Cross-validation Tutorial

Overview

Researchers in the social sciences often want to assess the predictive power of their model, but are frequently limited by the size of their data (i.e., they do not have enough data to split the data set and run/compare the model multiple times). One method of overcoming this issue is *cross-validation*.

This tutorial will provide code to conduct cross-validation (https://en.wikipedia.org/wiki/Cross-validation_(statistics)) (CV) for a simple regression model using the "sat.act" data set from the `psych` package. Specifically, we are going to predict participants' ACT scores from their gender, age, SAT verbal score, and SAT math score, and use cross-validation to estimate the prediction error of this model.

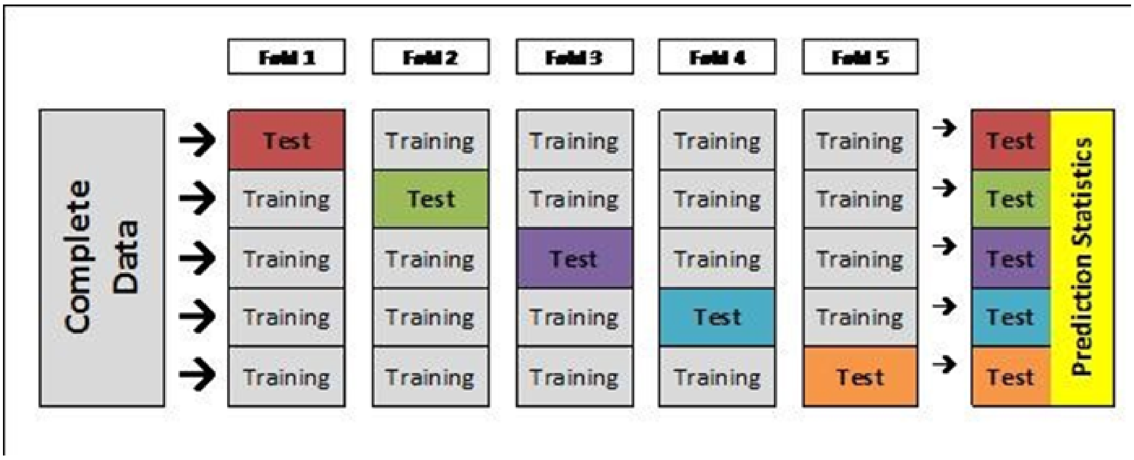
Outline

1. Read in data and needed packages.
2. Cross-validation using `caret` package.
3. Cautions.
4. Conclusions.

Introduction to Cross-validation

Cross-validation is a useful tool when the size of the data set is limited. In a perfect world, our data sets would be large enough that we could set aside a sizable portion of the data set to validate (i.e., examine the resulting prediction error) the model we run on the majority of the data set. Unfortunately, this type of data is not always available, especially in social science research.

To combat the issue of limited data, while still being able to assess the fit of the model, we use *cross-validation*. Essentially, cross-validation iteratively splits the data set into two portions: a test and a training set. The prediction errors from each of the test sets are then averaged to determine the expected prediction error for the whole model. The figure below (from https://stackoverflow.com/questions/40368467/cross-validation-extracting-the-model-values-out-per-row (https://stackoverflow.com/questions/40368467/cross-validation-extracting-the-model-values-out-per-row)) helps depict the cross-validation process.



In this case, the data were split (or *folded*) into five equally sized partitions. During the first fitting of the model, the first 20% of the data (i.e., the first fold) are considered the test set and the remaining 80% of the data (i.e., the remaining four folds) are considered the training set. In the following iterations (columns from left to right), a different 20% of the data are considered the test set, while the remaining 80% of the data are considered the training set. The model is fit to the test/training data a chosen number (*K*, or the number of folds) of times, and the prediction error from each model fitting is then averaged to determine the prediction statistics for the model.

The choice of the number of splits (or "folds") to the data is up to the research (hence why this is sometimes called *K-fold cross-validation*), but five and ten splits are used frequently. Additionally, *leave-one-out cross-validation* is when the number of folds is equal to the number of cases in the data set (*K* = *N*). The choice of the number of splits does impact bias (the difference between the average/expected value and the correct value - i.e., error) and variance. Generally, the fewer the number of splits, the lower the variance and the higher the bias/error (and vice versa). More can be learned about this trade-off here (https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff).

We will now walk through an example of using cross-validation to examine the average prediction error of a regression model.

Step 1: Read in data and needed packages.

```
# libraries needed
library(caret)
library(psych)

# read in the data
data <- sat.act
head(data)
```

```
##       gender education age ACT SATV SATQ
## 29442      2         3  19  24  500  500
## 29457      2         3  23  35  600  500
## 29498      2         3  20  21  480  470
## 29503      1         4  27  26  550  520
## 29504      1         2  33  31  600  550
## 29518      1         5  26  28  640  640
```

Step 2: Cross-validation using `caret` package.

We are going to use the `caret` package to predict a participant's ACT score from gender, age, SAT verbal score, and SAT math score using the "sat.act" data from the `psych` package, and assess the model fit using 5-fold cross-validation.

The `caret` package is relatively flexible in that it has functions so you can conduct each step yourself (i.e., split the data, run the model, assess the model performance) or conduct the whole process within one step. In this case, we're going to walk through code that uses fewer steps.

We first set up the number of folds for cross-validation by defining the training control. In this case, we chose 5 folds, but the choice is ulimately up to the researcher.

```
data_ctrl <- trainControl(method = "cv", number = 5)
```

Next, we run our regression model: ACT ~ gender + age + SATV + SATQ.

Note. The fifth line of the following code (na.action = na.pass) will pass the missing values along to the model, in this case linear regression. The missing values will be handled with the default setting of that function, which in the case of the "lm" function is listwise deletion. We do not recommend using listwise deletion because of the MCAR (missing completely at random) assumptions, but do so for the sake of demonstration. Other solutions for dealing with missing data include multiple imputation, etc., which will need to be done in a pre-processing step. Thus, we need complete data for the model we are fitting, not to run the cross-validation (i.e., cross-validation can handle missing data).

```
model_caret <- train(ACT ~ gender + age + SATV + SATQ,   # model to fit
                     data = data,
                     trControl = data_ctrl,              # folds
                     method = "lm",                      # specifying regression model
                     na.action = na.pass)                # pass missing data to model - some models will handle this
```

Examine model predictions.

```
model_caret
```

```
## Linear Regression
##
## 687 samples
##   4 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 561, 560, 559, 560, 560
## Resampling results:
##
##   RMSE      Rsquared
##   3.681462  0.4228643
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
##
```

```
model_caret$finalModel
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Coefficients:
## (Intercept)       gender          age         SATV         SATQ
##     7.93865      0.33397      0.07096      0.01328      0.01657
```

We find that after using 5-fold cross-validation, our model accounts for 42% of the variance (R-squared = 0.418) in ACT scores for these participants.

We can also examine model predictions for each fold.

```
model_caret$resample
```

```
##       RMSE  Rsquared Resample
## 1 3.474389 0.4336992    Fold1
## 2 3.387723 0.4919501    Fold2
## 3 3.608666 0.4109997    Fold3
## 4 3.948821 0.3349530    Fold4
## 5 3.987709 0.4427193    Fold5
```

Furthermore, we can find the standard deviation around the Rsquared value by examining the R-squared from each fold.

```
sd(model_caret$resample$Rsquared)
```

```
## [1] 0.05734466
```

The standard deviation around the R-squared value of the 5-fold cross-validation is 0.06, which is a relatively large window for a Rsquared value.

Just for demonstration, let’s examine what our R-squared values would’ve been if we used:

1) the whole sample, or

2) half of the sample.

Whole sample.

```
whole <- lm(ACT ~ gender + age + SATV + SATQ, data = data)
summary(whole)
```

```
##
## Call:
## lm(formula = ACT ~ gender + age + SATV + SATQ, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1182  -2.1742   0.1535   2.1222  18.7913
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.938650   1.105488   7.181 1.81e-12 ***
## gender      0.333969   0.299643   1.115    0.265
## age         0.070958   0.014842   4.781 2.14e-06 ***
## SATV        0.013278   0.001635   8.122 2.15e-15 ***
## SATQ        0.016572   0.001623  10.208  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.685 on 682 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.4217, Adjusted R-squared:  0.4183
## F-statistic: 124.3 on 4 and 682 DF,  p-value: < 2.2e-16
```

First half of the sample.

```
# randomly select half of the sample
set.seed(0123)
half_size <- floor(0.50 * nrow(data))
random_sample <- sample(seq_len(nrow(data)), size = half_size)
first_half_data <- data[random_sample, ]

# run the model
first_half <- lm(ACT ~ gender + age + SATV + SATQ, data = first_half_data)
summary(first_half)
```

```
##
## Call:
## lm(formula = ACT ~ gender + age + SATV + SATQ, data = first_half_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.8383  -2.4623   0.2748   2.2720  18.8077
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.054951   1.663263   4.843 1.95e-06 ***
## gender      0.496165   0.457700   1.084   0.2791
## age         0.055036   0.021661   2.541   0.0115 *
## SATV        0.013410   0.002419   5.543 5.99e-08 ***
## SATQ        0.015986   0.002525   6.332 7.69e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.912 on 339 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.3773, Adjusted R-squared:   0.37
## F-statistic: 51.36 on 4 and 339 DF,  p-value: < 2.2e-16
```

Second half of the sample.

```
#select half of data not used in the above model
second_half_data <- data[-random_sample, ]

# run the model
second_half <- lm(ACT ~ gender + age + SATV + SATQ, data = second_half_data)
summary(second_half)
```

```
##
## Call:
## lm(formula = ACT ~ gender + age + SATV + SATQ, data = second_half_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8287 -2.0990 -0.0252  1.8764 16.8531
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.500054   1.468866   5.106 5.51e-07 ***
## gender      0.301549   0.391159   0.771    0.441
## age         0.090694   0.020262   4.476 1.04e-05 ***
## SATV        0.012692   0.002202   5.765 1.85e-08 ***
## SATQ        0.017651   0.002087   8.459 8.23e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.432 on 338 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.4772, Adjusted R-squared:  0.471
## F-statistic: 77.11 on 4 and 338 DF,  p-value: < 2.2e-16
```

We can see that the R-squared for the whole sample was approximately equal to the CV results (i.e., both accounted for 42% of the variance in ACT scores), but the R-squared for the randomly selected sample of the first half of the data was a bit smaller (accounted for approximately 38% of the variance of ACT scores) and the R-squared for the remaining second half of the data was larger (accounted for approximately 48% of the variance). This demonstrates the value of using CV, as opposed to solely splitting your data into training and test sets, as a better method of obtaining model estimates.

Finally, when reporting the results of cross-validation, we want to report the accuracy of the cross-validation *and* the parameters from the whole sample.

Step 3: Cautions.

When using cross-validation, there are a few things to consider:

- The choice of *K* (i.e., the number of folds).
    - As mentioned above, there is a trade-off between bias and variance when choosing the number of folds for cross-validation.
    - Sample size may influence the choice of *K* - larger K result in smaller samples within each fold.
- The selection of predictor variables.
    - Do not pre-screen predictors (i.e., do not choose predictors because they are highly correlated with class labels or the outcome variable). Selecting variables based on pre-screening will result in inaccurate prediction errors because you haven already created an unfair advantage for the success of the model.
    - Given a large number of variables, theory should be the ultimate deciding factor in what variables should be included in the model.
- The reporting of results.
    - As mentioned above, the accuracy of cross-validation *and* the parameters from the whole sample should be report.
    - Note that there are cases where these parameters should be cautiously interpreted (e.g., cases of suppression).
- The requirement of IID (independent and identically distributed) data.
    - Cross-validation requires that folds (i.e., test and training sets) are independent. This precludes the use of repeated measures data, dyadic data, etc.

Step 4: Conclusions.

In this tutorial, we walked through an example of using cross-validation for a regression model using the `caret` package. However, cross-validation is applicable to a wide range of classification and regression problems. The use of cross-validation can help social scientists test the robustness of their models given the often limited size of the data available in the field.