Text as Data

Justin Grimmer

Professor
Department of Political Science
Stanford University

May 28th, 2019

Selecting λ

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

Selecting λ

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

$$\begin{aligned} L\left(Y_i, f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right) &= \left(Y_i - f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right)^2 \\ &= \left|Y_i - f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i)\right| \\ &= I\left(Y_i = 1 - I(f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i) > \tau)\right) \end{aligned}$$

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:

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

We'd like to estimate out of sample performance with

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

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:

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

We'd like to estimate out of sample performance with

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

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

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

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 $E[\epsilon_i] = 0$
$\text{var}(\epsilon_i) = \sigma^2_{\epsilon}$

Loss Functions and Model Complexity

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

Loss Functions and Model Complexity

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

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

Loss Functions and Model Complexity

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

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

Loss Functions and Model Complexity

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

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

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

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

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?

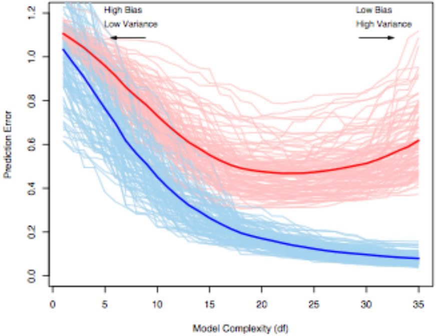


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$, *while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

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:

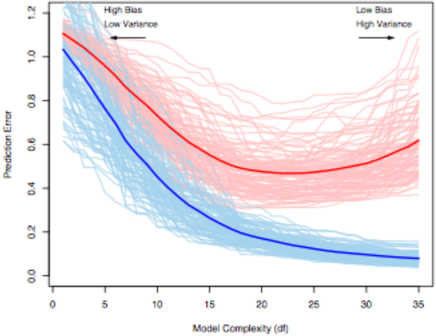


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error $\overline{\text{err}}$, while the light red curves show the conditional test error $\text{Err}_{\mathcal{T}}$ for 100 training sets of size 50 each, as the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

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)

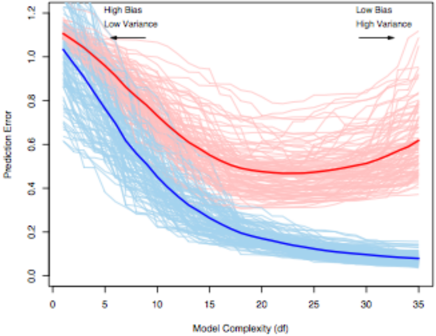


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$, *while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

How Do We Build A Model?

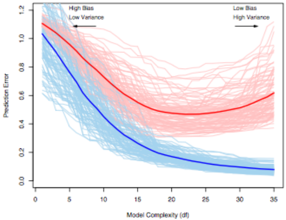


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

How Do We Build A Model?

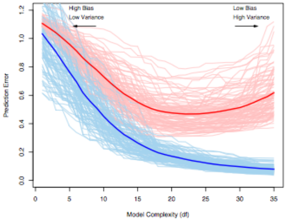


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

How Do We Build A Model?

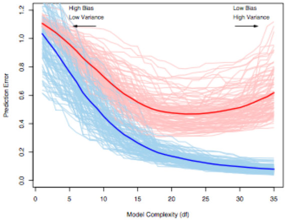


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$, *while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

How Do We Build A Model?

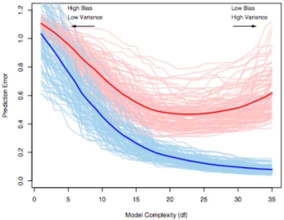


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$*, while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each,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

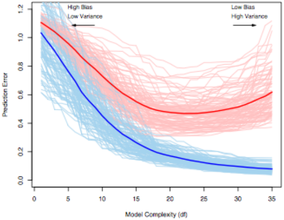


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

How Do We Build A Model?

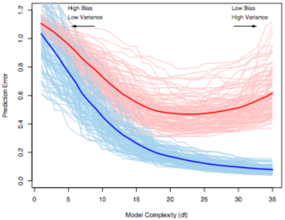


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

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

Analytic Solutions

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

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

Analytic Solutions

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

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

where d is the number of parameters in the model

Analytic Solutions

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

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

where d is 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 $N \rightarrow \infty$

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

where d is 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 $N \rightarrow \infty$

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

where d is 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 $N \rightarrow \infty$

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

where d is 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]

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

where d is again the effective number of parameters

Analytic Solutions

Bayesian Information Criterion (BIC) [Schwarz Criterion]

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

where d is again the effective number of parameters

- Intuition: balances model fit with penalty for complexity

Analytic Solutions

Bayesian Information Criterion (BIC) [Schwarz Criterion]

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

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

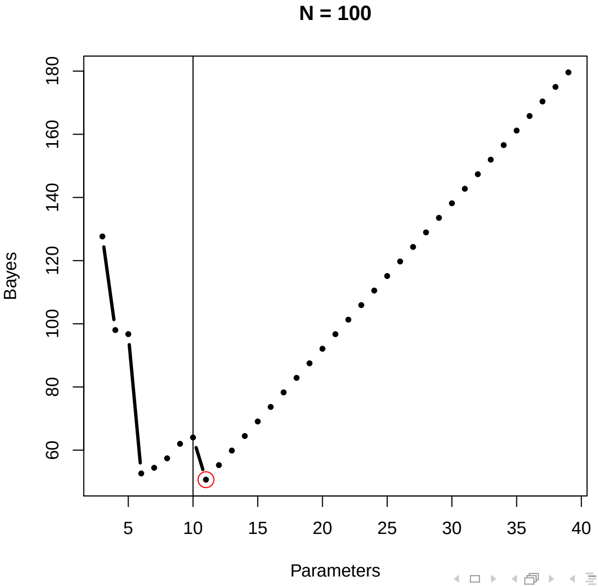
Bayesian Information Criterion (BIC) [Schwarz Criterion]

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

where d is again the effective number of parameters

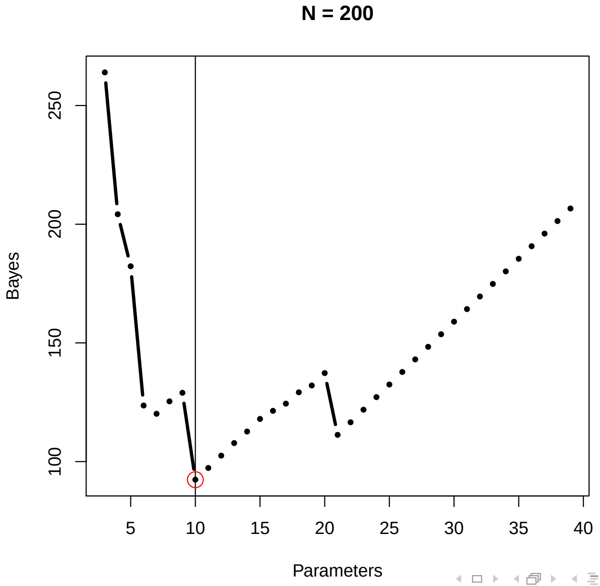
- Intuition: balances model fit with penalty for complexity
- Derived from Bayesian approach to model selection
- Approximation to Bayes' factor
- Penalizes more heavily than AIC

BIC or AIC?

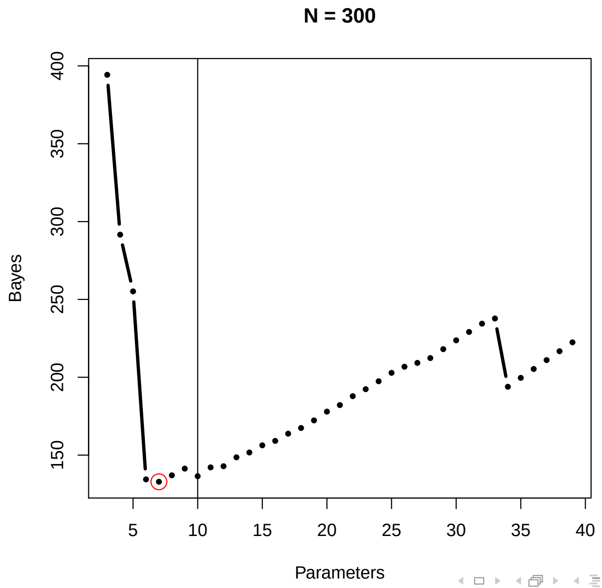


N = 100
Bayes
Parameters
60
80
100
120
140
160
180
5
10
15
20
25
30
35
40

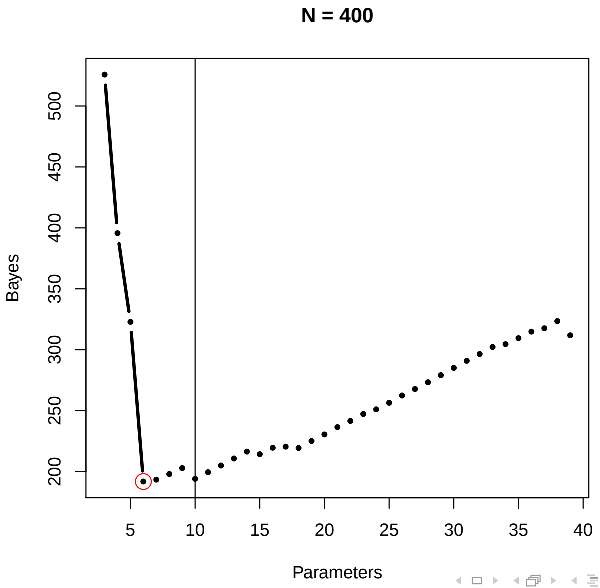
BIC or AIC?



BIC or AIC?

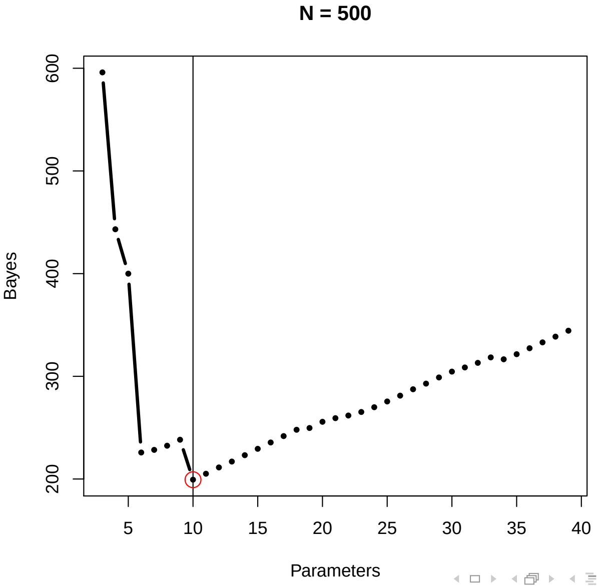

N = 300
Bayes
400
350
300
250
200
150
5
10
15
20
25
30
35
40
Parameters

BIC or AIC?

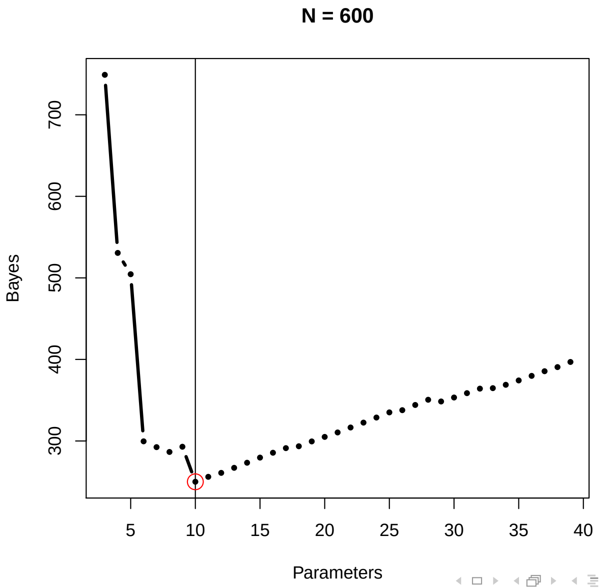


N = 400
Bayes
500
450
400
350
300
250
200
5
10
15
20
25
30
35
40
Parameters

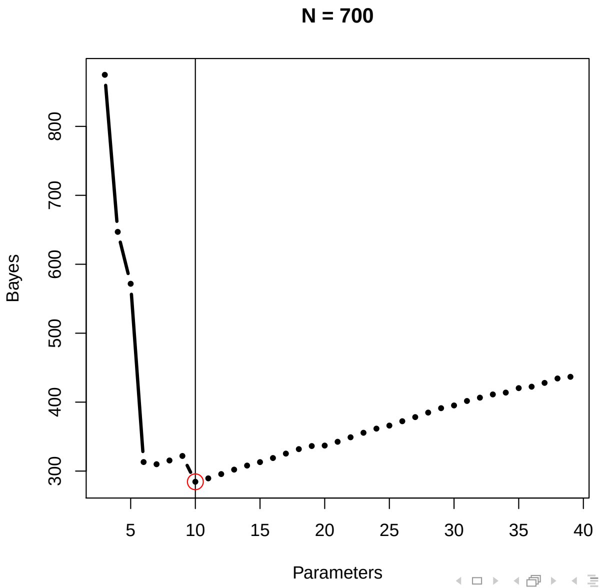
BIC or AIC?



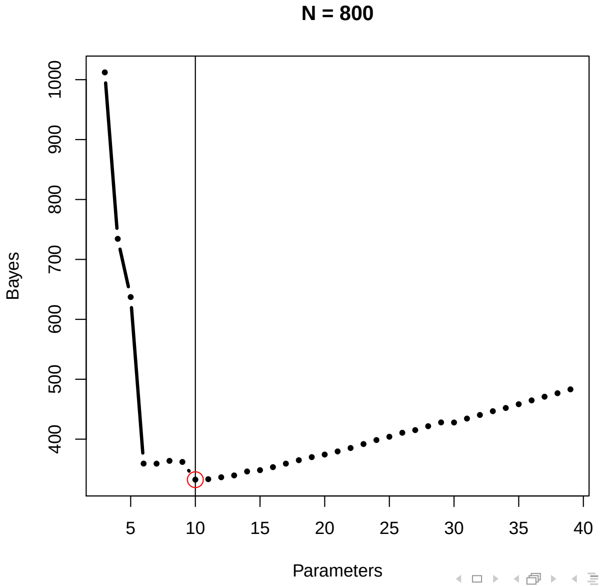
BIC or AIC?



BIC or AIC?

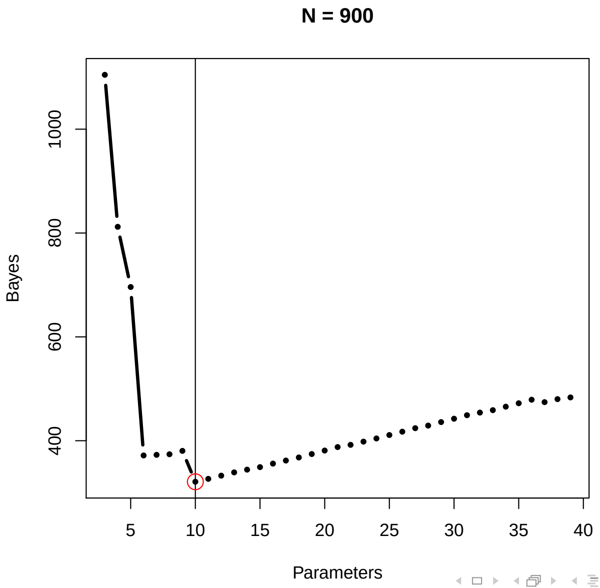

N = 700
Bayes
800
700
600
500
400
300
5
10
15
20
25
30
35
40
Parameters

BIC or AIC?

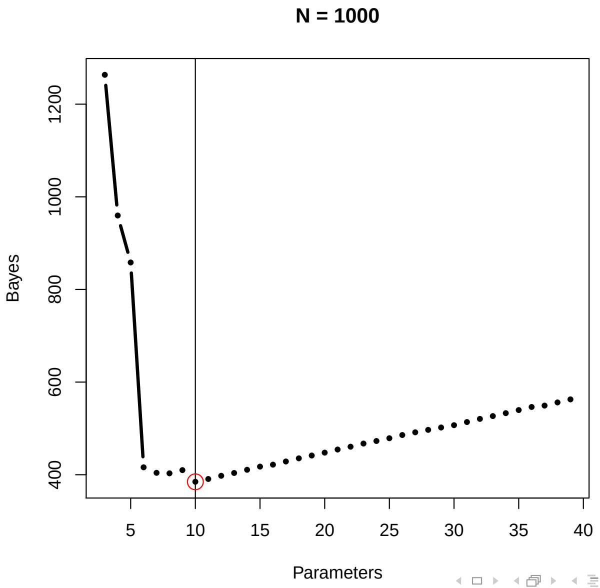


N = 800
Bayes
1000
900
800
700
600
500
400
5
10
15
20
25
30
35
40
Parameters

BIC or AIC?

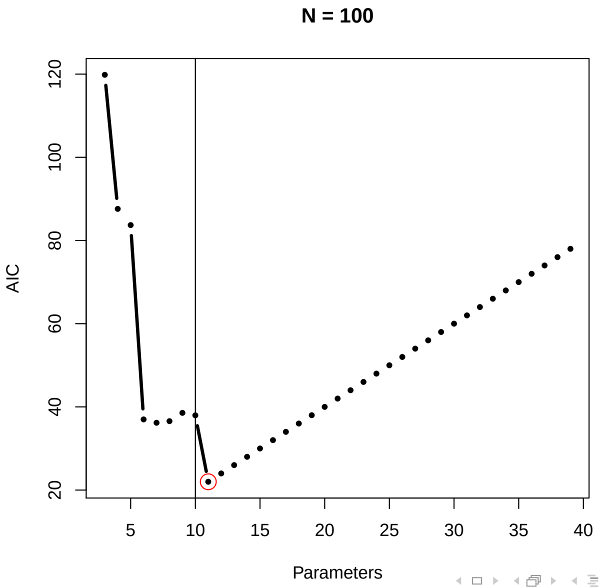


BIC or AIC?

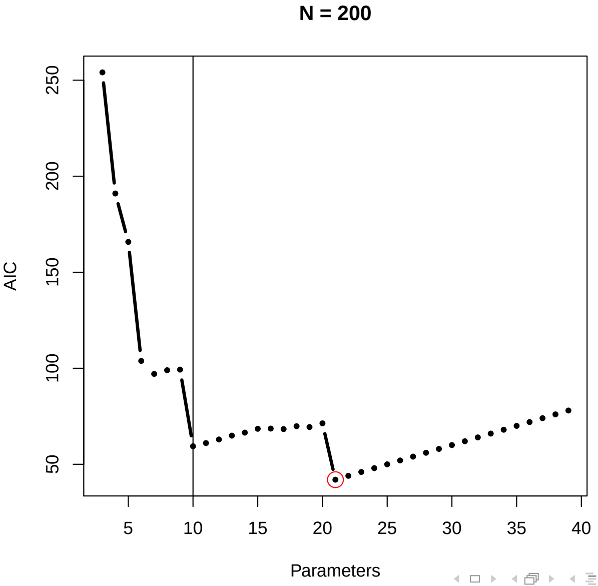


N = 1000
Bayes
1200
1000
800
600
400
5
10
15
20
25
30
35
40
Parameters

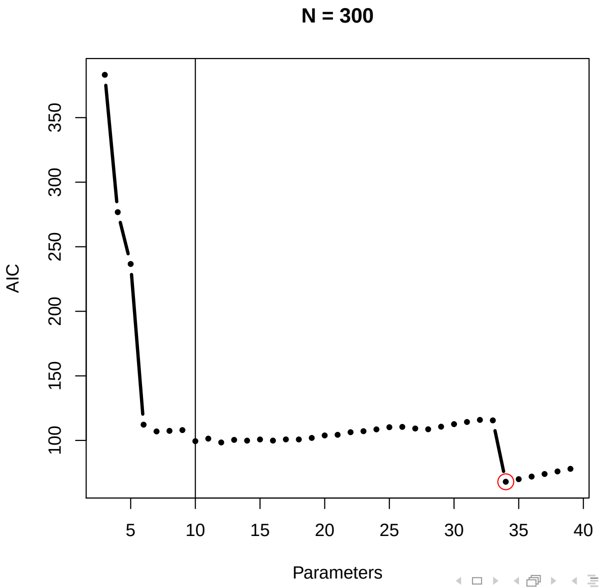
BIC or AIC?



BIC or AIC?

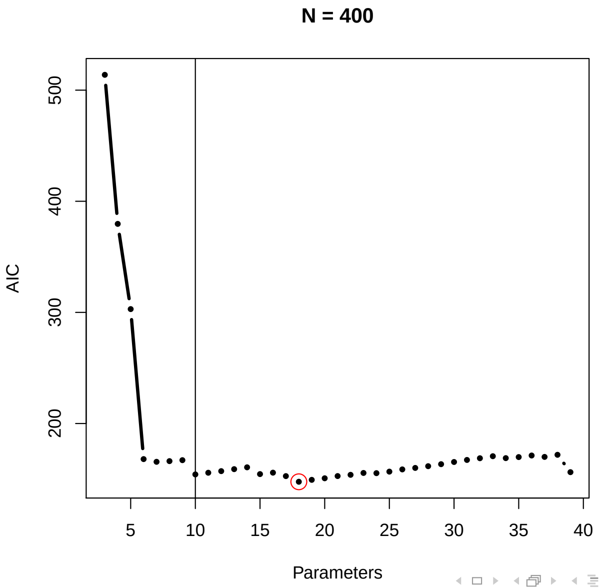


BIC or AIC?

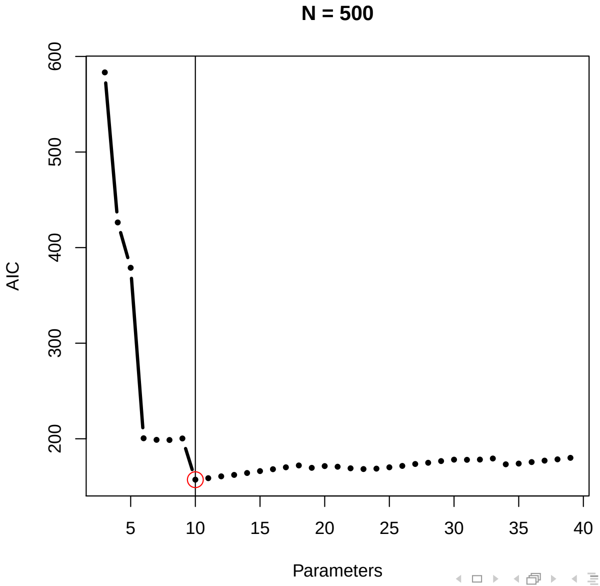


N = 300
AIC
350
300
250
200
150
100
5
10
15
20
25
30
35
40
Parameters

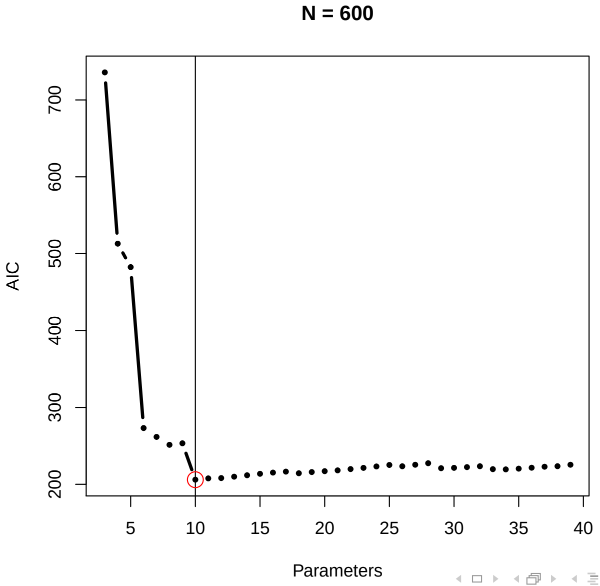
BIC or AIC?



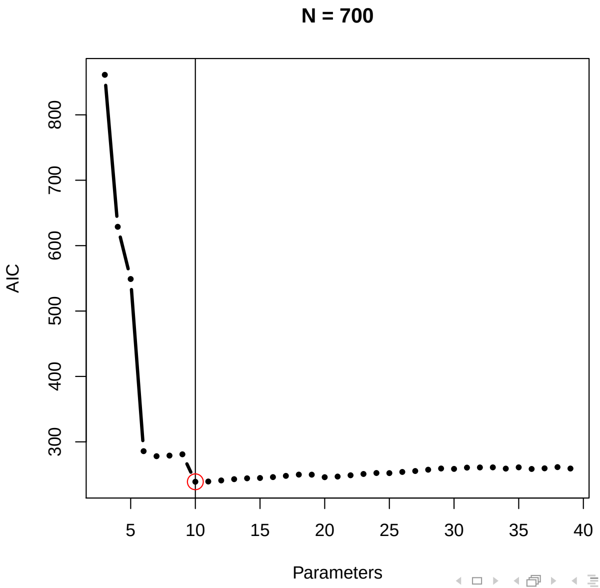
BIC or AIC?



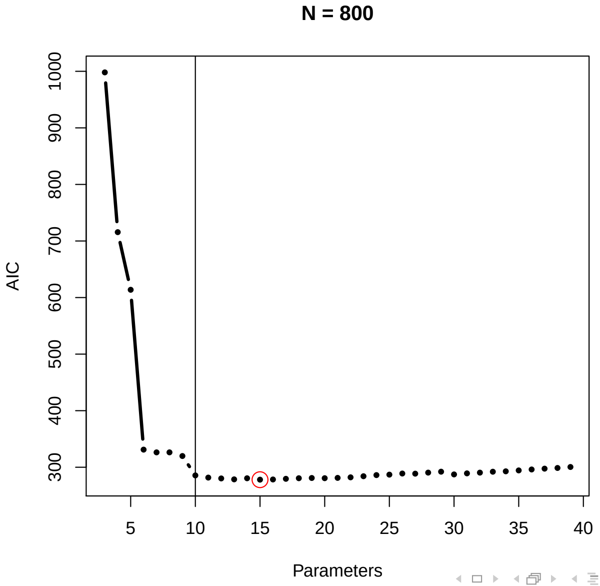
BIC or AIC?


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

BIC or AIC?

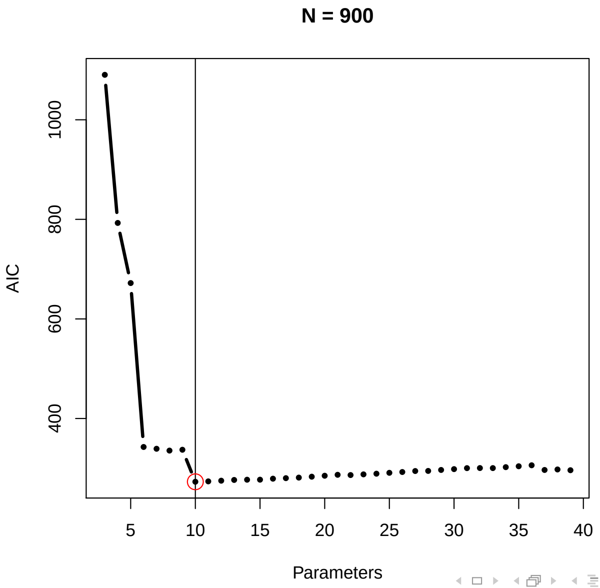

N = 700
AIC
800
700
600
500
400
300
5
10
15
20
25
30
35
40
Parameters

BIC or AIC?

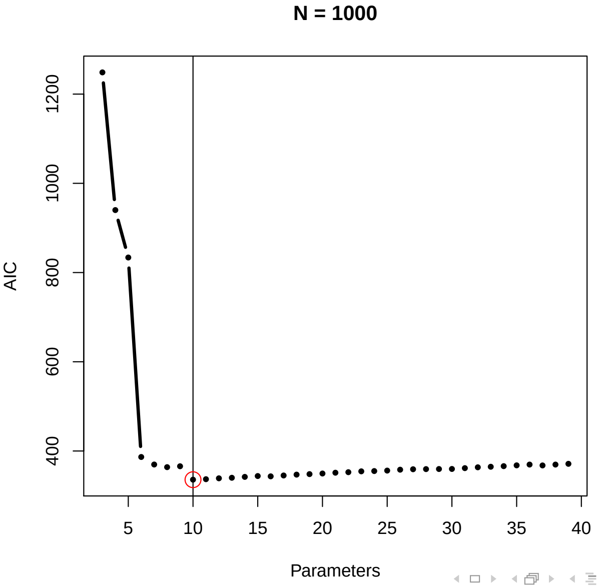


N = 800
AIC
1000
900
800
700
600
500
400
300
5
10
15
20
25
30
35
40
Parameters

BIC or AIC?



BIC or AIC?



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

BIC or AIC?

- BIC
  - Asymptotically consistent if true model is in choice set
  - As $N \rightarrow \infty$ 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:

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

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
$\vdots$	$\vdots$	$\vdots$
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
$\vdots$	$\vdots$	$\vdots$
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
$\vdots$	$\vdots$	$\vdots$
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
$\vdots$	$\vdots$	$\vdots$
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
$\vdots$	$\vdots$	$\vdots$
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 $\hat{f}^{-K}(\beta, \boldsymbol{X})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\beta, \boldsymbol{X}))$
  - 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, ...

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

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
$\vdots$	$\vdots$	$\vdots$
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

Credit Claiming (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 (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:

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

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:

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

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

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

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

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

Generalized Cross Validation and Ridge Regression

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

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

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

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

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

Generalized Cross Validation and Ridge Regression

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

- Trace($\boldsymbol{H}$) $\equiv$ Tr($\boldsymbol{H}$) $= \sum_{i=1}^{N} H_{ii}$
- 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 j^{th} Eigenvalue from $\boldsymbol{\Sigma} = \boldsymbol{X}^{'}\boldsymbol{X}$

Generalized Cross Validation and Ridge Regression

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

- Trace($\boldsymbol{H}$) $\equiv$ Tr($\boldsymbol{H}$) $= \sum_{i=1}^{N} H_{ii}$
- 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 j^{th} Eigenvalue from $\boldsymbol{\Sigma} = \boldsymbol{X}^{'}\boldsymbol{X}$ (!!!!!)

Generalized Cross Validation and Ridge Regression

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

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

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

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

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