Introduction to Machine Learning for Social Science

Class 7: LASSO Regression & Cross Validation

Rochelle Terman

Postdoctoral Fellow
Center for International Security Cooperation
Stanford University

January 30th, 2018

Questions?

Supervised Learning $\rightsquigarrow$ Text analysis

Goal: predict article from National desk (1) or other desk (0)
Method: LASSO Regression
Evaluation:

1) In Sample Fit
2) Out of Sample Fit

Key Terms:

- Overfitting
- Regularization
- LASSO
- Mean Squared Error (MSE)
- Bias-Variance Trade-off
- Cross validation

Key R Functions and Terms

- `glmnet`, `cv.glmnet`

LASSO Regression

LASSO Regression

LASSO ("least absolute shrinkage and selection operator"): a regularization procedure that shrinks regression coefficients toward zero.

LASSO Regression

LASSO ("least absolute shrinkage and selection operator"): a regularization procedure that shrinks regression coefficients toward zero.

Document i, $(i = 1, \ldots, N)$

LASSO Regression

LASSO ("least absolute shrinkage and selection operator"): a regularization procedure that shrinks regression coefficients toward zero.

Document i, ($i = 1, \ldots, N$)
Count vector $\boldsymbol{x}_i$

LASSO Regression

LASSO ("least absolute shrinkage and selection operator"): a regularization procedure that shrinks regression coefficients toward zero.

Document i, $(i = 1, \ldots, N)$
Count vector $\boldsymbol{x}_i$
Labels Y_i

LASSO Regression

LASSO ("least absolute shrinkage and selection operator"): a regularization procedure that shrinks regression coefficients toward zero.

Document i, $(i = 1, \ldots, N)$
Count vector $\boldsymbol{x}_i$
Labels Y_i
Linear regression: Choose $\beta's$ to minimize sum of squared residuals

$$\beta_{\text{OLS}} \quad = \quad \text{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{N} (Y_i - \boldsymbol{\beta} \cdot \boldsymbol{x}_i)^2$$

LASSO Regression

LASSO ("least absolute shrinkage and selection operator"): a regularization procedure that shrinks regression coefficients toward zero.

Document i, $(i = 1, \ldots, N)$
Count vector $\boldsymbol{x}_i$
Labels Y_i
LASSO Regression: Choose $\beta's$ to minimize sum of squared residuals and penalty on size of coefficients

LASSO Regression

LASSO ("least absolute shrinkage and selection operator"): a regularization procedure that shrinks regression coefficients toward zero.

Document i, $(i = 1, \ldots, N)$
Count vector $\boldsymbol{x}_i$
Labels Y_i
LASSO Regression: Choose $\boldsymbol{\beta}'s$ to minimize sum of squared residuals and penalty on size of coefficients

$$\boldsymbol{\beta}_{\text{LASSO}} = \text{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{N} (Y_i - \boldsymbol{\beta} \cdot \boldsymbol{x}_i)^2 + \lambda \underbrace{\sum_{p=1}^{P} |\beta_p|}_{\text{penalty}}$$

LASSO Regression

LASSO ("least absolute shrinkage and selection operator"): a regularization procedure that shrinks regression coefficients toward zero.

Document i, $(i = 1, \ldots, N)$
Count vector $\boldsymbol{x}_i$
Labels Y_i
LASSO Regression: Choose $\boldsymbol{\beta}'s$ to minimize sum of squared residuals and penalty on size of coefficients

$$\boldsymbol{\beta}_{\text{LASSO}} = \text{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{N} (Y_i - \boldsymbol{\beta} \cdot \boldsymbol{x}_i)^2 + \lambda \underbrace{\sum_{p=1}^{P} |\beta_{\boldsymbol{p}}|}_{\text{penalty}}$$

What does λ do?

How Do We Choose λ?

Best $\rightsquigarrow$ best performing model

How Do We Choose λ?

Best $\rightsquigarrow$ best performing model $\rightsquigarrow$ lowest mean squared error (MSE).

How Do We Choose λ?

Best $\rightsquigarrow$ best performing model $\rightsquigarrow$ lowest mean squared error (MSE).

Mean squared error (MSE): performance metric used to evaluate between competing models.

How Do We Choose λ?

Best $\rightsquigarrow$ best performing model $\rightsquigarrow$ lowest mean squared error (MSE).

Mean squared error (MSE): performance metric used to evaluate between competing models.

Define:

$$
\begin{aligned}
\widehat{\boldsymbol{\beta}}^{\lambda} &= \text{Coefficients at } \lambda \\
\widehat{p}_{i,\lambda} &= \Pr(Y_i = 1 | \boldsymbol{X}_i, \widehat{\boldsymbol{\beta}}^{\lambda})
\end{aligned}
$$

How Do We Choose λ?

Best $\rightsquigarrow$ best performing model $\rightsquigarrow$ lowest mean squared error (MSE).

Mean squared error (MSE): performance metric used to evaluate between competing models.

Define:

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}^{\lambda} &= \text{Coefficients at } \lambda \\
\widehat{p}_{i,\lambda} &= \Pr(Y_i = 1 | \boldsymbol{X}_i, \widehat{\boldsymbol{\beta}}^{\lambda}) \\
\text{MSE} &= \frac{\sum_{i=1}^{N} (Y_i - \widehat{p}_{i,\lambda})^2}{N}
\end{aligned}$$

Loss Function

Goal: Find λ that minimizes MSE (loss function).

Loss Function

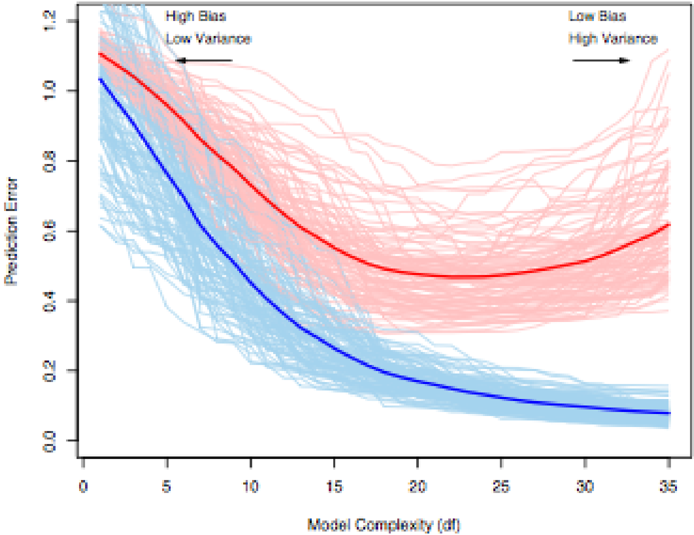
Goal: Find λ that minimizes MSE (loss function).

- Optimize in-sample fit?
- Optimize out-of-sample fit?

In-Sample Fit

In sample fit is greedy: adding more variables will always improve fit

Bad way to choose: in sample fit!



Bias-Variance Tradeoff

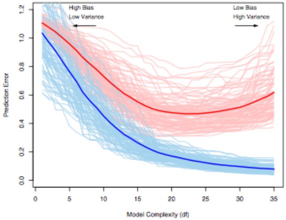


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$, *while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each,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

Bias-Variance Tradeoff

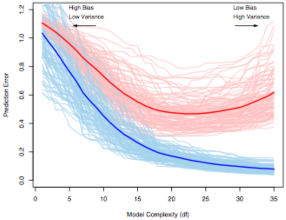


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$, *while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

Bias-Variance Tradeoff

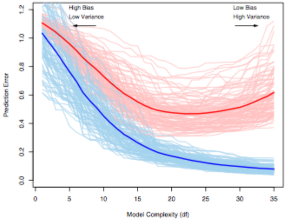


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

Bias-Variance Tradeoff

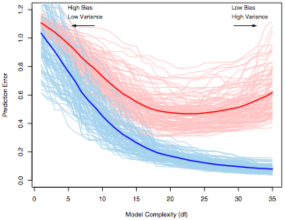


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$*, while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, 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 (both training and test sets).
- Reduces error in both training and test set.
- Additional model complexity: idiosyncratic features of the training set.

Bias-Variance Tradeoff

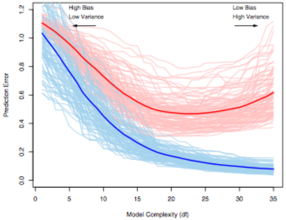


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\text{err}}$*, while the light red curves show the conditional test error* $\text{Err}_{\mathcal{T}}$ *for 100 training sets of size 50 each, as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

Bias-Variance Decomposition

$$\text{total error} = \text{Variance(Predictions)} + \text{Bias(Predictions)}$$

Bias-Variance Decomposition

$$\text{total error} = \text{Variance(Predictions)} + \text{Bias(Predictions)}$$

- Bias: error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

Bias-Variance Decomposition

$$\text{total error} = \text{Variance}(\text{Predictions}) + \text{Bias}(\text{Predictions})$$

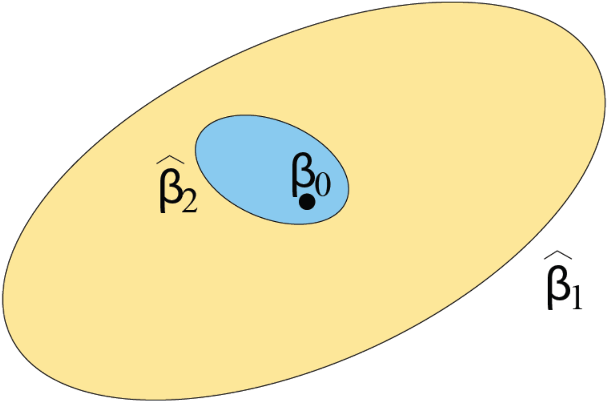
- Bias: error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

- Variance: error from sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (overfitting).

Bias-Variance Decomposition

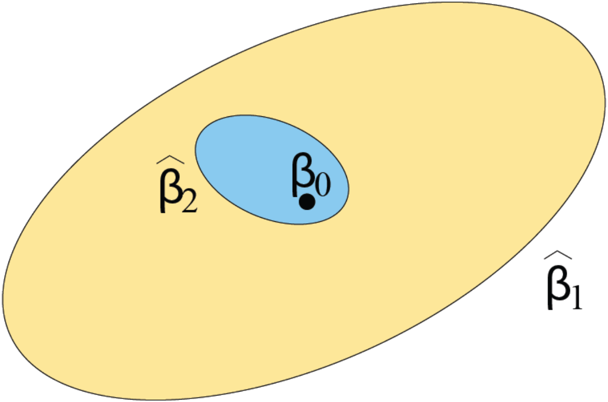
Bias-Variance Tradeoff

Introducing bias can help minimize variance, leading to better performance.



Bias-Variance Tradeoff

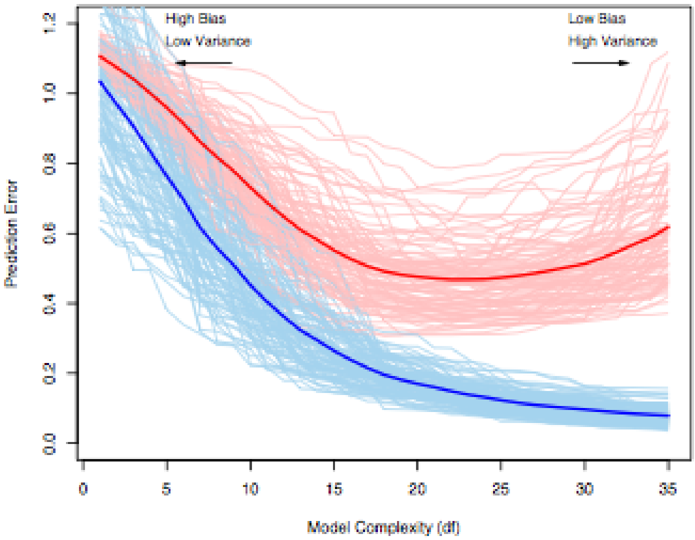
Introducing bias can help minimize variance, leading to better performance.



This is what lasso does!

The Goal:

Find the sweet spot between bias and variance using out-of-sample data.



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

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
⋮	⋮	⋮
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.

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.
- 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.
- Predict values for K^{th}
- Summarize performance with loss function:

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.
- Predict values for K^{th}
- Summarize performance with loss function:
  - 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.
- Predict values for K^{th}
- Summarize performance with loss function:
    - Mean square error, Absolute error, Prediction error, ...
- Final choice: model with optimal 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 (LOOCV)

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 (LOOCV)

Considerations:

- How sensitive are inferences to number of coded documents? (ISL, pg 181-184)
    - 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 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 (LOOCV)

Considerations:

- How sensitive are inferences to number of coded documents? (ISL, pg 181-184)
  - 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

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 (LOOCV)

Considerations:

- How sensitive are inferences to number of coded documents? (ISL, pg 181-184)
  - 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
- Bias-Variance in Cross Validation:
  - LOOCV: Less bias
  - K = 5 or 10: Less Variance
  - In brief: k = 5 or 10 shown to be "sweet spot", yielding low test error estimates

Key Terms:

- Overfitting
- Regularization
- LASSO
- Mean Squared Error (MSE)
- Bias-Variance Trade-off
- Cross validation

Key R Functions and Terms

- `glmnet`, `cv.glmnet`