UNIVERSITY *of* WEST FLORIDA

Cross-Validation

Statistics for Data Science II

Introduction

We are often interested in how well our model performs with “real data.”

We cannot truly determine the performance of the model with data used to construct the model as this data is considered *biased*.

In this lecture, we will learn about the validation set approach, leave-one-out cross-validation, and k-fold cross-validation.

Example:

We will return to the penguin dataset as a basic example. Let us consider modeling penguin body mass (g) as a function of flipper length (mm).

M1: body mass $\sim$ flipper

M2: body mass $\sim$ flipper $+$ flipper^2

Introduction

Example:

```
m1 <- glm(body_mass_g ~ flipper_length_mm, data = data)
summary(m1)[12]
```

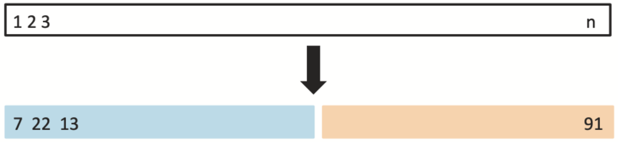
```
## $coefficients
##                      Estimate Std. Error   t value      Pr(>|t|)
## (Intercept)       -5872.09268 310.285155 -18.92483  1.183941e-54
## flipper_length_mm    50.15327   1.540231  32.56217 3.132836e-105
```

```
data <- data %>% mutate(flipper2 = flipper_length_mm^2)
m2 <- glm(body_mass_g ~ flipper_length_mm + flipper2, data = data)
summary(m2)[12]
```

```
## $coefficients
##                        Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept)       16579.8692148 4616.2466470  3.591634 3.785927e-04
## flipper_length_mm  -171.6139765   45.5244701 -3.769708 1.935957e-04
## flipper2              0.5449478    0.1118072  4.873996 1.701917e-06
```

Validation Set Approach

We will first split our data into training and validation datasets.



The training data will be what we use to construct the model.

We will then apply the model to the validation data to determine how good the prediction is.

Why do we need a separate (validation) dataset?

We need to use data that was *not* used to create the model. (i.e., unbiased data)

UNIVERSITY of WEST FLORIDA

Validation Set Approach

We will use the mean square error (MSE) to assess how well the model performs.

- squared error $= (y_i - \hat{y}_i)^2$
- MSE $=$ average of squared error for all observations in the validation set

We should construct multiple training and validation sets.

- When we construct multiple models, we will choose the model with the lowest MSE.

We will use the `tidymodels` package to split the data.

UNIVERSITY of WEST FLORIDA

Validation Set Approach

Example:

```
set.seed(906282) # reproducible sampling
split <- initial_split(data, prop = 0.5)
training <- training(split)
validation <- testing(split)
head(training)
```

```
## # A tibble: 6 x 10
##     obs species island bill_length_mm bill_depth_mm flipper_length_~ body_mass_g
##   <int> <fct>   <fct>           <dbl>         <dbl>            <int>       <int>
## 1   280 Chinst~ Dream            45.9          17.1              190        3575
## 2   199 Gentoo  Biscoe           50.1          15                225        5000
## 3   253 Gentoo  Biscoe           48.1          15.1              209        5500
## 4    73 Adelie  Torge~           36.2          16.1              187        3550
## 5   146 Adelie  Dream            41.5          18.5              201        4000
## 6   160 Gentoo  Biscoe           48.4          14.6              213        5850
## # ... with 3 more variables: sex <fct>, year <int>, flipper2 <dbl>
```

Validation Set Approach

Example:

Now, we will construct our models using the training dataset.

```r
m1v <- lm(body_mass_g ~ flipper_length_mm, data = training)
m2v <- lm(body_mass_g ~ flipper_length_mm + flipper2, data = training)
```

We now need to compute the MSE for each model using the validation dataset.

First, we will set up the squared errors, $(y_i - \hat{y}_i)^2$, under each model.

Then, we take the average of the squared error to find the MSE for each model.

Validation Set Approach

Example:

```
validation <- validation %>%
  mutate(yhat_m1 = predict(m1v, newdata = validation),
         yhat_m2 = predict(m2v, newdata = validation)) %>%
  mutate(sqerr_m1 = (yhat_m1 - body_mass_g)^2,
         sqerr_m2 = (yhat_m2 - body_mass_g)^2)
```

```
## # A tibble: 6 x 14
##     obs body_mass_g yhat_m1 sqerr_m1 yhat_m2 sqerr_m2 species island
##   <int>       <int>   <dbl>    <dbl>   <dbl>    <dbl> <fct>   <fct>
## 1     1        3750   3082.  446651.   3228.  272091. Adelie  Torgersen
## 2     2        3800   3351.  201979.   3409.  152703. Adelie  Torgersen
## 3     4        3450   3727.   76751.   3699.   61935. Adelie  Torgersen
## 4     5        3650   3566.    7107.   3570.    6476. Adelie  Torgersen
## 5     6        3625   3082.  295196.   3228.  157310. Adelie  Torgersen
## 6     8        3200   3135.    4165.   3263.    3945. Adelie  Torgersen
## # ... with 6 more variables: bill_length_mm <dbl>, bill_depth_mm <dbl>,
## #   flipper_length_mm <int>, sex <fct>, year <int>, flipper2 <dbl>
```

Validation Set Approach

Example:

```
mean(validation$sqerr_m1)
```

```
## [1] 171215.7
```

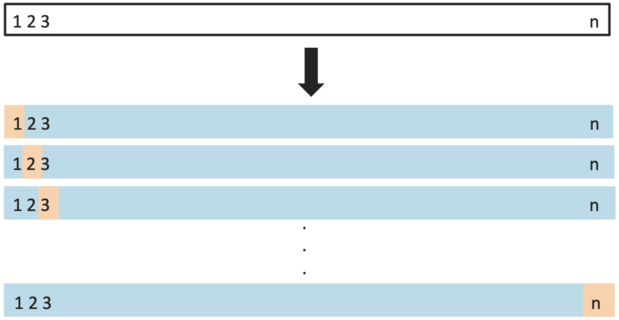
```
mean(validation$sqerr_m2)
```

```
## [1] 153882.4
```

Of the two candidate models, the model including the quadratic term (M2) gives the lowest MSE.

Leave-One-Out Cross-Validation

The leave-one-out cross-validation is similar to what we discussed under the validation set approach, however, we now leave out a single observation.



Leave-One-Out Cross-Validation

As we are only excluding a single observation for validation purposes, we will have n MSEs to consider.

$$\text{MSE}_i = (y_i - \hat{y}_i)^2$$

Because the MSE is now based on a single observation, the variability is high.

Thus, we then consider the leave-one-out cross-validation estimate for the test MSE,

$$\text{CV}_{(n)} = \frac{\sum_{i=1}^n \text{MSE}_i}{n}$$

These are useful when considering various models in terms of what is being included as predictors (e.g., higher order polynomial terms).

Leave-One-Out Cross-Validation

Note that as n increases *and* as the model complexity increases, it will be computationally intensive/expensive to implement this method.

If using least squares regression, we can use the following estimate:

$$\text{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2,$$

where h_i is the leverage as defined in the lecture on model assumptions and diagnostics.

Leave-One-Out Cross-Validation

Example:

```r
cv_error <- cv.glm(data, m1)
cv_error$delta
```

```
## [1] 155563.6 155560.9
```

```r
cv_error <- cv.glm(data, m2)
cv_error$delta
```

```
## [1] 145759.7 145756.3
```

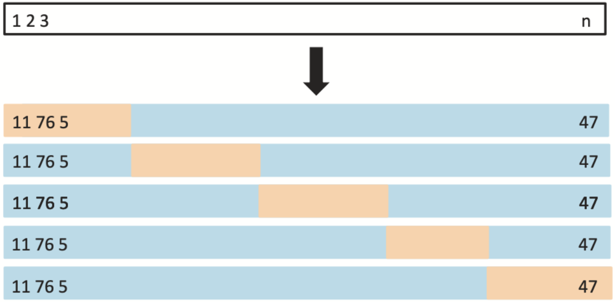
The first value is the estimated $\text{CV}_{(n)}$ while the second value is the true $\text{CV}_{(n)}$.

The model with the quadratic term has a lower test $\text{CV}_{(n)}$, thus, fits better.

k-Fold Cross-Validation

An alternative to leave-one-out cross-validation is the k-fold cross-validation.

Instead of leaving a single observation out, we now leave a group of observations out.



k-Fold Cross-Validation

We are dividing the dataset into k groups (or folds) of approximately equal size.

We treat the first group as the validation set.

The other $k-1$ groups are used to construct the model.

Then, we compute the MSE on the first group.

We repeat this process k times, giving us k estimates of the test error.

We construct $\text{CV}_{(k)}$ as the average MSE,

$$\text{CV}_{(k)} = \frac{\sum_{i=1}^{k} \text{MSE}_i}{k}$$

Note that leave-one-out cross-validation is a special case of k-fold cross-validation, where $k = n$.

k-Fold Cross-Validation

Example:

```r
m1 <- glm(body_mass_g ~ flipper_length_mm, data=data)
cv_error <- cv.glm(data, m1, K=10)
cv_error$delta
```

```
## [1] 154948.9 154888.0
```

```r
m2 <- glm(body_mass_g ~ flipper_length_mm + flipper2, data=data)
cv_error <- cv.glm(data, m2, K=10)
cv_error$delta
```

```
## [1] 145677.2 145560.8
```

The first value is the estimated $\text{CV}_{(k)}$ while the second value is the true $\text{CV}_{(k)}$.

The values above indicate that the model with the quadratic term offers a better fit.

Cross-Validation in Classification Problems

What if we are working with categorical data and using logistic regression?

Instead of calculating a $\text{CV}_{(n)}$ based on the MSE, we will now base it on the number of misclassified observations.

$$\text{CV}_{(n)} = \frac{\sum_{i=1}^{n} \text{Err}_i}{n},$$

where $\text{Err}_i = I(y_i \neq \hat{y}_i)$. (i.e., is the count of the number of misclassifications.)

We will again use the glm() function for modeling and then employ the cv.glm() function for cross-validation.

Cross-Validation in Classification Problems

Example:

Recall the graduate school admissions data from the binary logistic regression lecture: A researcher is interested in how student characteristics affect admission into graduate school. We modeled graduate school admission as a function of GRE, college GPA, and prestige of the undergraduate institution.

```
m1 <- glm(admit ~ gre + gpa + rank, data = data, family = "binomial")
summary(m1)[12]
```

```
## $coefficients
##                Estimate  Std. Error   z value     Pr(>|z|)
## (Intercept) -3.44954840 1.132846009 -3.045029 2.326583e-03
## gre          0.00229396 0.001091839  2.101005 3.564052e-02
## gpa          0.77701357 0.327483878  2.372677 1.765968e-02
## rank        -0.56003139 0.127136989 -4.404945 1.058109e-05
```

Cross-Validation in Classification Problems

Example:

Let us compare models with/without the prestige of the undergraduate institution predictor using leave-one-out cross-validation:

```r
m1 <- glm(admit ~ gre + gpa + rank, data = data, family = "binomial")
cv_error <- cv.glm(data, m1)
cv_error$delta
```

```
## [1] 0.1992860 0.1992809
```

```r
m2 <- glm(admit ~ gre + gpa, data = data, family = "binomial")
cv_error <- cv.glm(data, m2)
cv_error$delta
```

```
## [1] 0.2099091 0.2099051
```

The model with the prestige predictor fits better than the model without.

Cross-Validation in Classification Problems

Example:

Let us repeat the last example under *k*-fold cross-validation:

```
m1 <- glm(admit ~ gre + gpa + rank, data = data, family = "binomial")
cv_error <- cv.glm(data, m1, K=10)
cv_error$delta
```

```
## [1] 0.1981200 0.1979561
```

```
m2 <- glm(admit ~ gre + gpa, data = data, family = "binomial")
cv_error <- cv.glm(data, m2, K=10)
cv_error$delta
```

```
## [1] 0.2097260 0.2095673
```

Again, the model with the prestige predictor fits better than the model without.

Conclusions

Cross-validation is another tool in our toolbox for quantifying the error in our models.

> When we use it to compare several candidate models, we can quantify that reduction in error.

While our examples showed that specific models fit better, it could be the case that the reduction is very small.

> It may not be “worth" a more complicated model for a small reduction in error.