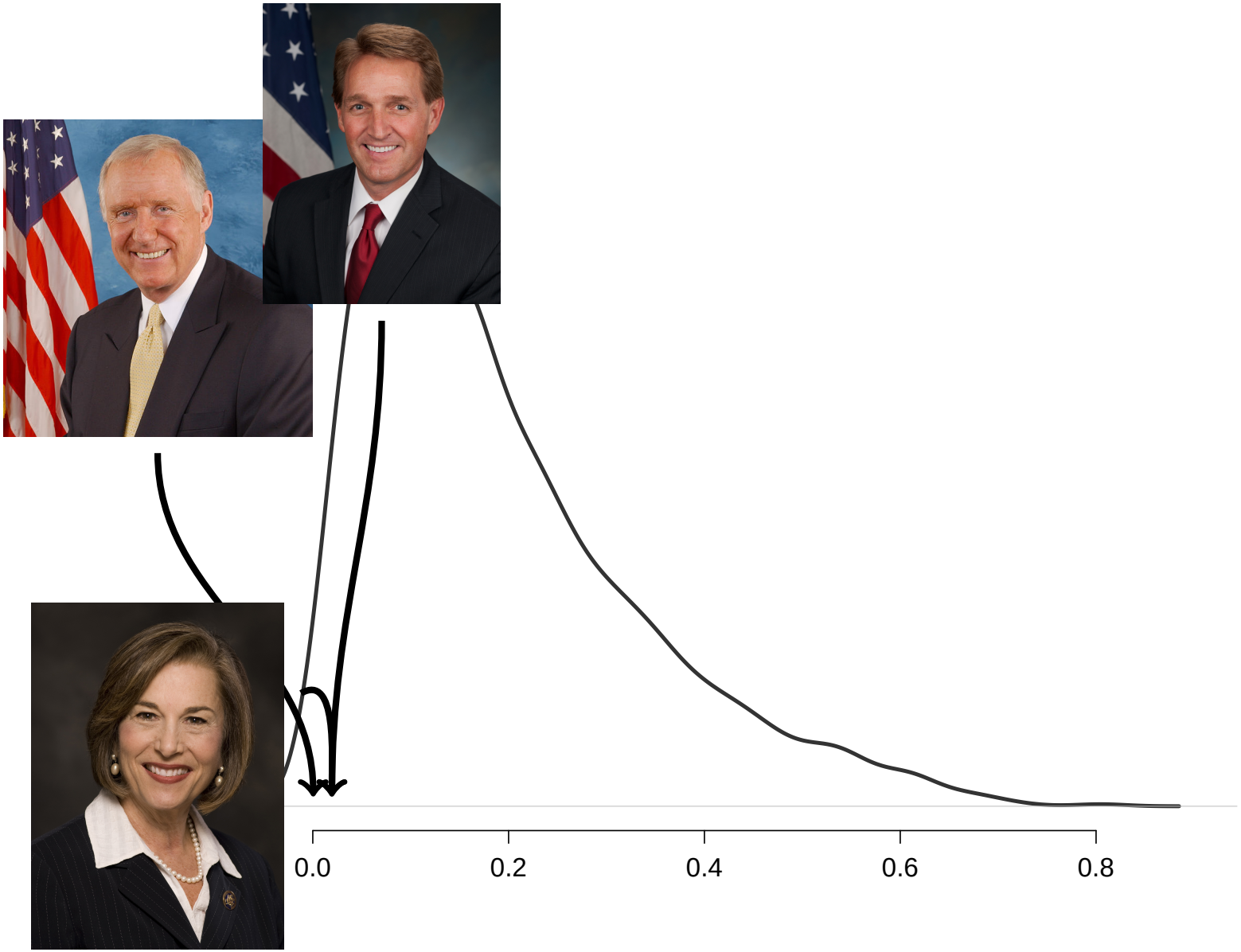


John McGroff: "Not the actions of a fiscally " conservative congressman who cares about personal responsibility"

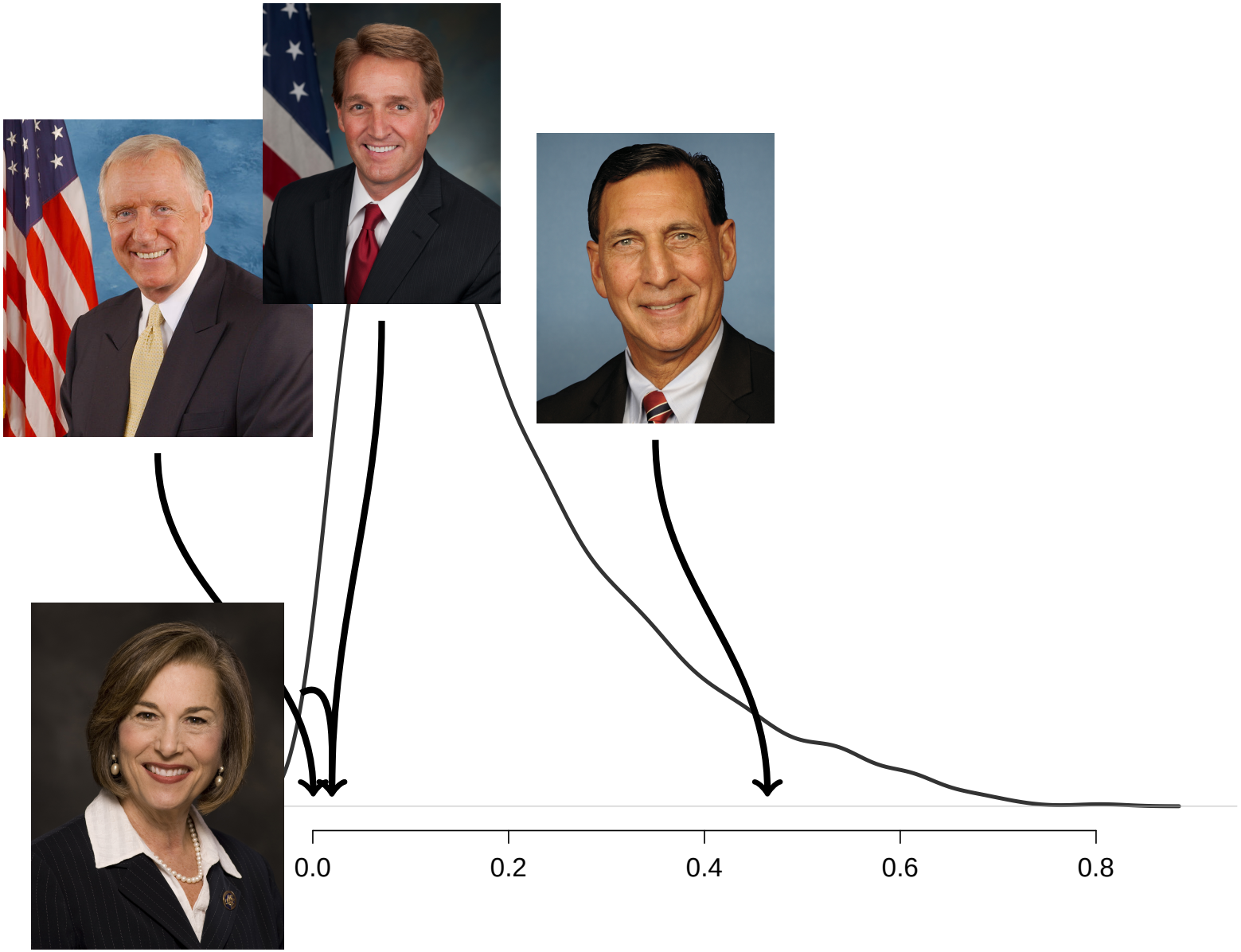
Strategic Credit Claiming to Build a Personal Vote



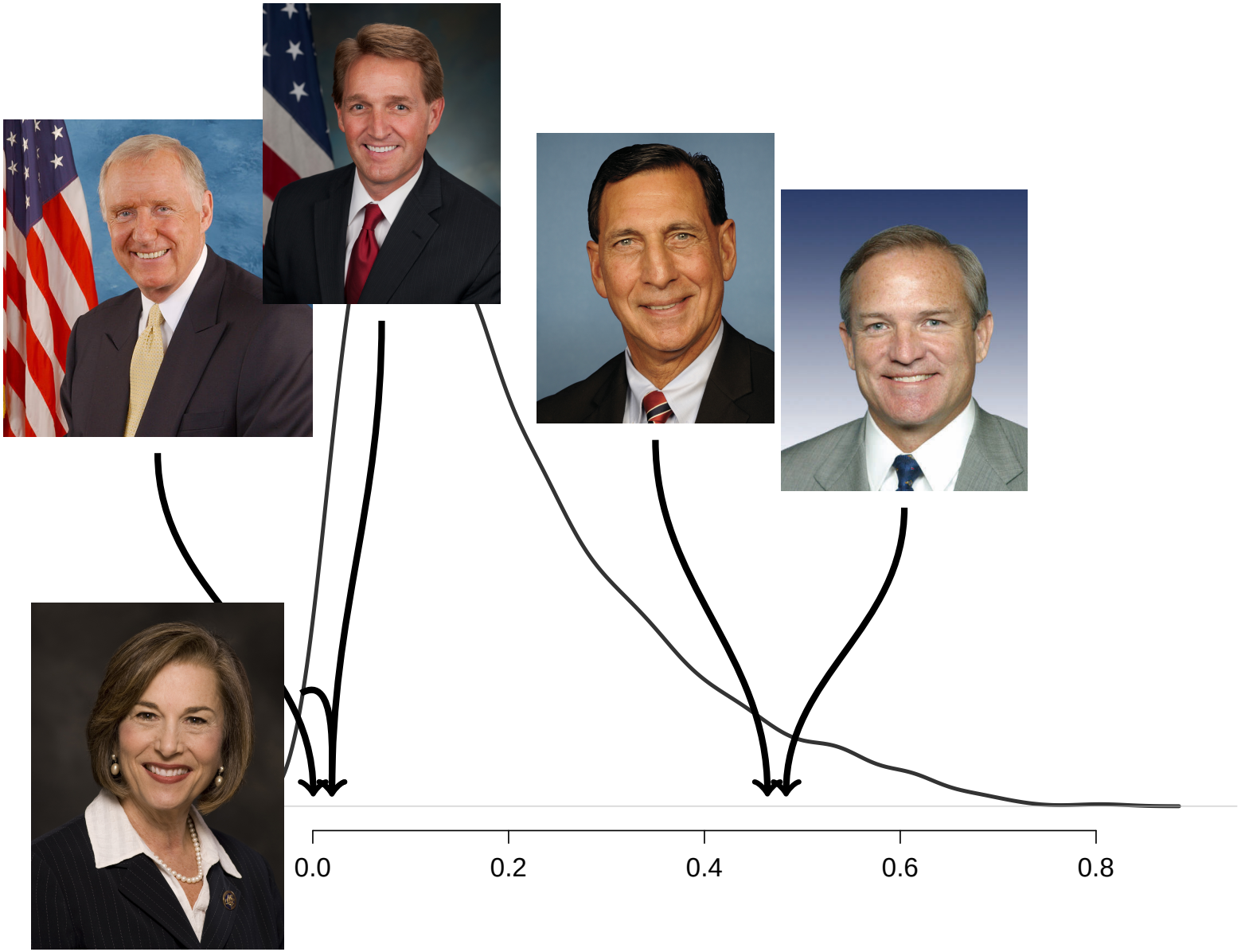
Strategic Credit Claiming to Build a Personal Vote



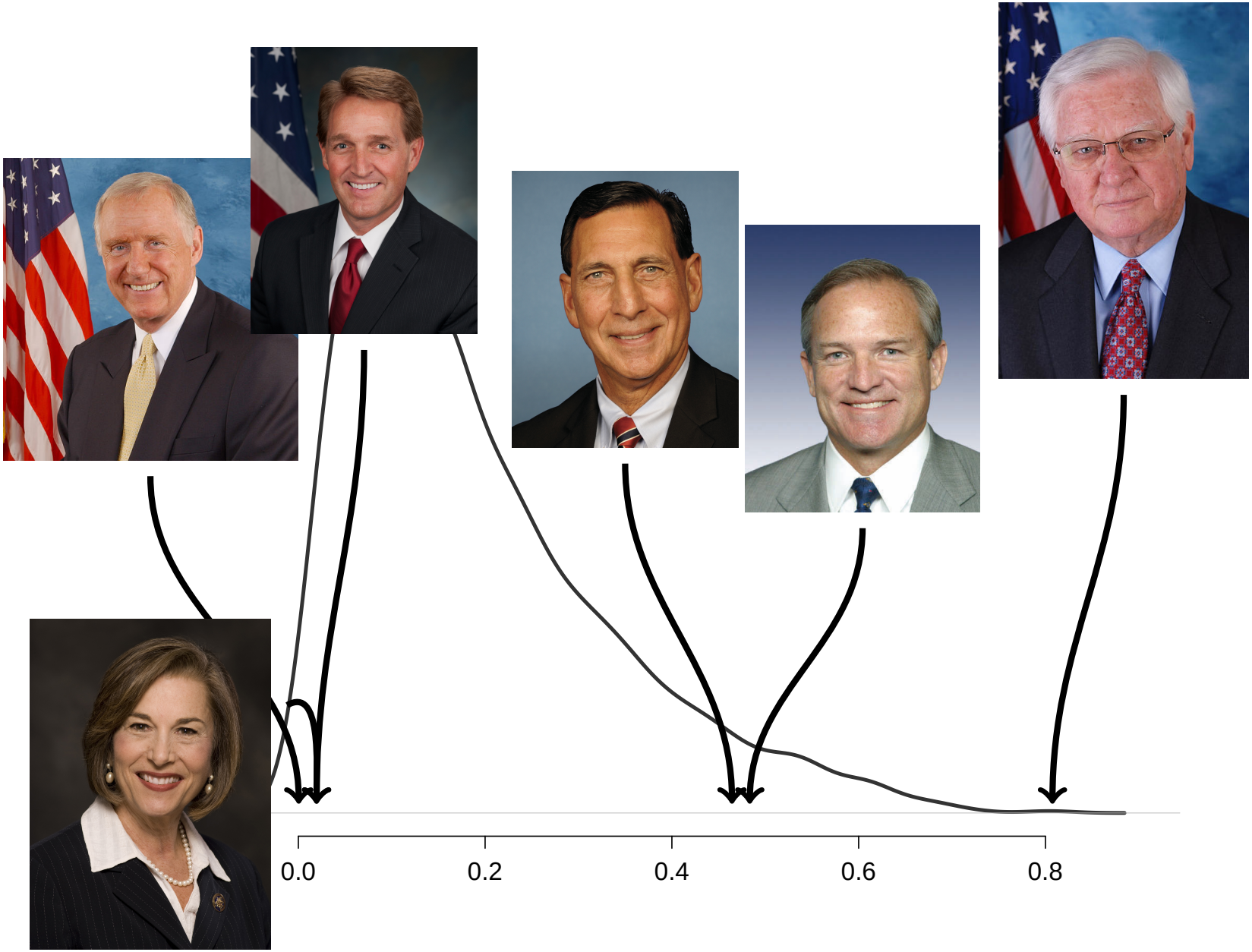
Strategic Credit Claiming to Build a Personal Vote



Strategic Credit Claiming to Build a Personal Vote

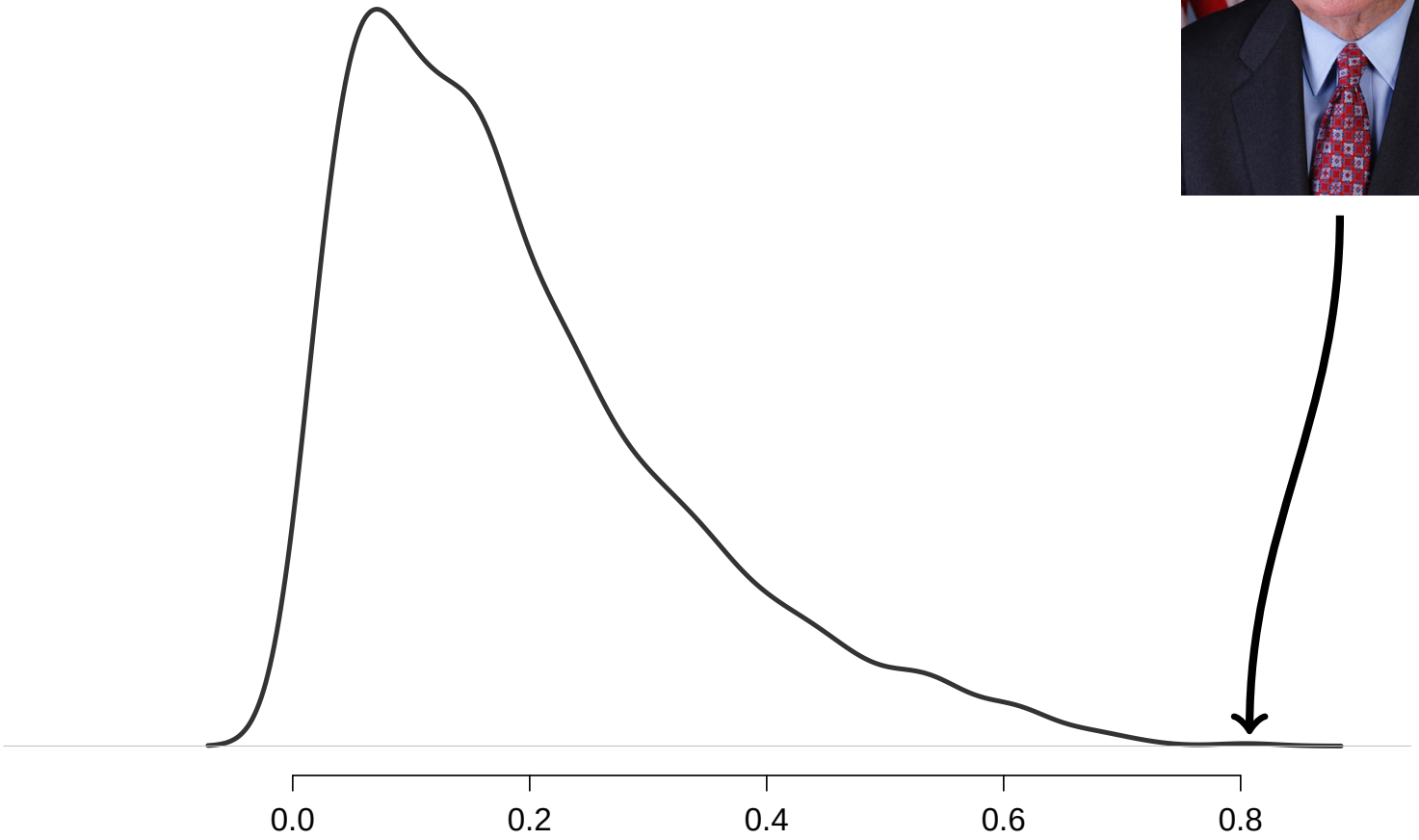


Strategic Credit Claiming to Build a Personal Vote



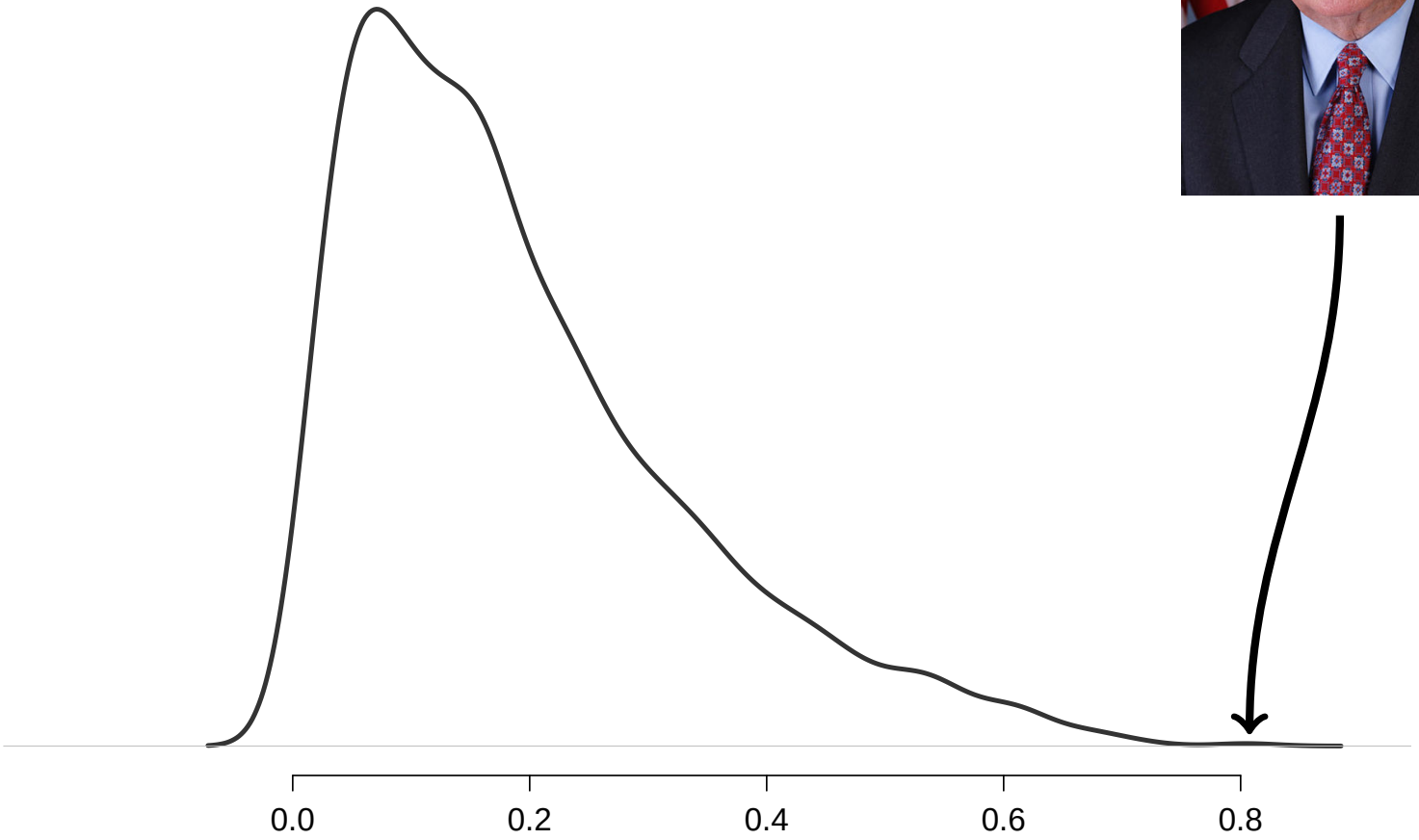
Strategic Credit Claiming to Build a Personal Vote

"We just can't afford luxuries like ideology"



Strategic Credit Claiming to Build a Personal Vote

Lexington Herald-Leader: Prince of Pork



Other Reasons to Ensemble (Dietterich 2000)

Statistical

Other Reasons to Ensemble (Dietterich 2000)

Statistical

- With little data, many algorithms offer similar performance

Other Reasons to Ensemble (Dietterich 2000)

Statistical

- With little data, many algorithms offer similar performance
- Ensemble ensures we avoid wrong model in test set

Other Reasons to Ensemble (Dietterich 2000)

Statistical

- With little data, many algorithms offer similar performance
- Ensemble ensures we avoid wrong model in test set

Computational

Other Reasons to Ensemble (Dietterich 2000)

Statistical

- With little data, many algorithms offer similar performance
- Ensemble ensures we avoid wrong model in test set

Computational

- Methods stuck in local modes

Other Reasons to Ensemble (Dietterich 2000)

Statistical

- With little data, many algorithms offer similar performance
- Ensemble ensures we avoid wrong model in test set

Computational

- Methods stuck in local modes
- Result: no one run provides best model

Other Reasons to Ensemble (Dietterich 2000)

Statistical

- With little data, many algorithms offer similar performance
- Ensemble ensures we avoid wrong model in test set

Computational

- Methods stuck in local modes
- Result: no one run provides best model
- Averages of runs may perform better

Other Reasons to Ensemble (Dietterich 2000)

Statistical

- With little data, many algorithms offer similar performance
- Ensemble ensures we avoid wrong model in test set

Computational

- Methods stuck in local modes
- Result: no one run provides best model
- Averages of runs may perform better

Complex "true" functional forms

Other Reasons to Ensemble (Dietterich 2000)

Statistical

- With little data, many algorithms offer similar performance
- Ensemble ensures we avoid wrong model in test set

Computational

- Methods stuck in local modes
- Result: no one run provides best model
- Averages of runs may perform better

Complex "true" functional forms

- One method may be unable to approximate true DGP

Other Reasons to Ensemble (Dietterich 2000)

Statistical

- With little data, many algorithms offer similar performance
- Ensemble ensures we avoid wrong model in test set

Computational

- Methods stuck in local modes
- Result: no one run provides best model
- Averages of runs may perform better

Complex "true" functional forms

- One method may be unable to approximate true DGP
- Mixtures of methods may approximate better

Other Reasons to Ensemble (Dietterich 2000)

Statistical

- With little data, many algorithms offer similar performance
- Ensemble ensures we avoid wrong model in test set

Computational

- Methods stuck in local modes
- Result: no one run provides best model
- Averages of runs may perform better

Complex "true" functional forms

- One method may be unable to approximate true DGP
- Mixtures of methods may approximate better