Lecture 10: Model evaluation and selection

Stats 32: Introduction to R for Undergraduates

Harrison Li

May 2, 2024

Agenda

1. Predictive accuracy
2. Choosing between nested linear regression models
3. Estimating out-of-sample error
4. Prediction vs. inference

Reading:

Predictive accuracy

Predictive accuracy

Suppose we have a predictive model like (multiple) linear regression, polynomial regression, or locally weighted regression. How do we know which to use for prediction?

A common metric for predictive accuracy when the outcome variable is quantitative and continuous, like in linear regression, is the **mean squared error (MSE)**.

As the name suggests, MSE is the average of the squared differences (“errors”) between the actual data values and the predicted data values. Lower MSE is better.

In-sample vs. out-of-sample error

In-sample error refers to the error for a model in predicting the outcome variable *in sample*, meaning for the data used to fit the model.

By contrast, **out-of-sample error** is the error for the model when used to predict the outcome variable in new data that was NOT used to fit the model.

In-sample vs. out-of-sample error

Recall that in linear regression, a *residual* refers to the difference between the outcome variable and the corresponding based on the predictor variable(s). The OLS algorithm seeks to minimize the sum of the squared residuals on the data it is fit to.

Thus, the in-sample MSE for a linear regression model is simply the *mean squared residual*.

Minimizing the sum of squared residuals is the same as minimizing their mean (just divide by n, the number of observations), so the OLS algorithm in fact minimizes in-sample MSE.

In-sample vs. out-of-sample error

Does minimizing in-sample error necessarily correspond to minimizing out-of-sample error? No!

It's very easy to come up with a model that has zero in-sample error. If all the `x` values in your data are distinct, simply predict `y` to be the actual value corresponding to each `x`, and predict 0 (or any other arbitrary number) for any `x` not in your data.

As you might imagine, this model would not typically do well out-of-sample. Out-of-sample error is what we really care about, if we want to make good predictions.

Choosing between nested linear regression models

Choosing between nested linear regression models

Suppose Alice fits a simple linear regression of y onto x_1, while Bob fits a multiple regression of y onto x_1 and x_2.

Then Alice models $y \approx \beta_0 + \beta_1 \cdot x_1$ while Bob models $y \approx \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2$.

We say Alice's model is **nested** within Bob's model, because any possible configuration (i.e. set of values for β_0 and β_1) of Alice's model corresponds to a possible configuration of Bob's model. That is, Bob can always end up with the same model as Alice (by setting $\beta_2 = 0$).

Nested models

Note that Bob's more flexible model cannot have higher in-sample error than Alice's model. This is because the OLS algorithm for both models finds the model configuration that minimizes in-sample error. But Bob can always choose the same β_0 and β_1 as Alice, and set $\beta_2 = 0$, to get the same in-sample error as Alice.

As we've seen, this doesn't mean Bob's model will have better out-of-sample error (otherwise, we should always add as many predictors as humanly possible!). In general, increased model flexibility will make it more likely that your model fits the noise in the data, rather than the actual "signal". On the other hand, if your model is not flexible enough, it may not capture the full "signal".

Testing nested linear regression models

To help us decide between Alice and Bob's models, we can test the following hypotheses, corresponding to *Bob's* model, using the **linear regression t-test**:

$$H_0 : \beta_2 = 0$$
$$H_1 : \beta_2 \neq 0$$

There is a standard t-test (different from the ones in Lecture 9) that can be used to generate p-values for these hypotheses.

Testing nested linear regression models

The p-value for the linear regression t-test can be found in the output of the regression table from fitting Bob's model, in the row corresponding to x_2. For instance, we consider the `abalone` data and suppose `y` is `Length`, x_1 is `Diameter`, and x_2 is `Height`:

```r
library(tidyverse)
library(moderndive)
abalone <- read_csv("abalone.csv")
model <- lm(Length~Diameter+Height, data=abalone)
```

Testing nested linear regression models

```
summary(model)
```

```
##
## Call:
## lm(formula = Length ~ Diameter + Height, data = abalone)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.096325 -0.010857  0.000186  0.010736  0.071136
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.037480   0.004587   8.171 8.91e-15 ***
## Diameter    1.166147   0.025815  45.173  < 2e-16 ***
## Height      0.085380   0.064002   1.334    0.183
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0191 on 297 degrees of freedom
## Multiple R-squared:  0.9767, Adjusted R-squared:  0.9765
## F-statistic:  6216 on 2 and 297 DF,  p-value: < 2.2e-16
```

With a p value of 0.183, we fail to reject H_0, and select Alice's model.

Anova F test

Now suppose Charlie adds a third predictor to Bob's model, (say x_3 is `Whole.wt`), and we want to decide between Alice's model and Charlie's model, ignoring Bob's model. Then in the language of Charlie's model we test the following hypotheses, which can be done with the **anova F test** implemented by `anova()`:

$$\begin{aligned} H_0 &: \beta_2 = \beta_3 = 0 \\ H_1 &: \beta_2 \neq 0 \text{ or } \beta_3 \neq 0 \end{aligned}$$

Anova F test

```r
alice_model <- lm(Length~Diameter, data=abalone)
charlie_model <- lm(Length~Diameter+Height+Whole.wt, data=abalone)
anova(alice_model, charlie_model)
```

```
## Analysis of Variance Table
##
## Model 1: Length ~ Diameter
## Model 2: Length ~ Diameter + Height + Whole.wt
##   Res.Df     RSS Df Sum of Sq     F   Pr(>F)
## 1    298 0.10905
## 2    296 0.10492  2 0.0041275 5.822 0.003311 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With a p-value of 0.0033, we reject H_0 and select Charlie's model.

Anova F test

You could also use an Anova F-test to compare Alice's model with Bob's model.

In that case it is equivalent to the linear regression *t*-test (gives the same p-value):

```r
bob_model <- lm(Length~Diameter+Height, data=abalone)
anova(alice_model, bob_model)
```

```
## Analysis of Variance Table
##
## Model 1: Length ~ Diameter
## Model 2: Length ~ Diameter + Height
##   Res.Df     RSS Df  Sum of Sq      F Pr(>F)
## 1    298 0.10905
## 2    297 0.10840  1 0.00064953 1.7796 0.1832
```

Estimating out-of-sample error

Estimating out-of-sample error

The hypothesis tests for the nested models above are specialized for the case of linear regression and require lots of assumptions.

A more modern approach to model selection is to directly estimate the out-of-sample error of candidate models, and then pick the model with the lowest such error estimate.

The most basic model selection method is called **data splitting**. As the name suggests, you simply split your data into a **training set** and a **test set**. You fit your models on the training set, and compute the MSE (or other error metric) based on the predictions for the **test set**.

Cross validation

The idea of **K-fold cross validation** improves upon data splitting by averaging the MSE estimated from K different data splits, each of which holds out a different $1/K$ proportion of the full dataset as the test set.

This reduces the noise in the error estimates, i.e. by making them less prone to the randomness in the training/test set split.

K-fold cross validation

1. Split your data at random into K roughly equally sized chunks, or "folds"
2. Treat the first fold as the test set and the other folds as the training set. Compute the out-of-sample errors from fitting your models on the training set and evaluating their errors on the test set
3. Repeat step 2 for all other folds, so that you have a total of K estimates of out-of-sample error for each model. Each time, you select a different fold as the test set, and the other $K - 1$ folds as the training set.
4. Average the K error estimates for each value to get a single estimate.

Model selection

Once you have the cross-validation error from a suite of models of interest, it is principled (and common practice) to simply pick the model with the lowest cross-validation error.

Prediction vs. inference

Prediction vs. inference

These days, there are many fancy machine learning models (neural networks, random forests, etc.) that have surprisingly good predictive accuracy on a wide range of problems.

These models often have lots of parameters (e.g. `span` in locally weighted regression) that are typically chosen using cross-validation or something similar.

However, these models tend to be "black-box", with no obvious way of illuminating the overall structure of the model.

Prediction vs. inference

Prediction is concerned with accurately forecasting unseen data.

Inference is concerned with understanding the overall structure and patterns in your data.

Black-box prediction

Self-driving car algorithms: High predictive accuracy is very important, don't want to sacrifice that for inference.

Medical settings (who benefits most from a drug?): Inference may be more important to help illuminate scientific understanding

Statistics vs. ML

Traditionally, statistics has focused more on inference, and machine learning more on prediction.

Today, the lines are becoming ever more blurred. Take courses in both statistics and machine learning to get the most complete understanding of how to do good data analysis!