Text as Data

Justin Grimmer

Associate Professor
Department of Political Science
Stanford University

November 13th, 2014

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, SVM, LASSO, Ridge, . . .: focused on individual document classification.

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, SVM, LASSO, Ridge, . . .: focused on individual document classification.
But what if we're focused on proportions only?

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, SVM, LASSO, Ridge, ...: focused on individual document classification.

But what if we're focused on proportions only?

Hopkins and King (2010): method for characterizing distribution of classes

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, SVM, LASSO, Ridge, . . .: focused on individual document classification.

But what if we're focused on proportions only?

Hopkins and King (2010): method for characterizing distribution of classes

Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, SVM, LASSO, Ridge, ...: focused on individual document classification.

But what if we're focused on proportions only?

Hopkins and King (2010): method for characterizing distribution of classes

Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, SVM, LASSO, Ridge, . . .: focused on individual document classification.

But what if we're focused on proportions only?

Hopkins and King (2010): method for characterizing distribution of classes

Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies
- Hopkins and King (2010): extend the method to text documents

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, SVM, LASSO, Ridge, . . .: focused on individual document classification.

But what if we're focused on proportions only?

Hopkins and King (2010): method for characterizing distribution of classes

Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies
- Hopkins and King (2010): extend the method to text documents

Basic intuition:

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Naive Bayes, SVM, LASSO, Ridge, ...: focused on individual document classification.

But what if we're focused on proportions only?

Hopkins and King (2010): method for characterizing distribution of classes

Can be much more accurate than individual classifiers, requires fewer assumptions (do not need random sample of documents) .

- King and Lu (2008): derive method for characterizing causes of deaths for verbal autopsies
- Hopkins and King (2010): extend the method to text documents

Basic intuition:

- Examine joint distribution of characteristics (without making Naive Bayes like assumption)
- Focus on distributions (only) makes this analysis possible

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(J x 1)$ vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$P(\boldsymbol{x}) = \text{probability of observing } \boldsymbol{x}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$\begin{aligned} P(\boldsymbol{x}) &= \text{probability of observing } \boldsymbol{x} \\ P(\boldsymbol{x}|C_j) &= \text{Probability of observing } \boldsymbol{x} \text{ conditional on category } C_j \end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$\begin{aligned}
P(\boldsymbol{x}) &= \text{probability of observing } \boldsymbol{x} \\
P(\boldsymbol{x}|C_j) &= \text{Probability of observing } \boldsymbol{x} \text{ conditional on category } C_j \\
P(\boldsymbol{X}|C) &= \text{Matrix collecting vectors}
\end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [(Jx1) vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$\begin{aligned}
P(\boldsymbol{x}) &= \text{probability of observing } \boldsymbol{x} \\
P(\boldsymbol{x}|C_j) &= \text{Probability of observing } \boldsymbol{x} \text{ conditional on category } C_j \\
P(\boldsymbol{X}|C) &= \text{Matrix collecting vectors} \\
P(C) &= P(C_1, C_2, \ldots, C_K) \text{ target quantity of interest}
\end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

Measure only presence/absence of each term [$(J x 1)$ vector]

$$\boldsymbol{x}_i = (1, 0, 0, 1, \ldots, 0)$$

What are the possible realizations of $\boldsymbol{x}_i$?

- 2^J possible vectors

Define:

$$\begin{aligned}
P(\boldsymbol{x}) &= \text{probability of observing } \boldsymbol{x} \\
P(\boldsymbol{x}|C_j) &= \text{Probability of observing } \boldsymbol{x} \text{ conditional on category } C_j \\
P(\boldsymbol{X}|C) &= \text{Matrix collecting vectors} \\
P(C) &= P(C_1, C_2, \ldots, C_K) \text{ target quantity of interest}
\end{aligned}$$

ReadMe: Optimization for a Different Goal (Hopkins and King 2010)

$$\underbrace{P(\boldsymbol{x})}_{2^J x 1} = \underbrace{P(\boldsymbol{x}|C)}_{2^J x K}\underbrace{P(C)}_{K x 1}$$

Matrix algebra problem to solve, for $P(C)$
Like Naive Bayes, requires two pieces to estimate
Complication $2^J >>$ no. documents
Kernel Smoothing Methods (without a formal model)

- $P(\boldsymbol{x})$ = estimate directly from test set
- $P(\boldsymbol{x}|C)$ = estimate from training set
  - Key assumption: $P(\boldsymbol{x}|C)$ in training set is equivalent to $P(\boldsymbol{x}|C)$ in test set
- If true, can perform biased sampling of documents, worry less about drift...

Algorithm Summarized

- Estimate $\hat{p}(\boldsymbol{x})$ from test set
- Estimate $\hat{p}(\boldsymbol{x}|C)$ from training set
- Use $\hat{p}(\boldsymbol{x})$ and $\hat{p}(\boldsymbol{x}|C)$ to solve for $p(C)$

Assessing Model Performance

Not classifying individual documents → different standards
Mean Square Error :

$$E[(\hat{\theta} - \theta)^2] = \text{var}(\hat{\theta}) + \text{Bias}(\hat{\theta}, \theta)^2$$

Suppose we have true proportions $P(C)^{\text{true}}$. Then, we'll estimate Root Mean Square Error

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{J}(P(C_j)^{\text{true}} - P(C_j))}{J}}$$

$$\text{Mean Abs. Prediction Error} = |\frac{\sum_{j=1}^{J}(P(C_j)^{\text{true}} - P(C_j))}{J}|$$

Visualize: plot true and estimated proportions

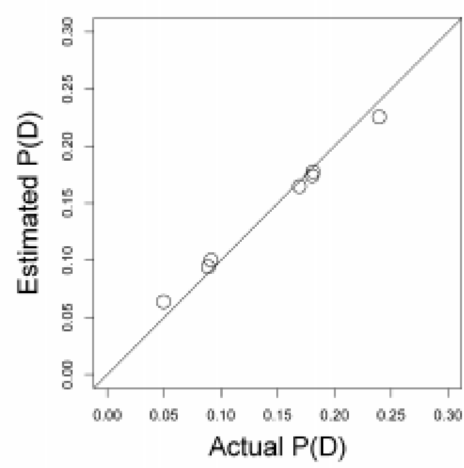

Estimated P(D)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Actual P(D)

Table 1 Performance of Our Nonparametric Approach and Four Support Vector Machine Analyses

	Percent of Blog Posts Correctly Classified			
	In-Sample Fit	In-Sample Cross-Validation	Out-of-Sample Prediction	Mean Absolute Proportion Error
Nonparametric	—	—	—	1.2
Linear	67.6	55.2	49.3	7.7
Radial	67.6	54.2	49.1	7.7
Polynomial	99.7	48.9	47.8	5.3
Sigmoid	15.6	15.6	18.2	23.2

Notes: Each row is the optimal choice over numerous individual runs given a specific kernel. Leaving aside the sigmoid kernel, individual classification performance in the first three columns does not correlate with mean absolute error in the document category proportions in the last column.

Using the House Press Release Data

Method	RMSE	APSE
ReadMe	0.036	0.056
NaiveBayes	0.096	0.14
SVM	0.052	0.084

Code to Run in R

Control file:

filename	truth	trainingset
20July2009LEWIS53.txt	4	1
26July2006LEWIS249.txt	2	0

```
tdm<- undergrad(control=control, fullfreq=F)
process<- preprocess(tdm)
output<- undergrad(process)
output$est.CSMF ## proportion in each category
output$true.CSMF ## if labeled for validation set (but not
used in training set)
```

Classification (Prediction)

1) Task
   - Classify Documents
   - Measure proportions
2) Objective Function

$$Y = f(\underbrace{\boldsymbol{\beta}}_{\text{coefficients}}, \boldsymbol{X}, \boldsymbol{Y}, \underbrace{\boldsymbol{\lambda}}_{\text{Tuning}}) + \epsilon$$

   Models often assume $\boldsymbol{\lambda}$ are known $\rightsquigarrow$ search over lambda values
3) Optimization
   - Grid search, examine loss function
   - Best procedure: test performance on data held in "vault"
   - Approximate in two ways:
     a) Analytically: AIC, BIC
     b) Computationally: Cross validation
4) Validation
   - Out of sample predictive performance

Loss Functions and Model Complexity

Suppose each document i has labels (scores) Y_i and count vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

Loss Functions and Model Complexity

Suppose each document i has labels (scores) Y_i and count vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.
Fit model to obtain $\hat{\boldsymbol{\beta}}$.

Loss Functions and Model Complexity

Suppose each document i has labels (scores) Y_i and count vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.
Fit model to obtain $\hat{\boldsymbol{\beta}}$.
Potential loss functions:

Loss Functions and Model Complexity

Suppose each document i has labels (scores) Y_i and count vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.
Fit model to obtain $\hat{\boldsymbol{\beta}}$.
Potential loss functions:

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

Loss Functions and Model Complexity

Suppose each document i has labels (scores) Y_i and count vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.
Fit model to obtain $\hat{\boldsymbol{\beta}}$.
Potential loss functions:

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

Loss Functions and Model Complexity

Suppose each document i has labels (scores) Y_i and count vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.
Fit model to obtain $\hat{\boldsymbol{\beta}}$.
Potential loss functions:

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

Loss Functions and Model Complexity

Suppose each document i has labels (scores) Y_i and count vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.
Fit model to obtain $\hat{\boldsymbol{\beta}}$.
Potential loss functions:

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

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:

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

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:

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

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:

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

We'd like to estimate out of sample performance with

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

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:

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

We'd like to estimate out of sample performance with

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

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:

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

We'd like to estimate out of sample performance with

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

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

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

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 $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}, \boldsymbol{\lambda}) = \hat{f}(\boldsymbol{x})$

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 $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}, \boldsymbol{\lambda}) = \hat{f}(\boldsymbol{x})$
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 $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}, \boldsymbol{\lambda}) = \hat{f}(\boldsymbol{x})$
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 $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}, \boldsymbol{\lambda}) = \hat{f}(\boldsymbol{x})$
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 $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}, \boldsymbol{\lambda}) = \hat{f}(\boldsymbol{x})$
With squared error loss:

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

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 $f(\hat{\boldsymbol{\beta}}, \boldsymbol{x}, \boldsymbol{\lambda}) = \hat{f}(\boldsymbol{x})$
With squared error loss:

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

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?

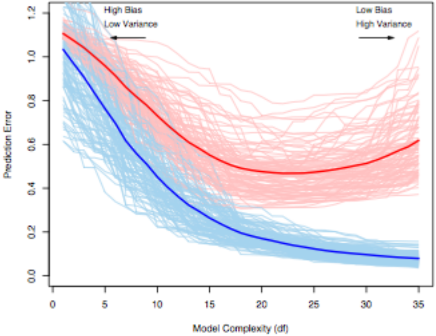


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

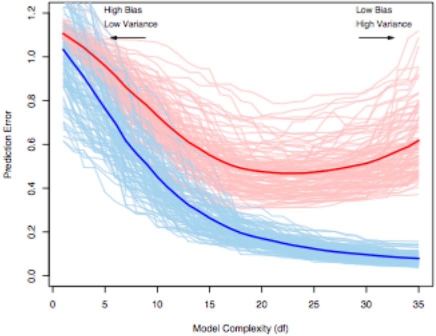


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

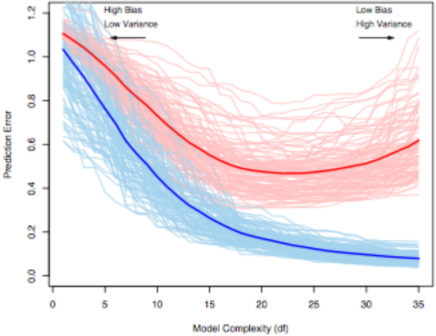


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$, *while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

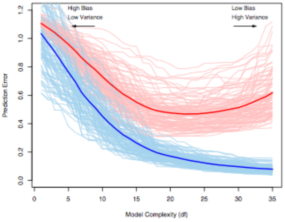


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

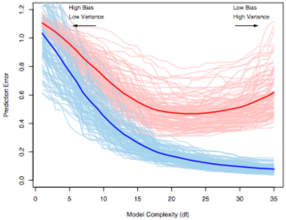


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

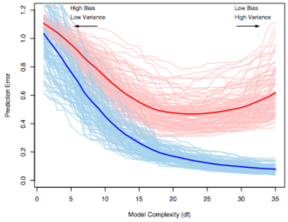


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

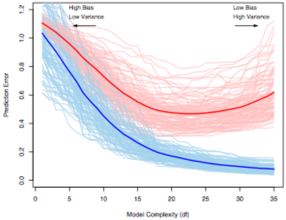


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

How Do We Build A Model?

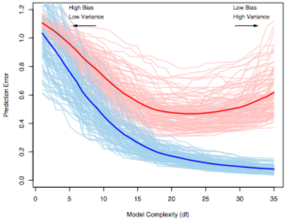


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

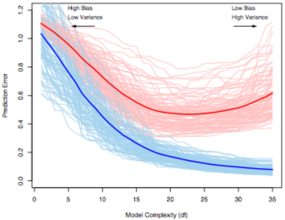


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$, *while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for* 100 *training sets of size* 50 *each,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

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, but optimistic$\rightsquigarrow$ improves with more parameters

Analytic Solutions

Approximate optimism and compensate in loss function.

Analytic Solutions

Approximate optimism and compensate in loss function.

Akaike Information Criterion (AIC).

Analytic Solutions

Approximate optimism and compensate in loss function.

Akaike Information Criterion (AIC).

As $N \rightarrow \infty$

Analytic Solutions

Approximate optimism and compensate in loss function.

Akaike Information Criterion (AIC).

As $N \rightarrow \infty$

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

Analytic Solutions

Approximate optimism and compensate in loss function.

Akaike Information Criterion (AIC).

As $N \to \infty$

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

Analytic Solutions

Approximate optimism and compensate in loss function.

Akaike Information Criterion (AIC).

As $N \rightarrow \infty$

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

where d are the number of parameters in the model

Analytic Solutions

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

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

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).

As $N \rightarrow \infty$

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

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).

As $N \rightarrow \infty$

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

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).
As $N \rightarrow \infty$

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

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]

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

Analytic Solutions

Bayesian Information Criterion (BIC) [Schwarz Criterion]

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

where d is again the effective number of parameters

Analytic Solutions

Bayesian Information Criterion (BIC) [Schwarz Criterion]

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

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]

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

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]

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

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

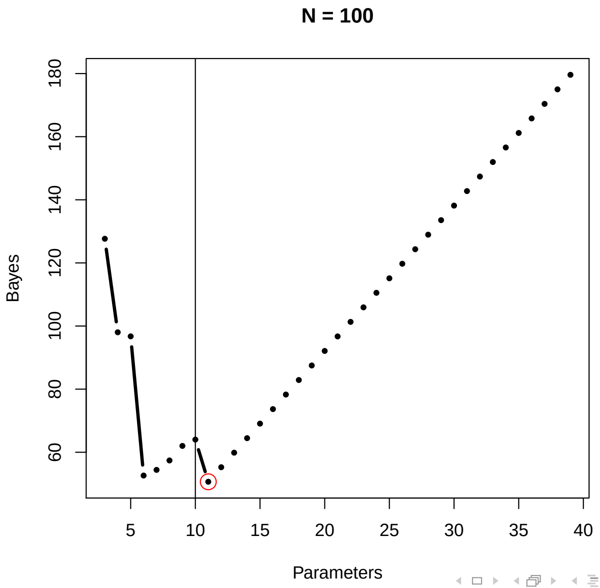
Bayesian Information Criterion (BIC) [Schwarz Criterion]

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

where d is again the effective number of parameters

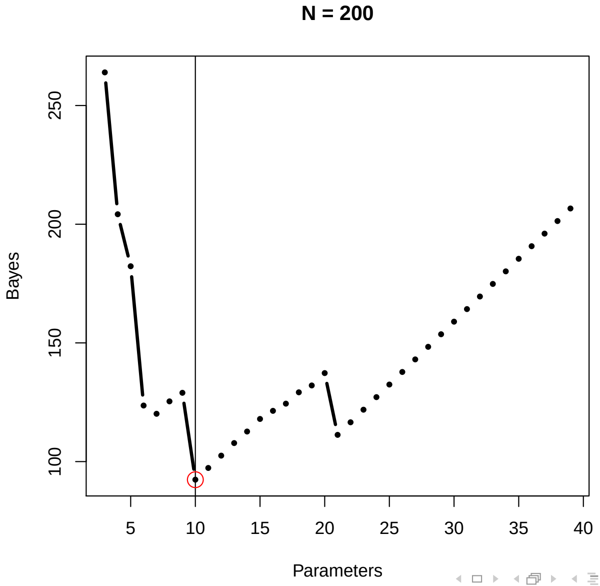
- Intuition: balances model fit with penalty for complexity
- Derived from Bayesian approach to model selection
- Approximation to Bayes' factor
- Penalizes more heavily than AIC

BIC or AIC?

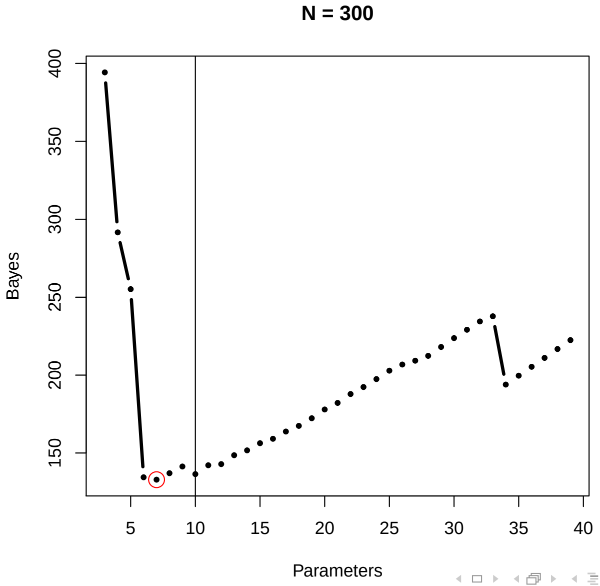


N = 100
Bayes
180
160
140
120
100
80
60
5
10
15
20
25
30
35
40
Parameters

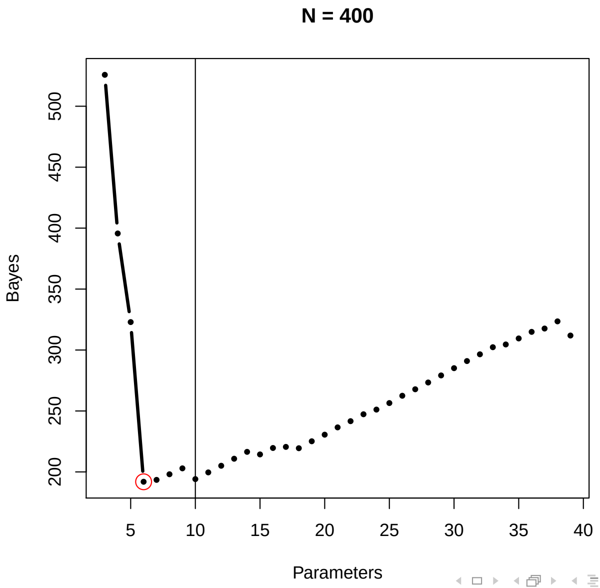
BIC or AIC?



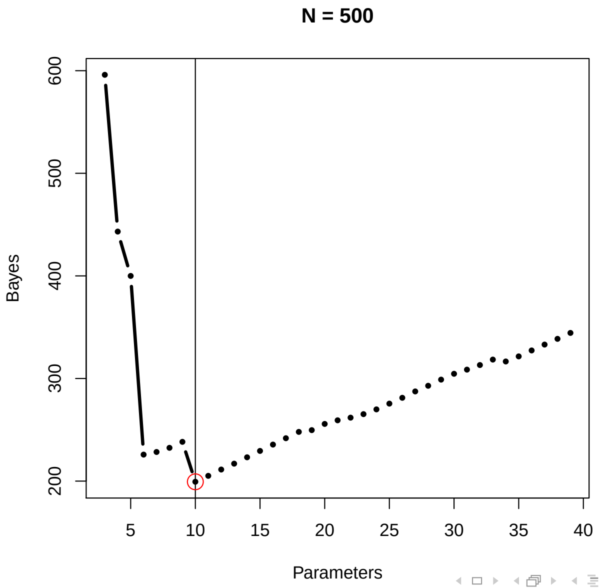
BIC or AIC?


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

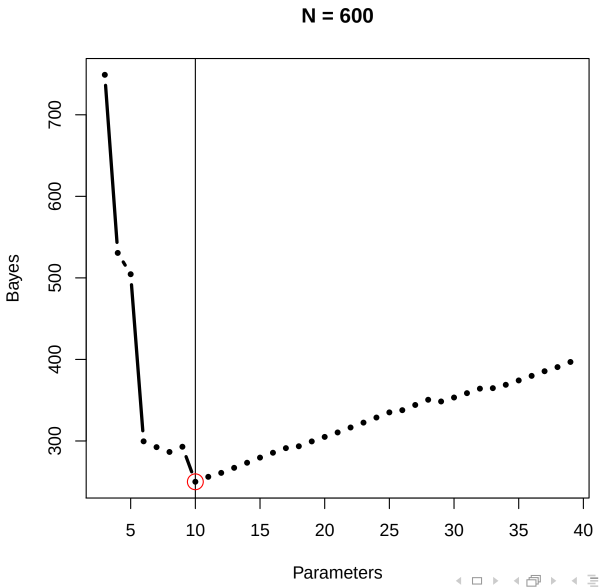
BIC or AIC?


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

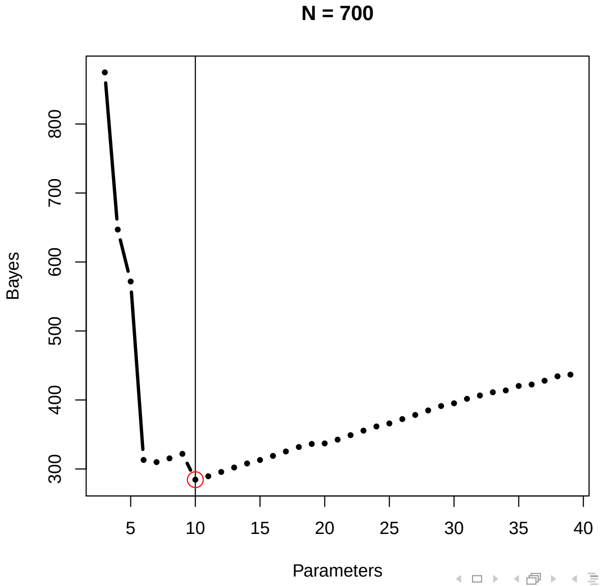
BIC or AIC?



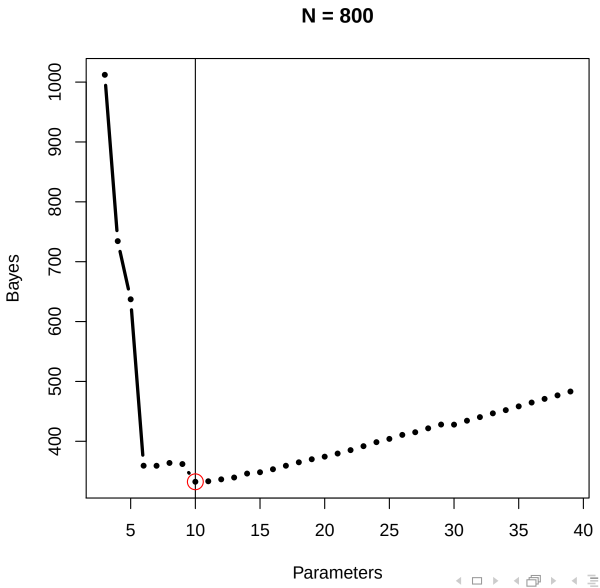
BIC or AIC?



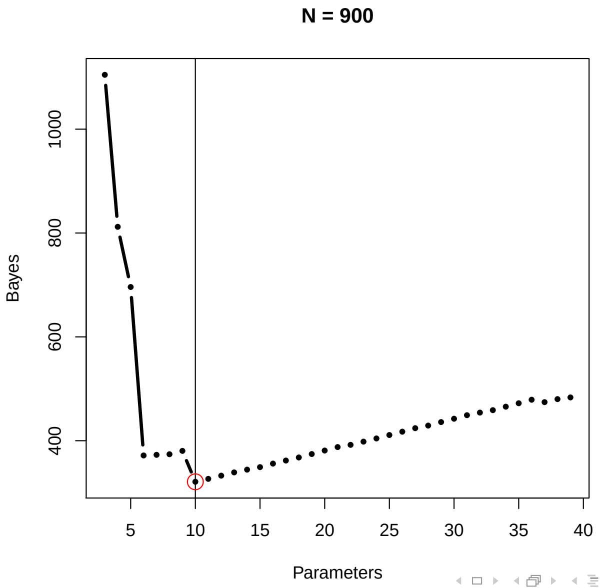
BIC or AIC?


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

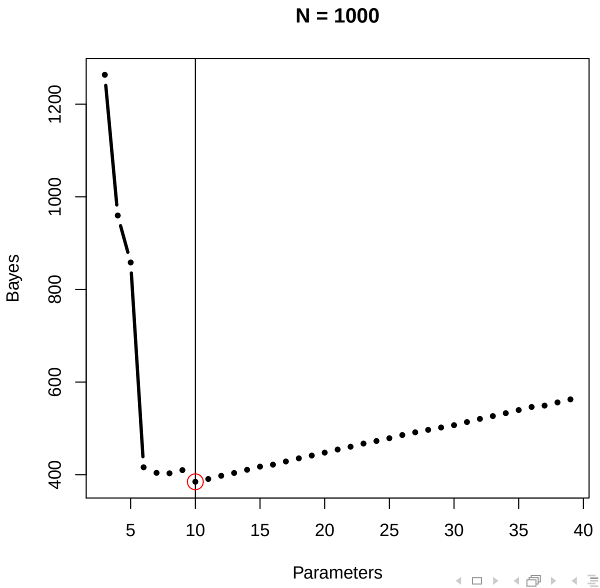
BIC or AIC?


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

BIC or AIC?

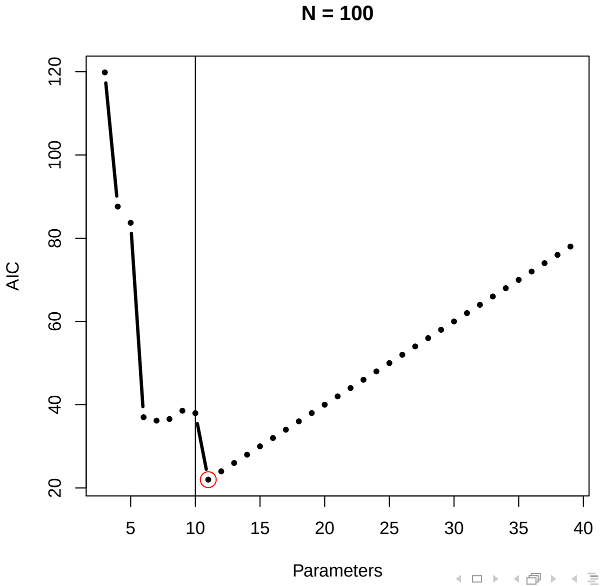


BIC or AIC?

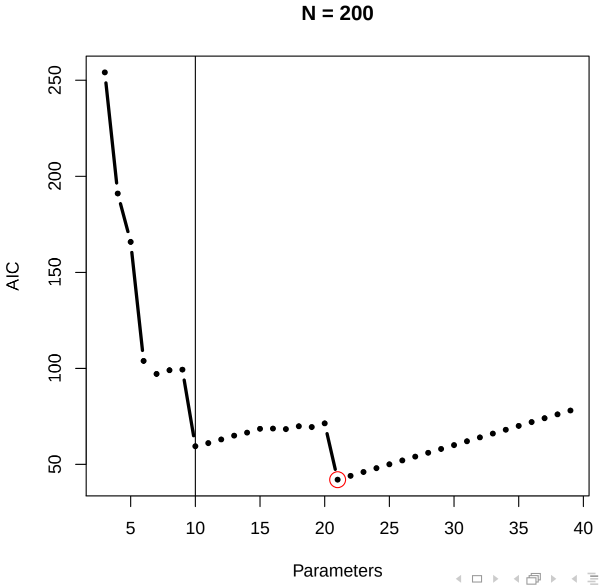


N = 1000
Bayes
1200
1000
800
600
400
5
10
15
20
25
30
35
40
Parameters

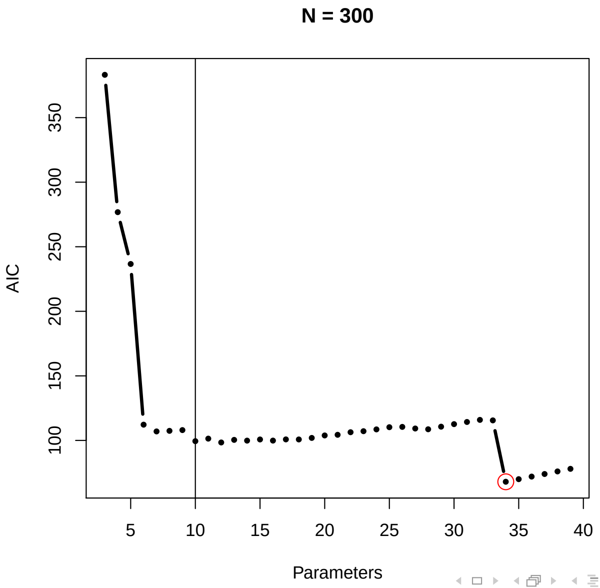
BIC or AIC?



BIC or AIC?

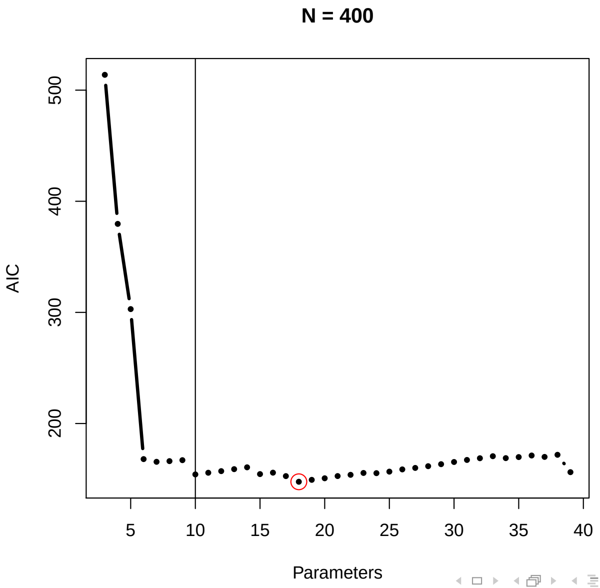


BIC or AIC?

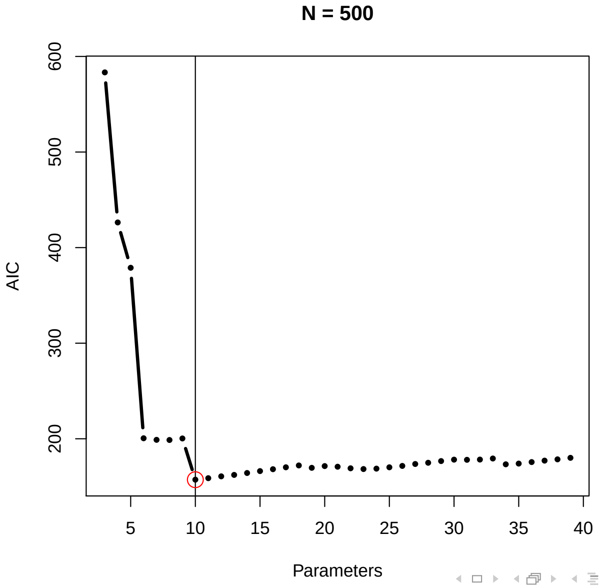


N = 300
AIC
350
300
250
200
150
100
5
10
15
20
25
30
35
40
Parameters

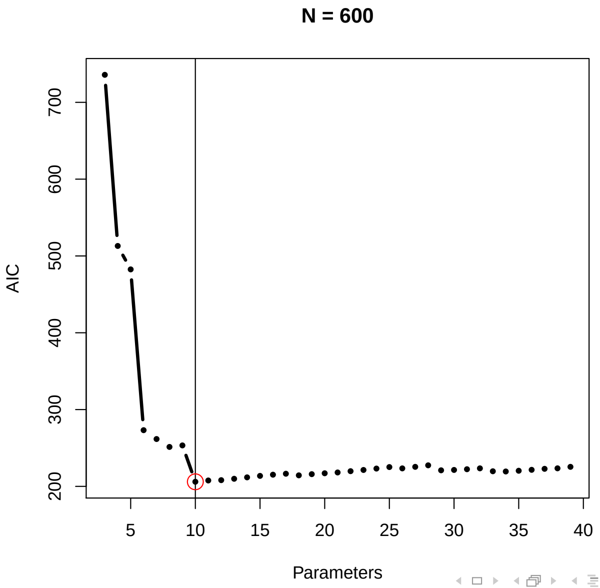
BIC or AIC?



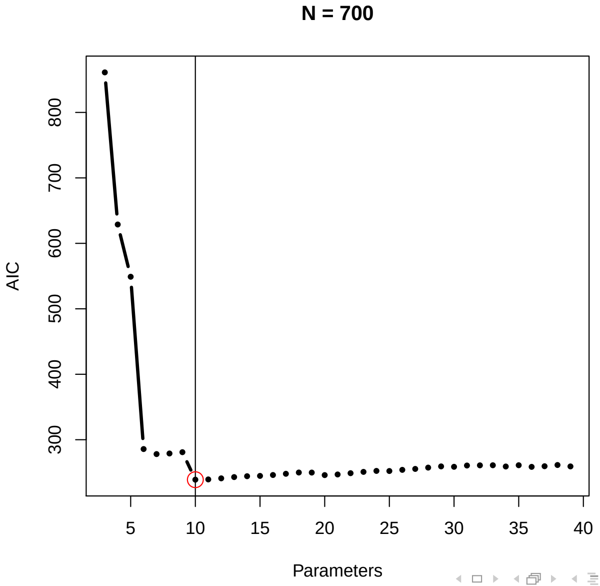
BIC or AIC?



BIC or AIC?

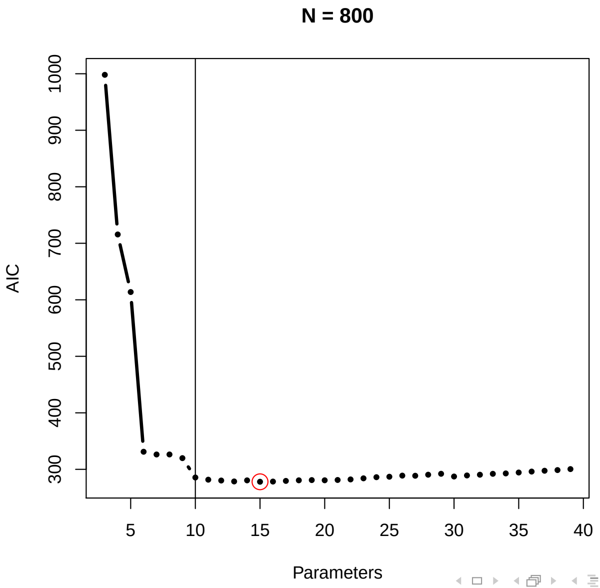

N = 600
AIC
700
600
500
400
300
200
5
10
15
20
25
30
35
40
Parameters

BIC or AIC?

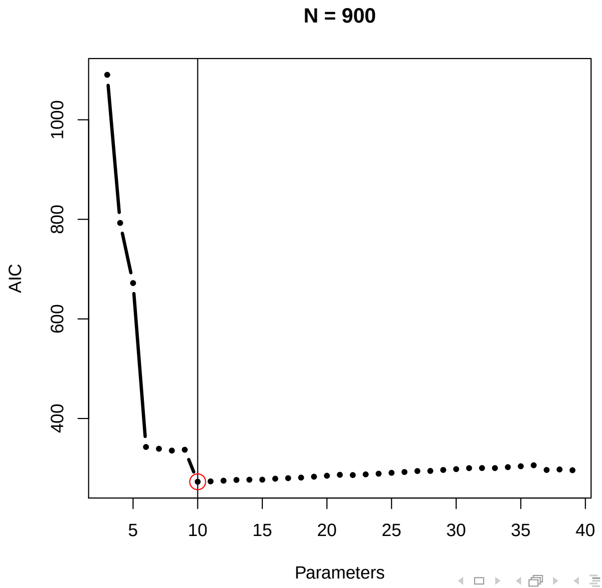


N = 700
AIC
800
700
600
500
400
300
5
10
15
20
25
30
35
40
Parameters

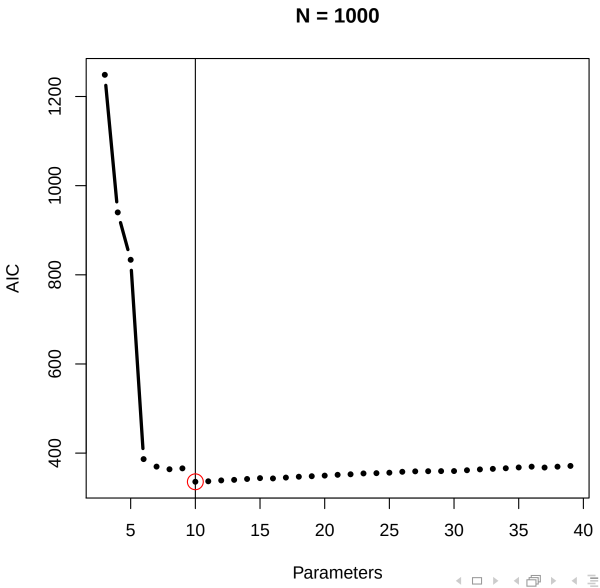
BIC or AIC?


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
  - As $N \to \infty$ will choose correct model with probability 1
  - Small samples$\rightsquigarrow$ overpenalize
- AIC
  - No asymptotic guarantees
  - 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

Recall Optimal division of data:

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

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

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

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

- Idea: use observations both in training and test sets

Cross-Validation: Some Intuition

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

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

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

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

Recall Optimal division of data:

- Train: build model
- Validation: assess model
- Test: classify remaining documents

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:

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

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
$\vdots$	$\vdots$	$\vdots$
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 $\hat{f}^{-K}(\boldsymbol{X}, \boldsymbol{\lambda})$

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

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

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

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 $\hat{f}^{-K}(\boldsymbol{X}, \boldsymbol{\lambda})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{X}, \boldsymbol{\lambda}))$
    - Mean square error, Absolute error, Prediction error, ...

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

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 $\hat{f}^{-K}(\boldsymbol{X}, \boldsymbol{\lambda})$
- Predict values for K^{th}
- Summarize performance with loss function: $L(\boldsymbol{Y}_i, \hat{f}^{-k}(\boldsymbol{X}, \boldsymbol{\lambda}))$
  - Mean square error, Absolute error, Prediction error, ...

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

  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

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:

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

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

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

Generalized Cross Validation and Ridge Regression

Why do we care?
Leave one out cross validation

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

$$= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{Y_i - f(\boldsymbol{X}, \boldsymbol{Y}, \lambda, \hat{\boldsymbol{\beta}})}{1 - H_{ii}}\right)^2$$

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

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:

$$\text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_i + \lambda}$$

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:

$$\text{Tr}(\boldsymbol{H}) \;=\; \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_i + \lambda}$$

where λ_i is the i^{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:

$$\text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_i + \lambda}$$

where λ_i is the i^{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:

$$\text{Tr}(\boldsymbol{H}) \;=\; \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_i + \lambda}$$

where λ_i is the i^{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:

$$\text{Tr}(\boldsymbol{H}) \;=\; \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_i + \lambda}$$

where λ_i is the i^{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:

$$\text{Tr}(\boldsymbol{H}) \quad = \quad \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_i + \lambda}$$

where λ_i is the i^{th} Eigenvalue from $\boldsymbol{\Sigma} = \boldsymbol{X}^{'}\boldsymbol{X}$ (!!!!!)

Define generalized cross validation:

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

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:

$$\text{Tr}(\boldsymbol{H}) = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda_i + \lambda}$$

where λ_i is the i^{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

Cross Validation

Use cross validation extensively:

1) Selecting tuning parameters
2) Learning weights in an ensemble
3) But it is no panacea:
   - Depends on K
   - Sampling$\rightsquigarrow$ maintain dependencies

Next week: Ensembles + Ideological scaling