Inference

October 30, 2017

Data analysis (cont.)

Let's take a more careful look at the model we fit before:

```
affairs <- read.csv("http://koaning.io/theme/data/affairs.csv")
sample_model <-  lm(nbaffairs ~ I(age - 18)*child + factor(religious), data=affairs)
```

We took a look at some values of interest, like the estimated coefficients or the confidence intervals around them. It may also be interesting to take a look at predictions on the original dataset that we used (remember that `sample_model` carries the data used to fit the model).

```
yhat <- predict(sample_model)
head(yhat)
```

```
##     1     2     3     4     5     6
## 2.612 0.177 2.875 1.380 0.638 1.802
```

The `predict` method takes a number of useful arguments, like `newdata`, which applies the estimated coefficients to a new dataset.

```
my_predictions <- predict(sample_model,
                          newdata=data.frame("age"=54, "child"="yes", religious=1))
my_predictions
```

```
##    1
## 3.23
```

Usually, we want to see predictions with their uncertainty. Let's take a look at the documentation to see how to get confidence intervals:

```
?predict
```

Not very useful, right? The reason is that `predict` is a *generic function* that operates on different kinds of objects/models. Think about predictions for a linear model or for a logistic regression. They are still predictions but they are calculated differently and they should be offering different options. But they user should not need to remember the class of the model that was fit: and the end of the day, we have been insisting on the fact that objects in `R` carry a lot of information around. If we look at the bottom of the help file, we will see the method for `lm` models, which is what we want:

```
?predict.lm
```

After this small detour, we finally see how to get the confidence intervals:

```
my_predictions <- predict(sample_model,
                          newdata=data.frame("age"=54, "child"="yes", religious=1),
                          interval="confidence")
my_predictions
```

```
##    fit  lwr upr
## 1 3.23 2.06 4.4
```

A bit more on modeling

We can think about running some other kinds of models on our dataset. For instance, we could think about running a logistic regression.

```
logit_model <-  glm(I(nbaffairs > 0) ~ I(age - 18)*child + factor(religious),
                    data=affairs,
                    family=binomial(link="logit")) # link="logit" is the default
summary(logit_model)
```

```
##
## Call:
## glm(formula = I(nbaffairs > 0) ~ I(age - 18) * child + factor(religious),
##     family = binomial(link = "logit"), data = affairs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.160  -0.824  -0.653  -0.342   2.361
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -1.3960     0.4364   -3.20  0.00138
## I(age - 18)             0.0545     0.0282    1.94  0.05288
## childyes                1.3319     0.4161    3.20  0.00137
## factor(religious)2     -0.7585     0.3524   -2.15  0.03137
## factor(religious)3     -0.4267     0.3561   -1.20  0.23077
## factor(religious)4     -1.3881     0.3606   -3.85  0.00012
## factor(religious)5     -1.2993     0.4394   -2.96  0.00311
## I(age - 18):childyes   -0.0533     0.0306   -1.74  0.08187
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 675.38  on 600  degrees of freedom
## Residual deviance: 638.87  on 593  degrees of freedom
## AIC: 654.9
##
## Number of Fisher Scoring iterations: 4
```

Nothing in the previous call should be odd, we just applied the same logic as before but to a new particular type of model.

One of the things that we could do now is check to what extent the model is performing well. We could take a significance testing approach, but we could also evaluate performance in terms of prediction. We are dealing with a categorical output, so we could for instance check the confusion matrix that is implicit from predicting probabilities:

```
phat <- predict(logit_model, newdata=affairs, type="response")
table(affairs$nbaffairs > 0, phat > 0.5, dnn=list("Observed", "Predicted"))
```

```
##         Predicted
## Observed FALSE TRUE
##    FALSE   451    0
##    TRUE    149    1
```

The model performs poorly, but that's probably because the model predicts low probabilities to a positive event (an affair). We could then play with the probability threshold to have a more realistic confusion matrix:

```
table(affairs$nbaffairs > 0, phat > quantile(phat, .5), dnn=list("Observed", "Predicted"))
```

```
##          Predicted
## Observed FALSE TRUE
##    FALSE   256  195
##    TRUE     49  101
```

Still not a good performance, but still much better than the original matrix we got.

We could also explore the predictors and see their marginal effects. For instance, by checking how the probability of a positive even changes as we move some of the variables on the RHS. One way of accomplishing this is by, for instance, applying our model to a grid of variables:

```
fake_data <- expand.grid(age = c(18, 36, 54, 72),
                         child = c("no", "yes"),
                         religious = 1)
fake_data$prediction <- predict(logit_model, newdata=fake_data, type="response")
fake_data
```

```
##   age child religious prediction
## 1  18    no         1      0.198
## 2  36    no         1      0.398
## 3  54    no         1      0.638
## 4  72    no         1      0.825
## 5  18   yes         1      0.484
## 6  36   yes         1      0.489
## 7  54   yes         1      0.495
## 8  72   yes         1      0.500
```

We did two things here. First, we created a fake dataset by expanding on all the combinations of the values that were passed to `expand.grid`. Then, we applied our predicted model to this new dataset and got the predicted probabilities for each case. Notice I put those predictions back on the fake dataset to be able to see to what combination each prediction corresponds.

We can now see how the change in the probability for different combinations of the age and child variable. But inspecting the model this way may be hard. It is probably better to accomplish this with plots.