Stat 435 Lecture Notes 4a

Xiongzhi Chen
Washington State University

Contents

Model assessment

Model and estimate

- Model: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon$
- Observations: $\mathbf{z}_i = (y_i, x_{i1}, x_{i2}, \ldots, x_{ip}), i = 1, \ldots, n$, where y_i is the ith observation for Y and x_{ij} that for X_j
- Estimate $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ of $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$
- Fitted model: $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \ldots + \hat{\beta}_p x_{ip}$
- Residuals: $e_i = y_i - \hat{y}_i$

Note: there are many ways to obtain an estimate $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$

Training error

- *Training error* is a measure on the performance of a model when it is fitted/trained from a set of observations that we call the *training set*. Namely, *training error measures how well a fitted/trained model fits/learns the training set*.
- A commonly used training error is the *residual sum of squares (RSS)* (though other choices are available)
- For example, if we take the set of n observations as the training set and use RSS as the training error, then the *least squares estimate (LSE)* minimizes $\text{RSS} = \sum_{i=1}^{m} e_i^2$

Test error

- *Test error* is a measure on the *predictive performance* of a (fitted/trained) model when it is used to predict the responses of new observations on the predictors $\tilde{X} = (X_1, \ldots, X_p)$ that are not in the training set (that is used to fit/train the model). Namely, *test error measures how well a fitted/trained model predicts responses for observations that are not part of a training set.*
- For example, given a new observation $\mathbf{x}_0 = (x_{01}, x_{02}, \ldots, x_{0p})$ for $\tilde{X}$, let y_0 be the predicted response given by the fitted/trained model. Then we can use $E(y_0 - \hat{y}_0)^2$ as the test error.

Note: other measures of test error are available

Training error and test error

- Training error measures how well a fitted/trained model fits/learns the *seen* observations (of the training set), whereas test error measures how well this model *predicts unseen responses* based on unseen observations on predictors
- *Test error is alway unknown and needs to be estimated*
- A model that has good predictive performance is referred to as having good "generalization performance" or "generalizes well"
- A model that fits the training set very well or perfectly is often referred to as "overfitting" or "overfits"

Training error and test error

- Classic wisdom: it is believed that an overfitting model that fits the training set very well or perfectly (i.e., with very small training error) cannot generalize well (i.e., cannot have small testing error)
- Modern wisdom: when there are relatively few parameters in the model, the classic wisdom is sensible. However, beyond a critical setting, an overfitting model (with many parameters) can have very good generalization performance.

Note: Modern wisdom, dubbed as the "**double descent curve**", was discovered by Dr. Belkin and his coauthors

Double descent curve

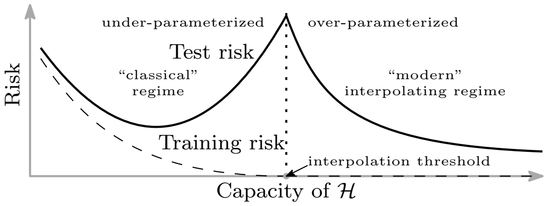

Image credit: Belkin et al; doi.org/10.1073/pnas.1903070116

Cross-validation

Overview

Cross-validation is a resampling technique to estimate test error, and is often implemented as follows:

- Randomly divides a set of observations into a *training set* and *validation set*
- Use the training set to fit a model (that optimizes some *training error*), and apply the fitted model to predict the responses for the observations in the validation set to obtain an estimate of the *test error* of the model
- Do the above independently for different random splits, and estimate test error from the estimates of test error

Pictorial description

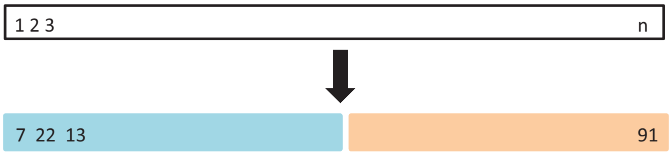

FIGURE 5.1. *A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.*

CV: estimate test error

With n observations $\mathbf{z}_i = (y_i, \mathbf{x}_i)$,

- Randomly split the n observations into a training set $\mathcal{T}_1$ with n_1 observations, and a validation set $\mathcal{V}_1$ with $n_2 = n - n_1$ observations
- Fit model M_l using $\mathcal{T}$, apply fitted model $\hat{M}_l$ to predict responses in $\mathcal{V}_1$, and compute the *mean squared error (MSE)*

  $$\text{MSE}(\mathcal{V}_1) = n_2^{-1} \sum_{y_i \in \mathcal{V}_1} (y_i - \hat{y}_i)^2,$$

  where $\hat{y}_i$ is the fitted value for y_i

Note: $\mathcal{T}_1$ and $\mathcal{V}_1$ are disjoint

CV: estimate test error

- Repeat the above “splitting-training-validating” steps independently k times for model M_l to obtain k MSE’s $\text{MSE}(\mathcal{V}_j), j = 1, \ldots, k$, and estimate the test *mean squared error (MSE)* of model M_l by

$$\text{CV}(M_l) = k^{-1} \sum_{j=1}^{k} \text{MSE}(\mathcal{V}_j)$$

Note: the above steps give an estimate of test error of model M_l

CV: model selection

- Obtain via cross-validation the estimated test error for each model $M_l, l = 1, \ldots, L$
- Pick the model that has the smallest estimated test MSE, i.e., pick the optimal model M^* such that

$$\text{CV}(M^*) = \min_{1 \le l \le L} \text{CV}(M_l)$$

Note: the above gives the best model among a set of models

Illustration

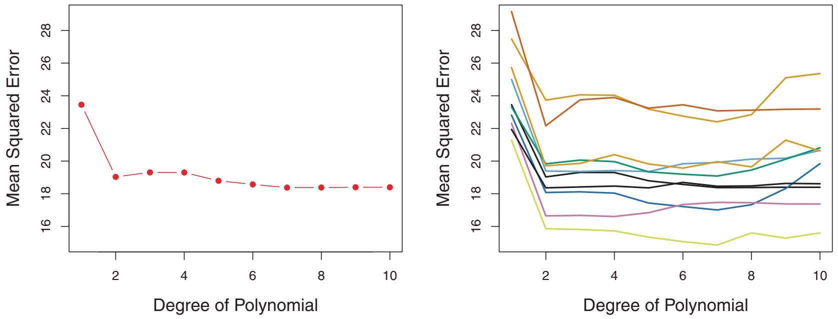

FIGURE 5.2. *The validation set approach was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`*.* Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.*

k-fold cross-validation

The k-fold CV for model M_l is implemented as follows:

- Randomly splits the data set into k groups, or "folds", of approximately equal sizes
- Pick a fold, say, folder j, as the validation set (i.e., the "held-out" fold), fit the model on the remaining $k-1$ folds, and compute the mean squared error, MSE_j, on the observations in the held-out fold
- Do the above for all $j, j = 1, \ldots, k$, and estimate the test MSE of the model by

$$\text{CV}(M_l) = \frac{1}{k}\sum_{j=1}^{k} \text{MSE}_j$$

Variable and model selection

Motivation

Settings:

- Model: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon$
- Observations: $(y_i, x_{i1}, x_{i2}, \ldots, x_{ip}), i = 1, \ldots, n$, where y_i is the ith observation for Y and x_{ij} that for X_j
- Estimate $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ of $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$

How to obtain an estimate $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ depends critically on

- the relative magnitudes of $p + 1$ and n
- what properties we want $\hat{\boldsymbol{\beta}}$ to have

Classic scenario: $p + 1 \leq n$

When the number of parameters is not larger than the sample size:

- The *least squares estimate (LSE)* is *uniquely* defined
- The LSE is unbiased, i.e., $E(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$, when $E(\varepsilon) = 0$
- The LSE is optimal in some sense (e.g., Gauss-Markov theorem, meaning that the LSE has the smallest variance among all linear unbiased estimators, if the random errors are uncorrelated and have equal variances and zero expectation)

Classic scenario: $p + 1 \leq n$

When the number of parameters is not larger than the sample size but *there are many potential predictors*, we often desire a *small* model that is easy to interpret and perform well. Namely, we still need to consider:

- Which predictors are the *most relevant* to the response
- how to build a *small* model using a few predictors that can well predict the response

Variable or model selection is needed in the classic scenario when there are many potential predictors.

Modern scenario: $p + 1 > n$

When the number of parameters is bigger than the sample size,

- The LSE is *not uniquely* defined, i.e., there are infinitely many models that minimizes the residual sum of squares
- An LSE is biased in general, i.e., $E(\hat{\boldsymbol{\beta}}) \neq \boldsymbol{\beta}$, even when $E(\varepsilon) = 0$
- An LSE has infinite variance (and Gauss-Markov theorem does not hold for LSE)

Modern scenario: $p + 1 > n$

In this scenario, we have a few choices:

- perform variable/model selection: select a few most relevant predictors to form a model and then apply LSE to estimate the model parameters
- employ different estimation methods: use a different method (such as *shrinkage*) than LSE to estimate coefficients of predictors in a model (as a *bias-variance trade-off*)
- use dimension reduction: projecting all predictors onto a smaller subspace and use transformed predictors to build a model

Best subset selection

Overview

- Consider a linear model with p predictors:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon$$

- *Best subset selection* is a "brute-force" method that checks *each of the* 2^p *possible linear submodels* and picks the best one under some criterion

- Best subset selection indeed gives the *best subset of predictors (in terms of linear model)* among all p predictors under a criterion

Criteria for subset selection

Some criteria for subset/variable selection:

- Mallow's C_p
- AIC (i.e., Akaike information criterion),
- BIC (i.e., Bayesian information criterion)
- Adjusted R^2
- Cross-validated prediction error

Implementation

Algorithm 6.1 *Best subset selection*

```
1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
    (a) Fit all $\binom{p}{k}$ models that contain exactly $k$ predictors.
    (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.
```

Illustration

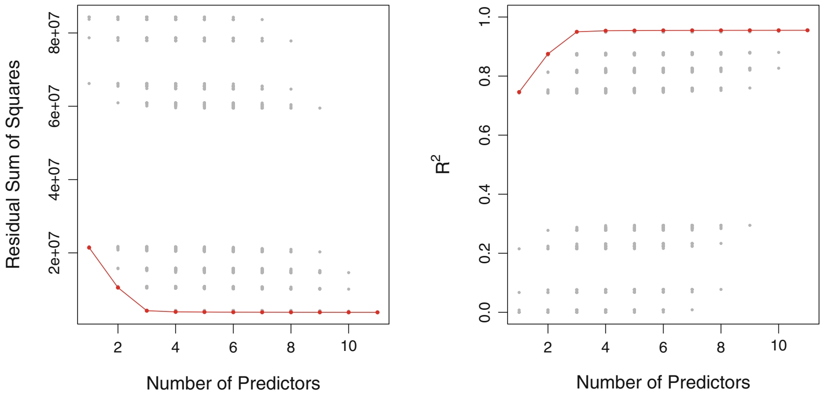

FIGURE 6.1. *For each possible model containing a subset of the ten predictors in the* `Credit` *data set, the RSS and R^2 are displayed. The red frontier tracks the* best *model for a given number of predictors, according to RSS and R^2. Though the data set contains only ten predictors, the x-axis ranges from 1 to 11, since one of the variables is categorical and takes on three values, leading to the creation of two dummy variables.*

Forward/backwards stepwise selection

Overview

Forward stepwise selection

- only fits
$$1+\sum_{k=0}^{p-1}(p-k) = 1+2^{-1}p\,(p+1)$$
linear submodels out of a total of 2^p linear submodels
- is not guaranteed to find the best model among all linear submodels
- has much smaller computational intensity than best subset selection

Implementation

Algorithm 6.2 *Forward stepwise selection*

```
1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \dots, p-1$:
    (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
    (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.
```

Illustration

# Variables	Best subset	Forward stepwise
One	`rating`	`rating`
Two	`rating`, `income`	`rating`, `income`
Three	`rating`, `income`, `student`	`rating`, `income`, `student`
Four	`cards`, `income`, `student`, `limit`	`rating`, `income`, `student`, `limit`

TABLE 6.1. *The first four selected models for best subset selection and forward stepwise selection on the* `Credit` *data set. The first three models are identical but the fourth models differ.*

Backwards stepwise selection

Algorithm 6.3 *Backward stepwise selection*

```
1. Let $\mathcal{M}_p$ denote the full model, which contains all $p$ predictors.
2. For $k = p, p-1, \ldots, 1$:
    (a) Consider all $k$ models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
    (b) Choose the best among these $k$ models, and call it $\mathcal{M}_{k-1}$. Here best is defined as having smallest RSS or highest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.
```

Choosing the optimal model

General principle

For best subset, forward stepwise, and backwards stepwise selections, *we need to select a best model from the best submodels*. However, neither *training* set RSS nor *training* set R^2 can be used for this purpose since

- neither ensures good predictive performance in terms of test error of the resultant model
- either tends to pick a model that has the largest possible size in terms of the number of parameters

So, a practical way to select a single best model is to **balance the RSS on the training error and the model size**

Four methods

Four methods to choose the optimal model:

- Mallow's C_p
- Akaike information criterion
- Bayesian information criterion
- Cross-validation

Mallow's C_p

Mallow's

$$C_p = \frac{1}{n}\left(\text{RSS} + 2d\hat{\sigma}^2\right) \tag{1}$$

- d is the number of predictors in the model
- Typically $\hat{\sigma}^2$, an estimate of $\sigma^2 = \text{Var}(\varepsilon)$, is estimated using the full model containing all predictors

Remark: If $\hat{\sigma}^2$ is an unbiased estimate of σ^2, then C_p is an unbiased estimate of test MSE

Akaike information criterion

- The *Akaike information criterion (AIC)* is mainly used together with the *maximum likelihood method*
- For a linear model with Gaussian errors, the *maximum likelihood estimate (MLE)* is the same as the *least squares estimate (LSE)*. In this case, AIC is, up to an additive constant, given by

$$\mathrm{AIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

Bayesian information criterion

- The *Bayesian information criterion (BIC)* is derived from a Bayesian perspective
- For the least squares model with d parameters, the BIC is, up to irrelevant constants, given by

$$\mathrm{BIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + d\hat{\sigma}^2 \log n\right)$$

- BIC generally places a heavier penalty on models with many variables. For large n, BIC is bigger than C_p

Adjusted R^2

- The adjusted R^2 is given by

$$\text{Adjusted } R^2 = 1 - \frac{\mathrm{RSS}/\left(n-d-1\right)}{\mathrm{TSS}/\left(n-1\right)}$$

- *Unlike C_p, AIC and BIC, for which a small value indicates a model with a low test error, a large value of adjusted R^2 indicates a model with a low test error*

The intuition behind the adjusted R^2 is that "once all of the correct variables have been included in the model, adding additional noise variables will lead to a very small decrease in RSS"

Recap

Let d be number of predictors in model, n sample size, and $\hat{\sigma}^2$ an estimate of $\sigma^2 = \mathrm{Var}(\varepsilon)$:

- $C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$
- $\mathrm{AIC} = \frac{1}{n\hat{\sigma}^2}\left(RSS + 2d\hat{\sigma}^2\right)$
- $\mathrm{BIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + d\hat{\sigma}^2 \log n\right)$
- Adjusted $R^2 = 1 - \frac{\mathrm{RSS}/(n-d-1)}{\mathrm{TSS}/(n-1)}$

All formulae for C_p, AIC and BIC are *for a linear model fit using least squares*; C_p, AIC and BIC all have good theoretical justifications

Illustration

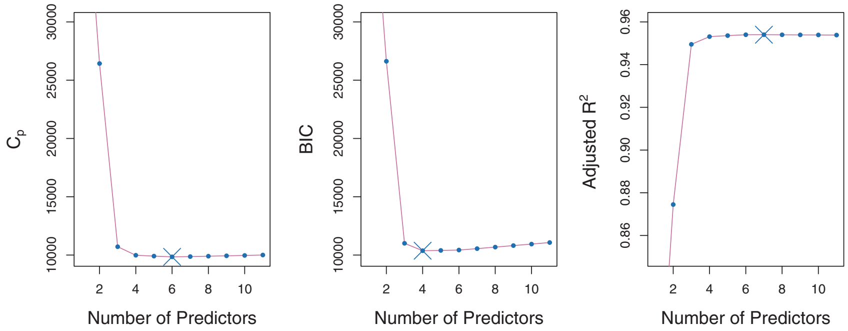

FIGURE 6.2. *C_p, BIC, and adjusted R^2 are shown for the best models of each size for the* `Credit` *data set (the lower frontier in Figure 6.1). C_p and BIC are estimates of test MSE. In the middle plot we see that the BIC estimate of test error shows an increase after four variables are selected. The other two plots are rather flat after four variables are included.*

k-fold cross-validation

Recall CV for model selection:

- Obtain via cross-validation the estimated test error for each model $M_l, l = 1, \dots, L$
- Pick the model that has the smallest estimated test MSE, i.e., pick the optimal model M^* such that

$$\mathrm{CV}(M^*) = \min_{1 \le l \le L} \mathrm{CV}(M_l)$$

k-fold cross-validation

The k-fold CV for model M_l is implemented as follows:

- Randomly splits the data set into k groups, or "folds", of approximately equal sizes
- Pick a fold, say, folder j, as the validation set (i.e., the "held-out" fold), fit the model on the remaining $k - 1$ folds, and compute the mean squared error, MSE_j, on the observations in the held-out fold
- Do the above for all $j, j = 1, \dots, k$, and estimate the test MSE of the model by

$$\mathrm{CV}(M_l) = \frac{1}{k} \sum_{j=1}^{k} \mathrm{MSE}_j$$

k-fold cross-validation

Guideline:

- Usually $k = 5$ or 10 is chosen, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance

k-fold cross-validation

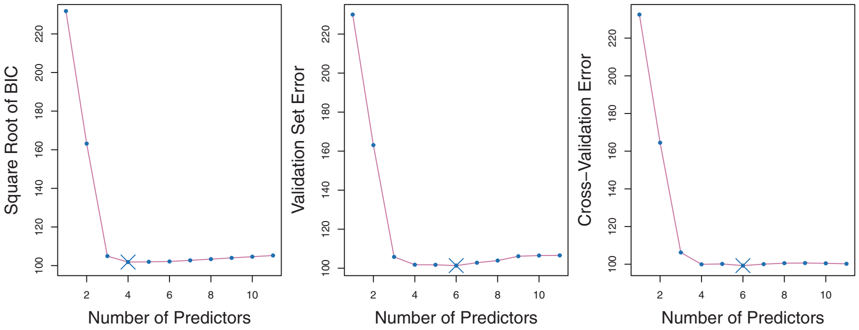

FIGURE 6.3. *For the* `Credit` *data set, three quantities are displayed for the best model containing d predictors, for d ranging from 1 to 11. The overall* best *model, based on each of these quantities, is shown as a blue cross.* Left: *Square root of BIC.* Center: *Validation set errors.* Right: *Cross-validation errors.*

One-standard-error rule

Different random splitting schemes often lead to differential optimal models. So,

- first, we can calculate the standard error of the estimated test MSE for each model size, by repeatedly validating "the best model" of this model size
- then select the smallest model for which the estimated test error is within one standard error of the lowest point on the front curve for the estimated MSEs of "the best models"

License and session Information

License

```
> sessionInfo()
R version 3.5.0 (2018-04-23)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19043)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods
[7] base

other attached packages:
[1] knitr_1.21
```

```
loaded via a namespace (and not attached):
 [1] compiler_3.5.0  magrittr_1.5    tools_3.5.0
 [4] htmltools_0.3.6 yaml_2.2.0      Rcpp_1.0.3
 [7] stringi_1.2.4   rmarkdown_1.11  stringr_1.3.1
[10] xfun_0.4        digest_0.6.18   evaluate_0.12
```