Introduction to Machine Learning and Cross-Validation

Jonathan Hersh[1]

February 27, 2019

Plan

Machine learning versus econometrics

Machine Learning

- Developed to solve problems in computer science

Econometrics

- Developed to solve problems in economics

Machine learning versus econometrics

Machine Learning

- Developed to solve problems in computer science
- Prediction/classification ✓

Econometrics

- Developed to solve problems in economics
- Explicitly testing a theory

Machine learning versus econometrics

Machine Learning

- Developed to solve problems in computer science
- Prediction/classification ✓
- Want: goodness of fit

Econometrics

- Developed to solve problems in economics
- Explicitly testing a theory
- "Statistical significance" more important than model fit

Machine learning versus econometrics

Machine Learning

- Developed to solve problems in computer science
- Prediction/classification ✓
- Want: goodness of fit
- Huge datasets (many terabytes), large # variables (1000s)

Econometrics

- Developed to solve problems in economics
- Explicitly testing a theory
- "Statistical significance" more important than model fit
- Small datasets, few variables

Machine learning versus econometrics

Machine Learning

- Developed to solve problems in computer science
- Prediction/classification ✓
- Want: goodness of fit
- Huge datasets (many terabytes), large # variables (1000s)
- Whatever works

Econometrics

- Developed to solve problems in economics
- Explicitly testing a theory
- "Statistical significance" more important than model fit
- Small datasets, few variables
- "It works in practice, but what about in theory?"

Machine learning versus econometrics

Machine Learning

- Developed to solve problems in computer science
- Prediction/classification ✓
- Want: goodness of fit
- Huge datasets (many terabytes), large # variables (1000s)
- Whatever works

Econometrics

- Developed to solve problems in economics
- Explicitly testing a theory
- "Statistical significance" more important than model fit
- Small datasets, few variables
- "It works in practice, but what about in theory?"

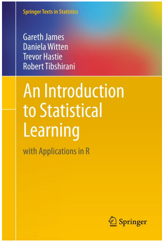
- Can we utilize some of this machinery to solve problems in development economics?

Applications of Machine Learning in economics (due to Sendhil Mullainathan)

1. **New data**
2. **Predictions for policy**
3. **Better econometrics**

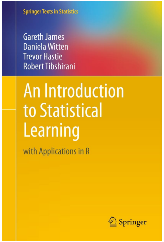
- See Machine learning: an applied econometric approach, JEP
- See *Athey "Beyond Prediction: Using Big Data for Policy Problems" (2017) Science*
- See Hersh, Jonathan, and Matthew Harding. "Big Data in economics." IZA World of Labor (2018)

This course



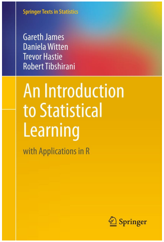
- **Primer in statistical learning theory, which grew out of statistics**
- How does this differ from ML? Machine learning places more emphasis on large scale applications and prediction accuracy. Statistical learning covers

This course



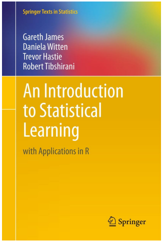
- **Primer in statistical learning theory, which grew out of statistics**
- How does this differ from ML? Machine learning places more emphasis on large scale applications and prediction accuracy. Statistical learning covers
- There is much overlap and cross-fertilization

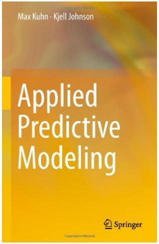
This course



- **Primer in statistical learning theory, which grew out of statistics**
- How does this differ from ML? Machine learning places more emphasis on large scale applications and prediction accuracy. Statistical learning covers
- There is much overlap and cross-fertilization
- Very little coding, but example code provided: jonathan-hersh.com/machinelearningdev

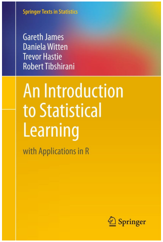
Topics covered

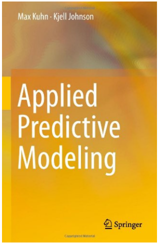




1. **Cross-validation** [Chapter 2]

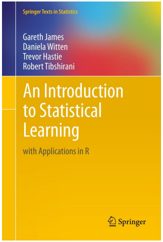
Topics covered

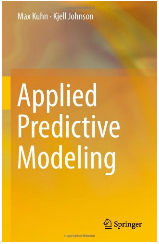




1. **Cross-validation** [Chapter 2]
2. **Shrinkage methods** (Ridge and LASSO) [Chapter 6]

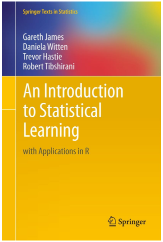
Topics covered

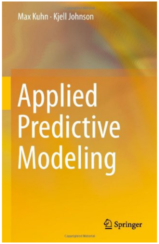




1. **Cross-validation** [Chapter 2]
2. **Shrinkage methods** (Ridge and LASSO) [Chapter 6]
3. **Classification** [Chapter 4, APM Chapter 11-12]

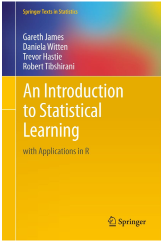
Topics covered

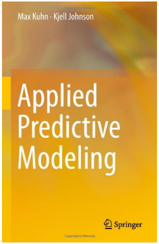




1. **Cross-validation** [Chapter 2]
2. **Shrinkage methods** (Ridge and LASSO) [Chapter 6]
3. **Classification** [Chapter 4, APM Chapter 11-12]
4. **Tree-based methods** (Decision trees, bagging, random forest boosting) [Chapter 8]

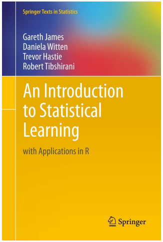
Topics covered

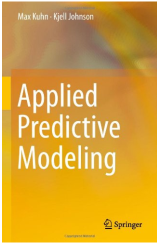




1. **Cross-validation** [Chapter 2]
2. **Shrinkage methods** (Ridge and LASSO) [Chapter 6]
3. **Classification** [Chapter 4, APM Chapter 11-12]
4. **Tree-based methods** (Decision trees, bagging, random forest boosting) [Chapter 8]
5. **Unsupervised learning** (PCA, k-means clustering, hierarchical clustering) [Chapter 10]

Topics covered





1. **Cross-validation** [Chapter 2]
2. **Shrinkage methods** (Ridge and LASSO) [Chapter 6]
3. **Classification** [Chapter 4, APM Chapter 11-12]
4. **Tree-based methods** (Decision trees, bagging, random forest boosting) [Chapter 8]
5. **Unsupervised learning** (PCA, k-means clustering, hierarchical clustering) [Chapter 10]
6. **Caret** Automated Machine Learning [APM]

Plan

Supervised vs. unsupervised learning

Def: Supervised learning

for every x_i we also observe a response y_i

Supervised vs. unsupervised learning

Def: Supervised learning

for every x_i we also observe a response y_i

Ex: Estimating housing values by OLS or random forest;

Supervised vs. unsupervised learning

Def: Supervised learning

for every x_i we also observe a response y_i

Ex: Estimating housing values by OLS or random forest;

Def: Unsupervised learning

for each observation we **only** observe x_i, **but do not observe** y_i

Supervised vs. unsupervised learning

Def: Supervised learning

for every x_i we also observe a response y_i

Ex: Estimating housing values by OLS or random forest;

Def: Unsupervised learning

for each observation we **only** observe x_i, **but do not observe** y_i

Ex: Clustering customers into segments; using principle component analysis for dimension reduction

Test, Training, and Validation Set

- **Training set:** (observation-wise) subset of data used to develop models

Test, Training, and Validation Set

- **Training set:** (observation-wise) subset of data used to develop models
- **Validation set:** subset of data used during intermediate stages to “tune” model parameters

Test, Training, and Validation Set

- **Training set:** (observation-wise) subset of data used to develop models
- **Validation set:** subset of data used during intermediate stages to "tune" model parameters
- **Test set:** subset of data (used sparingly) to approximate out of sample fit

Assessing model accuracy

Mean squared error

measures how well model predictions match observed data

$$MSE\left(\hat{f}(x)\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\underbrace{y_i}_{\textbf{data}} - \underbrace{\hat{f}(x_i)}_{\textbf{model}}\right)^2$$

Assessing model accuracy

Mean squared error

measures how well model predictions match observed data

$$MSE\left(\hat{f}(x)\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\underbrace{y_i}_{\textbf{data}} - \underbrace{\hat{f}(x_i)}_{\textbf{model}}\right)^2$$

- **Training MSE vs Test MSE:** good in-sample fit (low training MSE) can often obscure poor out of sample fit (high test MSE)

Plan

Quick example on out-of-sample fit

- **Let's compare three estimators to see how estimator complexity affects out of sample fit**

1. f_1 = linear regression (in orange)
2. f_2 = third order polynomial (in black)
3. f_3 = very flexible smoothing spline (in green)

Case 1: True $f(x)$ slightly complicated

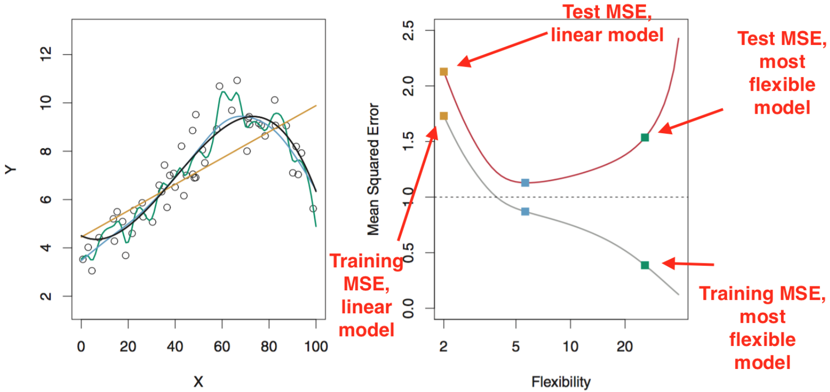


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand*

Case 2: True $f(x)$ not complicated at all

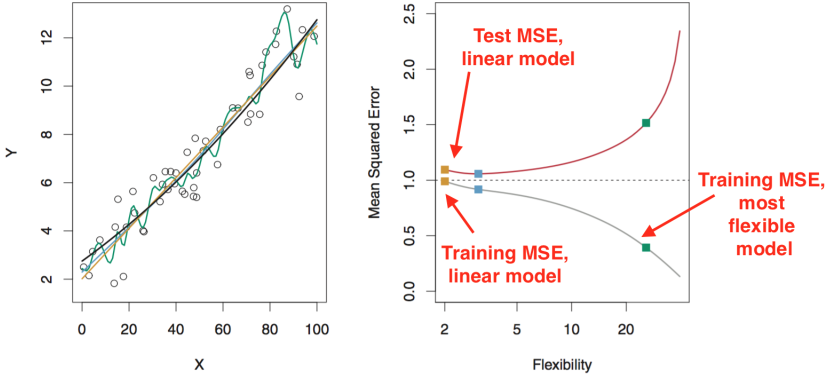


FIGURE 2.10. *Details are as in Figure 2.9, using a different true f that is much closer to linear. In this setting, linear regression provides a very good fit to the data.*

Case 2: True $f(x)$ not complicated at all

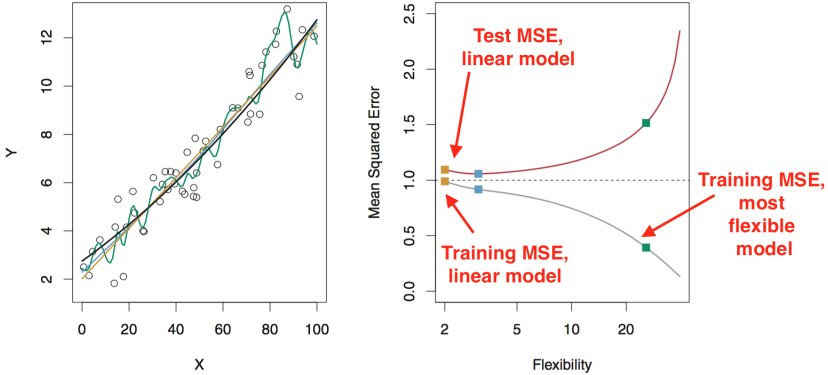


FIGURE 2.10. *Details are as in Figure 2.9, using a different true f that is much closer to linear. In this setting, linear regression provides a very good fit to the data.*

Case 3: True $f(x)$ very complicated

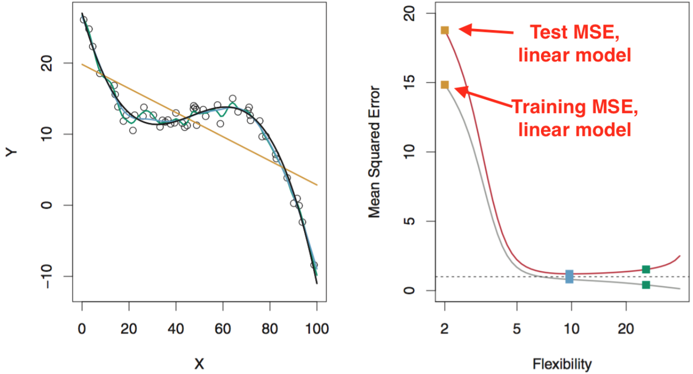
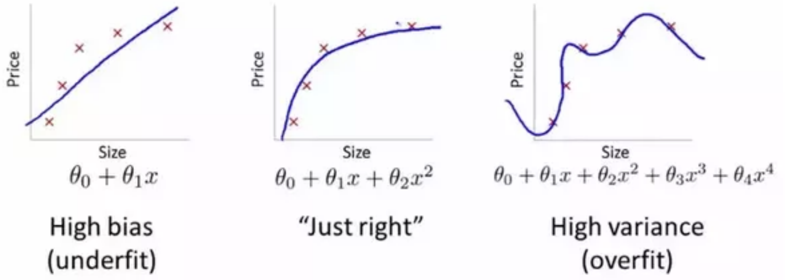


FIGURE 2.11. *Details are as in Figure 2.9, using a different f that is far from linear. In this setting, linear regression provides a very poor fit to the data.*

How to select right model complexity?



Generalizing this problem: Bias-Variance tradeoff

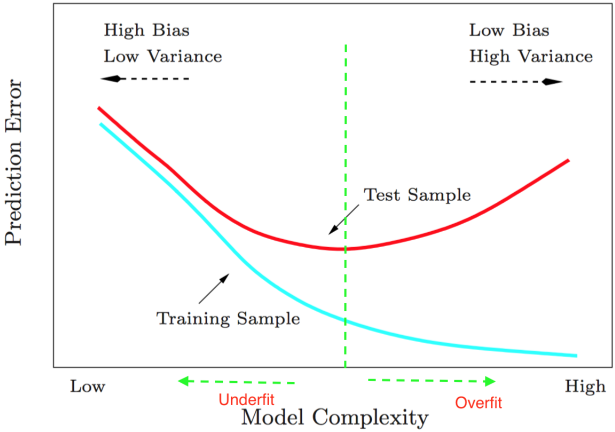


FIGURE 2.11. *Test and training error as a function of model complexity.*

Bias Variance Tradeoff in Math

Prediction Error: $$\mathbb{E}\left[\left(\underbrace{y_i}_{\text{data}} - \underbrace{\hat{f}\left(x_i\right)}_{\text{model}}\right)^2\right]$$

Bias Variance Tradeoff in Math

Prediction Error: $$\mathbb{E}\left[\left(\underbrace{y_i}_{\text{data}} - \underbrace{\hat{f}\left(x_i\right)}_{\text{model}}\right)^2\right]$$

$$\mathbb{E}\left[\left(y - \hat{f}\right)^2\right] = \mathbb{E}\left[y^2 + \hat{f}^2 - 2y\hat{f}\right] \quad \text{(expanding terms)}$$

Bias Variance Tradeoff in Math

$$\textbf{Prediction Error:} \qquad \mathbb{E}\left[\left(\underbrace{y_i}_{\text{data}} - \underbrace{\hat{f}\left(x_i\right)}_{\text{model}}\right)^2\right]$$

$$\mathbb{E}\left[\left(y - \hat{f}\right)^2\right] = \mathbb{E}\left[y^2 + \hat{f}^2 - 2y\hat{f}\right] \qquad \text{(expanding terms)}$$

$$= \underbrace{\mathbb{E}\left[y^2\right]}_{\equiv Var(y)+\mathbb{E}[y]^2} + \underbrace{\mathbb{E}\left[\hat{f}^2\right]}_{\equiv Var\left(\hat{f}\right)+\mathbb{E}\left[\hat{f}\right]^2} - \mathbb{E}\left[2y\hat{f}\right]$$

Bias Variance Tradeoff in Math

Prediction Error: $$\mathbb{E}\left[\left(\underbrace{y_i}_{\text{data}} - \underbrace{\hat{f}(x_i)}_{\text{model}}\right)^2\right]$$

$$\mathbb{E}\left[\left(y - \hat{f}\right)^2\right] = \mathbb{E}\left[y^2 + \hat{f}^2 - 2y\hat{f}\right] \quad \text{(expanding terms)}$$

$$= \underbrace{\mathbb{E}\left[y^2\right]}_{\equiv Var(y) + \mathbb{E}[y]^2} + \underbrace{\mathbb{E}\left[\hat{f}^2\right]}_{\equiv Var\left(\hat{f}\right) + \mathbb{E}\left[\hat{f}\right]^2} - \mathbb{E}\left[2y\hat{f}\right]$$

$$= Var(y) + \underbrace{\mathbb{E}[y]^2}_{y \equiv f(x) + \varepsilon, \mathbb{E}[\varepsilon] = 0} + Var\left(\hat{f}\right) + \mathbb{E}\left[\hat{f}\right]^2 - \mathbb{E}\left[2y\hat{f}\right] \quad \text{(def'n var)}$$

Bias Variance Tradeoff in Math

Prediction Error: $$\mathbb{E}\left[\left(\underbrace{y_i}_{\text{data}} - \underbrace{\hat{f}(x_i)}_{\text{model}}\right)^2\right]$$

$$\mathbb{E}\left[\left(y - \hat{f}\right)^2\right] = \mathbb{E}\left[y^2 + \hat{f}^2 - 2y\hat{f}\right] \quad \text{(expanding terms)}$$

$$= \underbrace{\mathbb{E}\left[y^2\right]}_{\equiv Var(y) + \mathbb{E}[y]^2} + \underbrace{\mathbb{E}\left[\hat{f}^2\right]}_{\equiv Var\left(\hat{f}\right) + \mathbb{E}\left[\hat{f}\right]^2} - \mathbb{E}\left[2y\hat{f}\right]$$

$$= Var(y) + \underbrace{\mathbb{E}[y]^2}_{y \equiv f(x) + \varepsilon, \mathbb{E}[\varepsilon] = 0} + Var\left(\hat{f}\right) + \mathbb{E}\left[\hat{f}\right]^2 - \mathbb{E}\left[2y\hat{f}\right] \quad \text{(def'n var)}$$

$$= Var(y) + Var\left(\hat{f}\right) + \left(\mathbb{E}\left[\hat{f}\right]^2 - \mathbb{E}\left[2y\hat{f}\right] + f^2\right) \quad \text{(def'n y)}$$

Bias Variance Tradeoff in Math

$$= \underbrace{Var\left(y\right)}_{=\mathbb{E}\left[((y-\mathbb{E}[y])^2\right]=\mathbb{E}\left[((y-f)^2\right]=\mathbb{E}[\varepsilon^2]=Var(\varepsilon)=\sigma_\varepsilon^2} + Var\left(\hat{f}\right) + \left(f - \mathbb{E}\left[\hat{f}\right]\right)^2$$

Bias Variance Tradeoff in Math

$$= \underbrace{Var(y)}_{=\mathbb{E}\left[((y-\mathbb{E}[y])^2\right]=\mathbb{E}\left[((y-f)^2\right]=\mathbb{E}[\varepsilon^2]=Var(\varepsilon)=\sigma_\varepsilon^2} + Var\left(\hat{f}\right) + \left(f - \mathbb{E}\left[\hat{f}\right]\right)^2$$

Prediction Error

$$\mathbb{E}\left[\left(y_i - \hat{f}\right)\right] = \underbrace{\sigma_\varepsilon^2}_{\text{irreducible error}} + \underbrace{Var\left(\hat{f}\right)}_{\text{variance}} + \underbrace{\left(\left[f - \mathbb{E}\left[\hat{f}\right]\right]\right)^2}_{\text{bias}}$$

Bias Variance Tradeoff in Math

$$= \underbrace{Var\left(y\right)}_{=\mathbb{E}\left[((y-\mathbb{E}[y])^2\right]=\mathbb{E}\left[((y-f)^2\right]=\mathbb{E}[\varepsilon^2]=Var(\varepsilon)=\sigma_\varepsilon^2} + Var\left(\hat{f}\right) + \left(f - \mathbb{E}\left[\hat{f}\right]\right)^2$$

Prediction Error

$$\mathbb{E}\left[\left(y_i - \hat{f}\right)\right] = \underbrace{\sigma_\varepsilon^2}_{\text{irreducible error}} + \underbrace{Var\left(\hat{f}\right)}_{\text{variance}} + \underbrace{\left(\left[f - \mathbb{E}\left[\hat{f}\right]\right]\right)^2}_{\text{bias}}$$

- Bias is minimized when $f = \mathbb{E}\left[\hat{f}\right]$
- **But total error (variance + bias) may be minimized by some other $\hat{f}$**

Bias Variance Tradeoff in Math

$$= \underbrace{Var(y)}_{=\mathbb{E}\left[((y-\mathbb{E}[y])^2\right]=\mathbb{E}\left[((y-f)^2\right]=\mathbb{E}[\varepsilon^2]=Var(\varepsilon)=\sigma_\varepsilon^2} + Var\left(\hat{f}\right) + \left(f - \mathbb{E}\left[\hat{f}\right]\right)^2$$

Prediction Error

$$\mathbb{E}\left[\left(y_i - \hat{f}\right)\right] = \underbrace{\sigma_\varepsilon^2}_{\text{irreducible error}} + \underbrace{Var\left(\hat{f}\right)}_{\text{variance}} + \underbrace{\left(\left[f - \mathbb{E}\left[\hat{f}\right]\right]\right)^2}_{\text{bias}}$$

- Bias is minimized when $f = \mathbb{E}\left[\hat{f}\right]$
- **But total error (variance + bias) may be minimized by some other $\hat{f}$**
- $\hat{f}(x)$ with smaller variance $\Rightarrow$ fewer variables, smaller magnitude coefficients

Plan

Cross-Validation

- Cross-validation is a tool to approximate out of sample fit

Cross-Validation

- **Cross-validation is a tool to approximate out of sample fit**
- In machine learning, many models have parameters that must be "tuned"
- **We adjust these parameters using using cross-validation**

Cross-Validation

- **Cross-validation is a tool to approximate out of sample fit**
- In machine learning, many models have parameters that must be "tuned"
- **We adjust these parameters using using cross-validation**
- Also useful to select between large classes of models
  - e.g random forest vs lasso

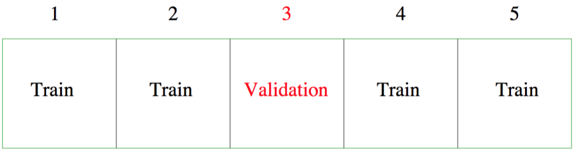
K-fold Cross-Validation (CV)

1	2	3	4	5
Train	Train	Validation	Train	Train

K-fold CV Algorithm

1. Randomly divide the data into K equal sized parts or “folds”.

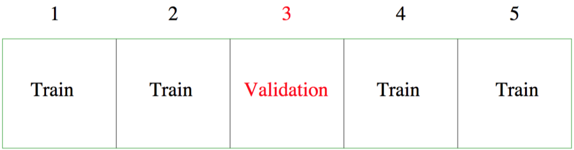
K-fold Cross-Validation (CV)



K-fold CV Algorithm

1. Randomly divide the data into K equal sized parts or “folds”.
2. Leave out part k, fit the model to the other $K-1$ parts.

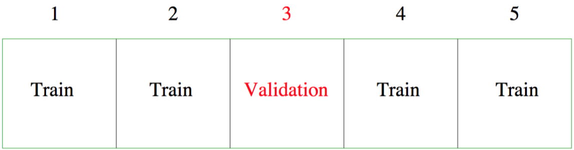
K-fold Cross-Validation (CV)



K-fold CV Algorithm

1. Randomly divide the data into K equal sized parts or “folds”.
2. Leave out part k, fit the model to the other $K - 1$ parts.
3. Use fitted model to obtain predictions for left-out k-th part

K-fold Cross-Validation (CV)



K-fold CV Algorithm

1. Randomly divide the data into K equal sized parts or "folds".
2. Leave out part k, fit the model to the other $K-1$ parts.
3. Use fitted model to obtain predictions for left-out k-th part
4. Repeat until $k = 1, \ldots, K$ and combine results

K-Fold CV Example 2

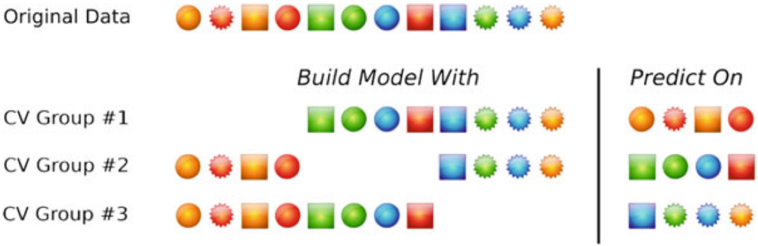


Fig. 4.6: A schematic of threefold cross-validation. Twelve training set samples are represented as symbols and are allocated to three groups. These groups are left out in turn as models are fit. Performance estimates, such as the error rate or R^2 are calculated from each set of held-out samples. The average of the three performance estimates would be the cross-validation estimate of model performance. In practice, the number of samples in the held-out subsets can vary but are roughly equal size

Details of K-Fold CV

- Let n_k be the number of test observations in fold k, where $n_k = N/K$
- Cross-Validation Error for fold k:

$$CV_{\{k\}} = \sum_{k=1}^{K} \frac{n_k}{N} MSE_k$$

where $MSE_k = \sum_{i \in C_k} (y_i - \hat{y})^2 / n_k$ is the mean squared error of fold k

Details of K-Fold CV

- Let n_k be the number of test observations in fold k, where $n_k = N/K$
- Cross-Validation Error for fold k:

$$CV_{\{k\}} = \sum_{k=1}^{K} \frac{n_k}{N} MSE_k$$

where $MSE_k = \sum_{i \in C_k} (y_i - \hat{y})^2 / n_k$ is the mean squared error of fold k

- Setting $k = N$ is referred to as leave-one out cross-validation (LOOCV)

Classical frequentist model selection

Akaike information criterion (AIC)

$$AIC = -\frac{2}{N} \cdot \text{loglik} + 2 \cdot \frac{d}{N}$$

where d is the number of parameters in our model

Bayesian information criterion (BIC)

$$BIC = -2 \cdot \text{loglik} + (\log N) \cdot d$$

Classical frequentist model selection

Akaike information criterion (AIC)

$$AIC = -\frac{2}{N} \cdot \text{loglik} + 2 \cdot \frac{d}{N}$$

where d is the number of parameters in our model

Bayesian information criterion (BIC)

$$BIC = -2 \cdot \text{loglik} + (\log N) \cdot d$$

- Both penalize models with more parameters in a somewhat arbitrary fashion
- This usually helps with model selection, but still does not answer the important question of model assessment

What is the Optimal Number of Folds, K?

- Do you want big or a small training folds?

What is the Optimal Number of Folds, K?

- Do you want big or a small training folds?
- Because training set is only $(K-1)/K$ as big as the full dataset, the estimates of the prediction error will be biased upward.
- Bias is minimized when $K = N$ (LOOCV)
- But LOOCV has higher variance!

What is the Optimal Number of Folds, K?

- Do you want big or a small training folds?
- Because training set is only $(K-1)/K$ as big as the full dataset, the estimates of the prediction error will be biased upward.
- Bias is minimized when $K = N$ (LOOCV)
- But LOOCV has higher variance!
- No clear statistical rules for how to set k
- Convention is to set $K = 5$ or 10 – in practice is a good trade-off between bias and variance for most problems

Plan

Conclusion

- Econometric models are at risk for overfitting
- But what we want are theories that extend beyond the dataset we have on our computer
- Cross-validation is a key tool that allows us to adjust models so that they more closely match reality