3. Model Selection and Assessment

Jesper Armouti-Hansen

University of Cologne

January 7, 2019

Course website

jeshan49.github.io/eemp2/

Today

- Lecture[1]:
  - Generalization error
  - Cross-validation
  - Bootstrap
- Tutorial:
  - Simulating data and test precision of cross validation methods

[1] Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani

Introduction

- Assessment of the general performance of our learning method is what we truly care about
- The *generalization performance* refers to a method's prediction/classification capability on independent test data
  - Or, more generally, its capability on the population
- If we know methods generalization performance, we
  1. can select the best of our considered learning methods
  2. know the best method's performance on unseen test data
- Today, we'll consider key methods that will give us estimates of the generalization performance

Bias, Variance and Model Complexity

- Let us first suppose Y is quantitative - i.e. prediction

- We have a prediction method $\hat{f}$ that we have estimated on the basis of inputs and outputs (X, y) in a training set Tr (with $|Tr| = N$)

- We denote the loss function for measuring errors between Y and $\hat{f}(X)$ by $L(Y, \hat{f}(X))$

- In a prediction setting, we often use *mean squared errors*

$$L(Y, \hat{f}(X)) = (Y - \hat{f}(X))^2 \tag{1}$$

- Based on our choice of $L(\cdot, \cdot)$, we would like to know our method's general performance

- The *test error* or *generalization error* is the error over an independent sample

  $$Err_{Tr} = E[L(Y, \hat{f}(X))|Tr] \tag{2}$$

  i.e. the expected error of our method, given that (X, Y) are drawn from their joint distribution (population), conditional on being trained on Tr

- Estimation of Err_{Tr} is our goal, however it turns out that our methods better estimate the *expected test error*

  $$Err = E[L(Y, \hat{f}(X))] = E[E[L(Y, \hat{f}(X))|Tr]] = E[Err_{Tr}] \tag{3}$$

  i.e. the expected error over everything that is random – including the training set

- The *training error* is simply the average loss in Tr

$$\bar{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i)) \tag{4}$$

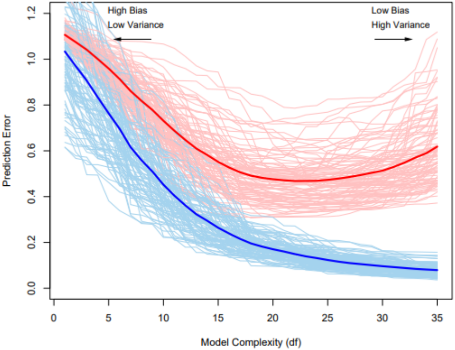


Figure: Light blue curves show $\bar{err}$ while light red curves show Err_{Tr} for 100 training sets of size 50, as model complexity is increased. The solid curves show $E[\bar{err}]$ and Err, respectively (See ESL p. 220)

- The story is the same when Y is qualitative – i.e. classification
- Typically our method will have a hyper/tuning parameter or parameters
  - e.g. # of neighbors for the kNN method
- We wish to find the parameter(s) that minimizes the generalization error
- However, we do in fact have two seperate goals in mind

Model selection: estimating the performance of different models in order to choose the best one

Model assessment: having chosen a final model, estimating its prediction error (generalization error) on new data

Data Rich Situation

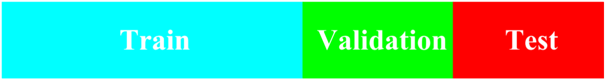
Idea: Estimate the test error by holding out a subset of the training observations from the fitting process, and then applying the statistical learning method to those held out observations.

Validation-set Approach (without model selection):

- Randomly divide the data into two parts: a training set and a validation or hold-out set
- The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set
- The resulting validation-set error provides an estimate of the test error

Validation-set Approach (with model selection)

- Best approach: randomly divide *Tr* into three parts:



Train Validation Test

- We
  - fit our methods on the training data ($\sim 50\%$)
  - estimate error for model selection on the validation data ($\sim 25\%$)
  - estimate the generalization error for our best model on the test data ($\sim 25\%$)
- If we use the test data for both model selection and assessment, we will usually underestimate the true test error of the best model
- In the remainder of the slides, we assume that we are not in a data-rich environment

The Bias-Variance Decomposition

- Let us return to the case where Y is quantitative – i.e. prediction
- Recall that, if we use squared loss, the expected test error at $X = x_0$ of a method $\hat{f}$ is given by:

$$
\begin{aligned}
Err_{x_0} &= E[(Y - \hat{f}(X))^2 | X = x_0] \\
&= Var[\varepsilon] + [f(x_0) - E[\hat{f}(x_0)]]^2 + E[(\hat{f}(x_0) - E[\hat{f}(x_0)])^2] \\
&= \underbrace{Var[\varepsilon]}_{\text{irreducible error}} + \underbrace{[f(x_0) - E[\hat{f}(x_0)]]^2}_{\text{bias}^2 \text{ of } \hat{f}} + \underbrace{Var[\hat{f}(x_0)]}_{\text{variance of } \hat{f}}
\end{aligned}
$$

- Typically, the more complex $\hat{f}$ is the lower the bias^2 of $\hat{f}$ but the higher the variance of $\hat{f}$

- $\mathbf{X} = (X_1, ..., X_{20})$ with $X_i \sim \mathcal{U}[0,1]$ for all i
- $Y = 1$ if $\sum_{i=1}^{10} X_i > 5$ and $Y = 0$ otherwise
- $|Tr| = 80$

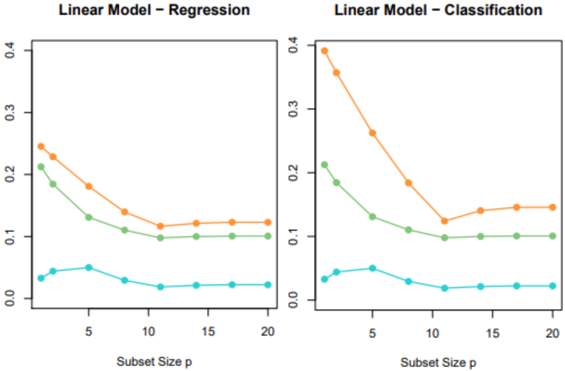


Figure: Expected test error (orange), squared bias (green) and variance (blue). Left: squared error. Right: 0-1 error. (See ESL p. 227)

- $\mathbf{X} = (X_1, ..., X_{20})$ with $X_i \sim \mathcal{U}[0,1]$ for all i
- $Y = 1$ if $X_1 \geq 0.5$ and $Y = 0$ otherwise
- $|Tr| = 80$

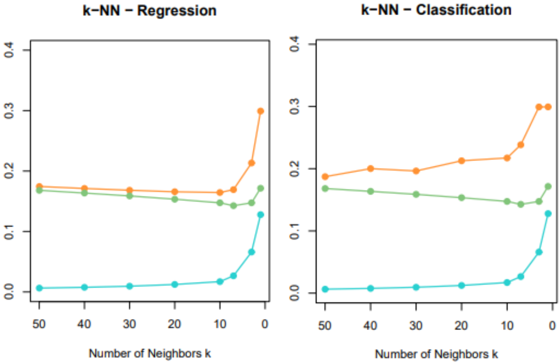


Figure: Expected test error (orange), squared bias (green) and variance (blue). Left: squared error. Right: 0-1 error. (See ESL p. 227)

Optimism of the Training Error Rate

- Typically, the training error

$$\bar{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i)) \tag{5}$$

will be smaller than the test error Err_{Tr}

- This is because the same data is used to both fit and assess its error
- We can think of Err_{Tr} as being the *extra-sample error*, since the test input need not coincide with training input
- The optimism in $\bar{err}$ is easier to understand if we focus on *in-sample error*:

$$Err_{in} = \frac{1}{N}\sum_{i=1}^{N} E[L(Y_i, \hat{f}(x_i)|Tr] \tag{6}$$

- The optimism of the training error can then be considered to be $Err_{in} - \bar{err}$

- Given this, one estimate of the error of a method would be to estimate the optimism and add it to the training error

- Methods such as C_p, AIC and BIC do this for a special class of models that are linear in their parameters

- The estimate will usually not be of direct interest since future inputs are unlikely to coincide with training inputs

- However, in-sample error often leads to effective model selection

- We will see this in the next lecture

K-fold Cross Validation

The simplest and most popular way to estimate *Err*:

1. Randomly split the data into K roughly equal parts
2. For each $k = 1, ..., K$:
   - leave the kth part out and fit the model using the other $K - 1$ parts: $\hat{f}^{-k}(x)$
   - calculate the error of $\hat{f}^{-k}(x)$ on the held out kth part
3. Average the k errors

- Define the index function $\kappa : \{1, ..., N\} \rightarrow \{1, ..., K\}$ (allocating membership)
- Then, the K-fold cross validation estimate of *Err* is:

$$CV(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i)) \tag{7}$$

Leave-Out-One Cross Validation (LOOCV)

- If $K = N$, we have LOOCV
- LOOCV is approximately unbiased for *Err* (low bias) since each training set contains $N - 1$ observations
- However, LOOCV can have high variance
  - we are averaging over N fitted models with almost identical observations
  - thus, our outputs will be highly (positively) correlated with each other
- Additionally, LOOCV is computationally expensive since we are fitting N models
- However, for a class of linear models, this is not an issue

What value should we choose for K?

- We have seen that $K = N$ can suffer from high variance, so we might want to reduce K

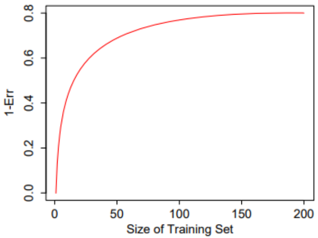


Figure: Hypothetical learning curve for a classifier on a given task: a plot of $1 - Err$ versus the size of the training set N (See ESL p. 243)

- The optimal choice of K thus depends on the slope of the learning curve
- In general, 5- or 10-fold CV are recommended as a good compromise between bias and variance

Choosing hyper parameters with CV

- Often we consider a class of models with hyper parameter(s) α
  - e.g. number of neighbors in kNN
- Denote by $\hat{f}^{-k}(x, \alpha)$ the αth model fit with the kth part of data removed
- Then, we have

$$CV(\hat{f}, \alpha) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i, \alpha)) \tag{8}$$

- Usually, instead of choosing the $\hat{\alpha}$ which minimizes (8), we apply the "one standard error rule":

choose the most parsimonious model with error no more than one standard error above the best error

Example - Linear Classification (Cont'd)

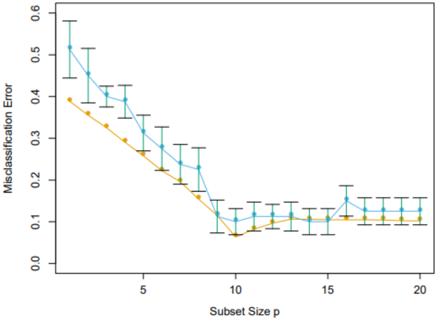


Figure: Test error (orange) and 10-fold CV curve (blue) from a single training set (See ESL p. 244)

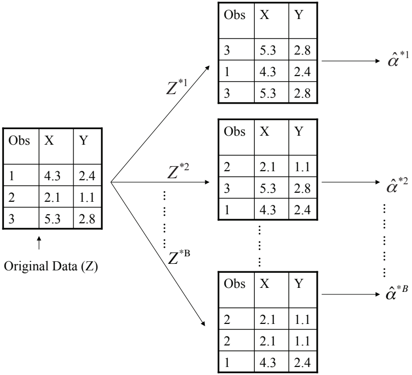
- Which model would be chosen here?

Drawbacks of K-Fold Cross Validation

- Recall that in the data-rich situation:
  - If we use the test data for both model selection and assessment, we will risk underestimating the true test error of the best model
- The same holds with cross validation
- Model selection with CV uses the same data to tune model parameters and evaluate model performance
- Information may thus “leak” into the model and overfit the data
- To limit this problem, two approaches are used in practice:
  - Repeated K-fold CV (e.g. performing 100 5-fold CVs with different splits)
  - Nested K-fold CV: Model selection on inner CV and model assessment on outer cv

Bootstrap Methods

- As with CV, bootstrap seeks to estimate Err_{Tr} but will usually only estimate Err well



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

Figure: Schematic of bootstrap process. We wish to estimate accuracy of α computed from Z (See ISLR p. 190)

- Let $\hat{f}^{*b}(x_i)$ be the predicted value of a fitted model at x_i on bootstrap sample b

- Then our estimate of Err is given by

$$\hat{Err}_{boot} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L(y_i, \hat{f}^{*b}(x_i)) \qquad (9)$$

- However, this is not a good estimate of Err. Why?

- The original training set acts as the test set, while the bootstrap samples act as training sets
  - however, they will necessarily have common observations
  - this overlap can make overfit predictions look unrealistically good
  - this is the reason why CV uses non-overlapping data

Example

- Consider a balanced two-classed classification problem
- Suppose class labels and inputs are independent
  - This means that the true misclassification rate is 0.5
- We wish to estimate the misclassification rate of 1-NN classifier using bootstrapping
- Consider the *i*th observation:
  - if i is in bootstrap sample b, its contribution to $\hat{Err}_{boot}$ will be zero
  - if i is not in bootstrap sample b, its expected contribution to $\hat{Err}_{boot}$ will be 0.5

- Note that

$$Pr(i \in b) = 1 - \left(1 - \frac{1}{N}\right)^N \approx 1 - exp(-1) = 0.632 \quad (10)$$

- Thus, the expected $\hat{Err}_{boot} \approx 0.5 \times 0.368 = 0.184$, far below 0.5
- We can mimic CV to correct for this: *Leave-one-out bootstrap*:

$$\hat{Err}^{(1)} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|C^{-i}|} \sum_{b \in C^{-i}} L(y_i, \hat{f}^{*b}(x_i)) \qquad (11)$$

Where C^{-i} is the set of bootstrap samples which do not contain observation i

- LOO-bootstrap solves the overfitting problem, but suffers from a similar bias compared to CV:
  - # of distinct observations in each b is approx. $0.632 \times N$, on average
- If the learning curve is step at this point, LOO-bootstrap will be considerably upward biased
- The ".632 bootstrap estimator" attemps to alleviate this bias:

$$\hat{Err}^{(.632)} = .368 * \bar{err} + .632 * \hat{Err}^{(1)} \tag{12}$$

- .632 bootstrap works well in some situations, but in others, it will tend to favor overfit models as the standard bootstrap
- There exists a method called ".632+ bootstrap estimator" which attempts to correct for this

Final remarks

- In general, for model selection CV, bootstrap (and the model-based approaches such as AIC) work approximately equal
- We care less about bias in estimates here given that it is equally distributed
- For CV, choosing the # of folds involves a bias-variance trade-off:
  - $K = N$ is approx. unbiased, but may suffer from substantial variance
  - $K = 5, 10$ are biased, but do not suffer from as much variance
- The LOO-bootstrap is biased for the same reason as CV, although methods exist for correcting this
- If we perform CV for both model selection and assessment, we may get a too optimistic estimate of the test error for the best model

References

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). **Chapter 5**

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). **Chapters 7**