Logistic Regression Part 3

Adam J Sullivan, PhD

03/05/2018

Logistic Regression: Part 3

Logistic Regression

- We will begin model building with an example.
- We will look at a set of data which considers a self-assessed health exam.
- There is a natural ordering in the health rating so we would assume that the probability of death to change with the categories.
- Our data contains 1600 subjects and they are measured on

Variables

VARIABLE	DESCRIPTION
id	Identification Number
age	Age (years)
male	Sex
	1 = Male
	0 = Female
sah	Self Assessed Health
	1 = Excellent
	2 = Good
	3 = Fair
	4= Poor

VARIABLE	DESCRIPTION
pperf	Physical Performance Scale (0-12)
pefr	Peak Expiratory Flow Rate (average of 3)
cogerr	Number of cognitive errors on SPMSQ
sbp	Systolic Blood Pressure (mmHg)
mile	Ability to Walk 1 mile
	0 = No
	1 = Yes
digit	Use of digitalis
	0 = No
	1 = Yes

VARIABLE	DESCRIPTION
loop	Use of Loop Diuretics
	0 = No
	1 = Yes
untrt	Diagnosed but untreated diabetes
	0 = No
	1 = Yes

VARIABLE	DESCRIPTION
trtdb	Treated Diabetes
	0 = No
	1 = Yes
dead	Dead by 1991
	0 = No
	1 = Yes
time	Follow up (years)

Special Considerations

For Self-Assessed Health Study we need to consider how we model certain variables

- Age: this is critical since subjects were asked to compare their health to others
- Count variables: `pperf` and `cogerr`
- Measured Variables: `sbp` and `pefr`
- Diabetes is in 2 variables `untrt` and `trtdb`
- other binary variables: `male`, `mile`, `loop`, `digit` and `death`

Read in the Data

```
library(haven)
sah <- read_dta("sah.dta")
```

```
## Start:  AIC=1162
## dead ~ sah2 + age + male + pperf + perf + cogerr + sbp + mile +
##     digit + loop + untrt + trtdp
##
##          Df Deviance  AIC
## - age     1     1137 1161
## - digit   1     1137 1161
## - sbp     1     1138 1162
## <none>          1136 1162
## - sah2    1     1139 1163
## - pperf   1     1139 1163
## - perf    1     1140 1164
## - untrt   1     1141 1165
## - mile    1     1144 1168
## - cogerr  1     1145 1169
## - trtdp   1     1146 1170
## - loop    1     1149 1173
## - male    1     1177 1201
##
## Step:  AIC=1161
```

Self-Assessed Health

- With our data we are mainly interested in how well self-assessed health predicts death.
- There are 11 other variables that are possible confounders for this relationship. This means that if we consider all of this we have the possibility of 12 main effects and 11 possible 2 way interactions with self assessed health.
- This could be a very large model and we are only thinking of 2-way interactions here.

Assessing Model Fit

- Once we begin building models we need to know how to assess model fit.
- We have previously covered the likelihood ratio test where we can compare one model versus another.
- Now we move to assessing the fit of one model.
- What affects model fit?
  - Omitted covariates (Main effects, interactions, polynomial terms)
  - Poor choice of link function
  - Unusual subjects

Assessing Model Fit

We usually determine the goodness of fit for logistic regression based on

1. A model is well if the observed and predicted probabilities based on the model are reasonably close.

2. A model has good if the distribution of risk scores for cases and controls separate out.
    - This means Cases tend to have higher scores.
    - This means Controls tend to have lower scores.
    - There is little overlap.

Calibration

- We will use the Hosmer-Lemeshow Goodness of Fit test to assess calibration in logistic regression models:
- With this test we have:

$$\hat{p}_i = \frac{\exp\left(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_K x_{iK}\right)}{1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_K x_{iK}\right)}$$

- where $i = 1, \ldots, n$ and K is the number of covariates in the model.

How Do We Perform it?

- We then reorder the subjects in the order of the fitted probabilities so that

$$\hat{p}_1 < \hat{p}_2 < \cdots < \hat{p}_n$$

- We then group this data in `G` approximately equal sized groups based on the ordered $\hat{p}$.
- The default is typically `G=10` to create deciles of increasing risk.

With G Groups we have:

$$O_j = \sum_{i \in group_j} y_i = \text{ Observed number of events in group } j$$

$$E_j = \sum_{i \in group_j} \hat{p}_i = \text{ Expected number of events in group } j$$

$$\bar{p}_j = \frac{E_j}{n_j} \quad n_j = \text{ number of subjects in group} j = 1, \ldots, G$$

- We then compute the test statistic:

$$X^2_{HL} = \sum_{j=1}^{G} \frac{(O_j - E_J)^2}{n_j \bar{p}_j (1 - \bar{p}_j)} \sim \chi^2_{G-2}$$

Our Test Statistic

- This test statistic is for the following hypothesis

$$H_0 : \text{ The fitted Model is Correct}$$

$$\text{vs}$$

$$H_1 : \text{ The fitted Model is Not Correct}$$

- In order for this to work well $G \geq 6$ and n_j should be large.

	TERM	ESTIMATE	P.VALUE	CONF.LOW	CONF.HIGH
2	sah2	1.218	0.065	0.988	1.504
3	male	3.144	0.000	2.233	4.442
4	pperf	0.937	0.023	0.886	0.991
5	perf	0.998	0.016	0.996	1.000
6	cogerr	1.160	0.001	1.062	1.266
7	sbp	1.006	0.151	0.998	1.013
8	mile	0.566	0.003	0.388	0.826
9	loop	2.423	0.000	1.600	3.635
10	untrt	1.695	0.019	1.079	2.612
11	trtdp	2.119	0.001	1.367	3.234

We can test the calibration of this below:

```
library(ResourceSelection)
hoslem.test(sah$dead, fitted(mod.back.auto), g=10)
```

We get a warning message about factors. When this happens we need to check out data:

```
class(sah$dead)
table(sah$dead)
```

```
## [1] "labelled"
##
##    0    1
## 1373  227
```

We can see that `dead` is a factor without the 0 and 1 that we wanted to have for it. Basically this Stata dataset had labels and those labels are what R imported.

```
sah$dead <- as.numeric(sah$dead)
table(sah$dead)
```

```
##
##    0    1
## 1373  227
```

Our Test Again

```
library(ResourceSelection)
hoslem.test(sah$dead, fitted(mod.back.auto), g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  sah$dead, fitted(mod.back.auto)
## X-squared = 10, df = 8, p-value = 0.2
```

This time we get get a p-value of 0.243. This means our model is a good fit and it is calibrated well. We could try other values of G to make sure:

```
library(ResourceSelection)
hoslem.test(sah$dead, fitted(mod.back.auto), g=6)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  sah$dead, fitted(mod.back.auto)
## X-squared = 3, df = 4, p-value = 0.5
```

```
library(ResourceSelection)
hoslem.test(sah$dead, fitted(mod.back.auto), g=8)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  sah$dead, fitted(mod.back.auto)
## X-squared = 5, df = 6, p-value = 0.6
```

```
library(ResourceSelection)
hoslem.test(sah$dead, fitted(mod.back.auto), g=12)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  sah$dead, fitted(mod.back.auto)
## X-squared = 10, df = 10, p-value = 0.3
```

In each case our p-values are still showing this is a good fit.

Discrimination

We then assess discrimination. To do this we use something called
or

To understand what this is consider 2 different subjects

1. Subject 1 is dead
2. Subject 2 is not dead.

If we consider our model from above it predicts:

1. $\hat{p}_1$ the probability that subject 1 is dead.
2. $\hat{p}_2$ the probability that subject 2 is dead.

The is given by

$$\Pr\left(\hat{p}_1 > \hat{p}_2\right)$$

- If the risk prediction is worthless we find that $C = 0.5$ or essentially the same as flipping a coin.
- If the risk is larger for all who are dead than all who are not dead than we have $C = 1$.

We typically find this value with a Receiver Operating Characteristic (ROC) curve. This curve is:

- For probability p those with $\hat{p} > p$ are predicted to have the outcome and those with $\hat{p} < p$ are predicted to not have the outcome.

Sensitivity and Specificity

-
  - True Positive
  - $\dfrac{\text{Num. with Outcome and } \hat{p} > p}{\text{num. with Outcome}}$
  - $\Pr(\hat{p} > p | \text{Has Disease})$
-
  - True Negative
  - $\dfrac{\text{Num. without Outcome and } \hat{p} < p}{\text{num. with Outcome}}$
  - $\Pr(\hat{p} < p | \text{No Disease})$

ROC Curves

- p is arbitrary.
- If we increase p we increase the Specificity but decrease the Sensitivity.
- An ROC curve has the Sensitivity on the y-axis and 1-Specificity on the x-axis.
- It graphs those points over all choices of p.

ROC Curves

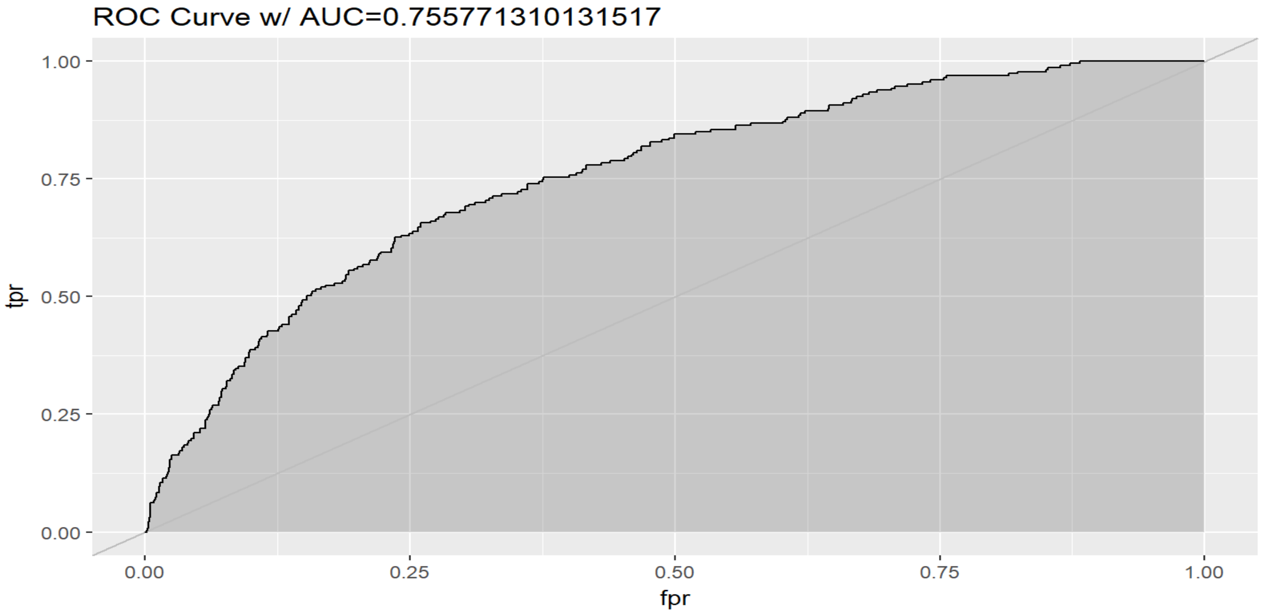
- If the prediction rule is non-discriminatory then the ROC curve will be a straight line from the points (0,0) to (1,1)
- From the ROC curve, it can be shown that the Area under this Curve (AUC) is the same as the C-statistic or

$$AUC = C = \Pr\left(\text{Subject with Disease's Risk} > \text{Subject without Disease's Risk}\right)$$

```
library(ggplot2)
library(ROCR)
```

```
prob <- predict(mod.back.auto)
pred <- prediction(prob, sah$dead)
perf <- performance(pred, "tpr", "fpr")
# I know, the following code is bizarre. Just go with it.
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]

roc.data <- data.frame(fpr=unlist(perf@x.values),
                       tpr=unlist(perf@y.values),
                       model="GLM")
ggplot(roc.data, aes(x=fpr, ymin=0, ymax=tpr)) +
    geom_ribbon(alpha=0.2) + geom_abline(intercept = 0, slope = 1, colour = "gray")+
    geom_line(aes(y=tpr)) +
    ggtitle(paste0("ROC Curve w/ AUC=", auc))
```



ROC Curve w/ AUC=0.755771310131517

We can see with the plot above that we have an AUC of 0.756. This is a decent fit for the model.

AUC	MODEL FIT
0.5	Random
0.6	Mediocre
0.7	Decent
0.8	Good
0.9	Excellent

Calibration vs Discrimination

1. A logistic model with good agreement between observed and predicted outcomes may fail to sharply distinguish between those with a disease and those without.

- If the range of predicted risk across the deciles is narrow.

1. A model that does a good job of distinguishing between those with and without the disease may get risks of events wrong if the calibration is poor.