Logistic Regression in R – Part Two

September 2, 2015
By atmathew

My previous post covered the basics of logistic regression. We must now examine the model to understand how well it fits the data and generalizes to other observations. The evaluation process involves the assessment of three distinct areas – goodness of fit, tests of individual predictors, and validation of predicted values – in order to produce the most useful model. While the following content isn't exhaustive, it should provide a compact 'cheat sheet' and guide for the modeling process.

Goodness of Fit: Likelihood Ratio Test

A logistic regression is said to provide a better fit to the data if it demonstrates an improvement over a model with fewer predictors. This occurs by comparing the likelihood of the data under the full model against the likelihood of the data under a model with fewer predictors. The null hypothesis, H_0 holds that the reduced model is true,so an α for the overall model fit statistic that is less than 0.05 would compel us to reject H_0.

```
mod_fit_one <- glm(Class ~ Age + ForeignWorker + Property.RealEstate + Housing.Own +
CreditHistory.Critical, data=training, family="binomial")

mod_fit_two <- glm(Class ~ Age + ForeignWorker, data=training, family="binomial")

library(lmtest)
lrtest(mod_fit_one, mod_fit_two)
```

Goodness of Fit: Pseudo R^2

With linear regression, the R^2 statistic tells us the proportion of variance in the dependent variable that is explained by the predictors. While no equivilent metric exists for logistic regression, there are a number of R^2 values that can be of value. Most notable is McFadden's R^2, which is defined as $1 - \frac{ln(L_M)}{ln(L_0)}$ where $ln(L_M)$ is the log likelihood value for the fitted model and $ln(L_0)$ is the log likelihood for the null model with only an intercept as a predictor. The measure ranges from 0 to just under 1, with values closer to zero indicating that the model has no predictive power.

```
library(pscl)
pR2(mod_fit_one) # look for 'McFadden'
```

Goodness of Fit: Hosmer-Lemeshow Test

The Hosmer-Lemeshow test examines whether the observed proportion of events are similar to the predicted probabilities of occurences in subgroups of the dataset using a pearson chi-square statistic from the 2 x g table of observed and expected frequencies. Small values with large p-values indicate a good fit to the data while large values with p-values below 0.05 indicate a poor fit. The null hypothesis holds that the model fits the data and in the below example we would reject H_0.

```
library(MKmisc)
HLgof.test(fit = fitted(mod_fit_one), obs = training$Class)

library(ResourceSelection)
hoslem.test(training$Class, fitted(mod_fit_one), g=10)
```

Tests of Individual Predictors: Wald Test

A wald test is used to evaluate the statistical significance of each coefficient in the model and is calculated by taking the ratio of the square of the regression coefficient to the square of the standard error of the coefficient. The idea is to test the hypothesis that the coefficient of an independent variable in the model is not significantly different from zero. If the test fails to reject the null hypothesis, this suggests that removing the variable from the model will not substantially harm the fit of that model.

```
library(survey)

regTermTest(mod_fit_one, "ForeignWorker")
regTermTest(mod_fit_one, "CreditHistory.Critical")
```

Tests of Individual Predictors: Variable Importance

To assess the relative importance of individual predictors in the model, we can also look at the absolute value of the t-statistic for each model parameter. This technique is utilized by the varImp function in the caret package for general and generalized linear models. The t-statistic for each model parameter helps us determine if it's significantly different from zero.

```
mod_fit <- train(Class ~ Age + ForeignWorker + Property.RealEstate + Housing.Own +
CreditHistory.Critical, data=training, method="glm", family="binomial")

varImp(mod_fit)
```

Validation of Predicted Values: Classification Rate

With predictive models, he most critical metric regards how well the model does in predicting the target variable on out of sample observations. The process involves using the model estimates to predict values on the training set. Afterwards, we will compare the predicted target variable versus the observed values for each observation.

```
pred = predict(mod_fit, newdata=testing)
accuracy <- table(pred, testing[,"Class"])
sum(diag(accuracy))/sum(accuracy)

pred = predict(mod_fit, newdata=testing)
confusionMatrix(data=pred, testing$Class)
```

Validation of Predicted Values: ROC Curve

The receiving operating characteristic is a measure of classifier performance. It's based on the proportion of positive data points that are correctly considered as positive, $TPR = \frac{TP}{n(Y=1)}$, and the proportion of negative data points that are accurately considered as negative, $TNR = \frac{TN}{n(Y=0)}$. These metrics are expressed through a graphic that shows the trade off between these values. Ultimately, we're concerned about the area under the ROC curve, or AUROC. That metric ranges from 0.50 to 1.00, and values above 0.80 indicate that the model does a great job in discriminating between the two categories which comprise our target variable.

```
library(pROC)
# Compute AUC for predicting Class with the variable CreditHistory.Critical
f1 = roc(Class ~ CreditHistory.Critical, data=training)
plot(f1, col="red")

library(ROCR)
# Compute AUC for predicting Class with the model
prob <- predict(mod_fit_one, newdata=testing, type="response")
pred <- prediction(prob, testing$Class)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf)

auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

This post has provided a quick overview of how to evaluate logistic regression models in R. If you have any comments or corrections, please comment below.