Part 3

Sara Altman

12/5/2017

```
library(tidyverse)
library(modelr)
```

```
data <- read_csv("../data/relapse_subjects.csv", na = c("", "NaN"))
```

Prediction on the test set

We didn't have enough data for a test set, so we used all our data in Part 2. Therefore, we use cross-validation to get an estimate of the test error.

Classification

```
test_err <- function(test, mod, threshold) {
  test <- as_tibble(test)

  errors <-
    test %>%
    mutate(
      pred = predict(mod, newdata = test, type = "response"),
      pred_group = ifelse(pred > threshold, 1, 0),
      pred_error = ifelse(relIn6Mos != pred_group, 1, 0)
    )

  return(mean(errors$pred_error))
}

test_err_fn <- function(test, mod, threshold) {
  test <- as_tibble(test)

  errors <-
    test %>%
    mutate(
      pred = predict(mod, newdata = test, type = "response"),
      pred_group = ifelse(pred > threshold, 1, 0),
      false_neg = ifelse(relIn6Mos > pred_group, 1, 0)
    )

  return(mean(errors$false_neg))
}

test_err_fp <- function(test, mod, threshold) {
  test <- as_tibble(test)

  errors <-
```

```
    test %>%
    mutate(
      pred = predict(mod, newdata = test, type = "response"),
      pred_group = ifelse(pred > threshold, 1, 0),
      false_pos = ifelse(relIn6Mos < pred_group, 1, 0)
    )

  return(mean(errors$false_pos))
}

set.seed(1)

by_subj_cv <-
  data %>%
  crossv_kfold(31)

#for threshold = .5
by_subj_cv %>%
  mutate(train = map(train, as_tibble),
         model = map(train, ~glm(relIn6Mos ~ nacc_neutral_beta,
                                 family = "binomial",
                                 data = .)),
         error = map2_dbl(test, model, test_err, threshold = .4),
         false_neg_rate = map2_dbl(test, model, test_err_fn, threshold = .4),
         false_pos_rate = map2_dbl(test, model, test_err_fp, threshold = .4)) %>%
  summarise(mean_error = mean(error),
            mean_fn_rate = mean(false_neg_rate),
            mean_fp_rate = mean(false_pos_rate))
```

```
## # A tibble: 1 x 3
##   mean_error mean_fn_rate mean_fp_rate
##        <dbl>        <dbl>        <dbl>
## 1  0.2258065    0.1129032    0.1129032
```

Our estimated test error is .23.

Regression

[insert hershel's regression cv error calculation]

Inference

(a)

Logistic regression model fitted on all data

```
fit_all_data <- glm(relIn6Mos ~ nacc_neutral_beta,
                    family = "binomial",
                    data = data)

summary(fit_all_data)
```

```
##
## Call:
## glm(formula = relIn6Mos ~ nacc_neutral_beta, family = "binomial",
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7498  -0.7060  -0.4723   0.7799   2.1074
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.3872     0.3967  -0.976  0.32898
## nacc_neutral_beta  21.0739     7.3456   2.869  0.00412 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 50.920  on 38  degrees of freedom
## Residual deviance: 38.645  on 37  degrees of freedom
## AIC: 42.645
##
## Number of Fisher Scoring iterations: 5
```

- The coefficient on nacc_neutral_beta is significant at the $\alpha = .001$ level. The p-value is .004. This means that, assuming the null hypothesis is true, the probability of observing a Wald statistic as extreme as 2.869 is .004. Here, the null hypothesis is that the coefficient on `nacc_neutral_beta` is 0.
- We only have one covariate. If the coefficient on this covariate really is 0, then the chance it will be significant at the $\alpha = .05$ level is 5% (and is 1% at the $\alpha = .01$ level). Therefore, the chance is low that this one covariate woulf be significant if the null were true. In contrast, if we had 100 covariates, we would expect 5 of them to be significant at the $\alpha = .05$ level and so we might not believe our results if it turned out that 5 of our coefficient were significant.

(b)

```
set.seed(1)

by_subj_cv <-
  data %>%
  crossv_kfold(10)

#for threshold = .5
by_subj_cv %>%
  mutate(train = map(train, as_tibble),
         model = map(train, ~glm(relIn6Mos ~ nacc_neutral_beta,
                                 family = "binomial",
                                 data = .)),
         summary = map(model, summary),
         p_value = map_dbl(summary, ~.$coefficients[8]),
         sig_05 = ifelse(p_value < .05, 1, 0),
         sig_01 = ifelse(p_value < .01, 1, 0),
         sig_001 = ifelse(p_value < .001, 1, 0))  %>%
  summarise(prop_sig_05 = mean(sig_05),
```

```
            prop_sig_01 = mean(sig_01),
            prop_sig_001 = mean(sig_001)) %>%
  knitr::kable()
```

prop_sig_05	prop_sig_01	prop_sig_001
1	0.9	0

- We didn't have enough data for a test set and used cross-validation in the prediction part of the project. Therefore, we looked at significance of our coefficient in 10 folds.

- The coefficient on `nacc_neutral_beta` is significant in 100% of the folds at the $\alpha = .05$ level and at and is significant 90% of the time at the $\alpha = .01$ level.
- Note that here, we run 31 hypothesis tests. Therefore, if the null is true, we should expect 5% of our tests (1.55) to be significant at the $\alpha = .05$ level.

(e)

Post selection inference may be a problem here. We fit lots of different models during part 2, and are only reporting our p-values for one such model. Therefore, we may be favorably biasing our p-value. One way to correct for this would be to validate our findings on new data (i.e., the test set). However, as explained previously, we don't have a test set and this option is not available.

Discussion

These models could be used for both predication and inference. The models could be used by clinicians to predict if a stimulant-dependent veteran will relapse after rehabilitation. The models could also be used to make inferences about the neuroscience of addiction.

Our models should hold up well over time. There's no obvious reason to suspect that time will affect the relationship between nucleus accumbens activity and relapsing. However, updating our data and then refitting the regression model after more time has passed would be useful. This is because our current analysis removes those who hadn't relapsed at the time the data set was completed. However, this doesn't mean that they never relapsed. It would be useful to continually follow-up with them (if possible), update our data, and then refit our regression model. Note that this isn't relevant to our current classification model, since that predicts relapse in 6 months.

Users of our models (e.g., clinicians helping stimulant addicts) should be aware that are data set was small and therefore our models are vulnerable to overfitting. Although we used cross-validation to get an estimate of our test error, this is likely still an underestimate of the true test error. Therefore, users of the model should be cautious about making important decisions (e.g., whether or not to give a patient extra treatment or to release them) using our model, and should definitely not rely solely on our models' predictions.

If we could recollect our data, we would try to increase the number of patients in the study. We would also increase the number of brain regions from which activity was recorded. Our current data includes covariates collected during follow-ups (e.g., obstime, relapse). We would add a survey to assess general well-being to the covariates collected during the follow-up.

If we were to analyse the same data set again, we would try a) using all by-trial data to create a hierarchical model (our current models just use our by-subject data set) and b) use survival analysis to model time to relapse.