MA478 - Lesson 6

Clark

We talked a bit about this last class, but today we're going to talk more about what it means to make inference for a GLM and also talk a little about conducting diagnostic tests.

Recall, a GLM is fit through using maximum likelihood estimates. If you took MA476 you definitely remember that one of the awesome things about MLEs is that they are asymptotically normal. That is, if n is big enough we have:

Therefore, in order to make inference about β we first note that MLEs again give us asymptotic unbiasedness, so we only need an estimate of the variance of $\hat{\beta}$ to take advantage of the asymptotic normality.

Fortunately, to compute this, we can use the weights that we estimated earlier and we have:

However, we see here that not only do we need to estimate the weights (which contain estimates of β) now we also have to estimate ϕ. For some distributions this is trivial. For example for the Poisson $\phi = 1$ and for the Normal distribution $\phi = \sigma^2$. But, how do we do this for a distribution like the Gamma?

Historically MLEs have not been used for ϕ, instead a moment based estimate is computed:

In R this is made unbiased by instead calculating:

```r
library(faraway)
```

```
## Warning: package 'faraway' was built under R version 4.3.2
```

```r
X <- model.matrix(~wool*tension,
                  data=warpbreaks) # wool, tension and interaction variables
y <- warpbreaks$breaks # the number of breaks (count response)


##Gamma



mu <-y
eta <- -1/mu
z <- eta + (y-mu)/mu^2
w <- mu^2
lmod <- lm(z~wool*tension,weights=w,warpbreaks)
coef(lmod)


for(i in 1:5){
  eta <- lmod$fitted.values
  mu <- -1/eta
  z <- eta + (y-mu)/(mu^2)
  w <- mu^2
  lmod <- lm(z~wool*tension,weights=w,warpbreaks)
}


df = length(y)-qr(model.matrix(lmod))$rank

phi <- 1/df*sum((y-mu)^2/mu^2)

r_glm = glm(breaks~wool*tension,data=warpbreaks,
            family=Gamma())

summary(r_glm)
```

Typically, inference isn't done on ϕ (why?)

Once we know the distribution of $\hat{\beta}$ we can use this to answer questions like:

However, if we want to test goodness of fit or significance of the overall model we cannot just rely on computing residuals like we did for the linear model. To see why, let's consider the following. There was an experiment measuring death rates for insects with 30 insects at each of five treatment levels.

```r
data(bliss)

bliss
```

The model we would want to fit:

```r
modl <- glm(cbind(dead,alive) ~ conc,
            family = binomial, data = bliss)
```

Here, our fitted values are:

```r
modl$fitted.values
```

Here, it really wouldn't mean anything to compute $y_i - \hat{y}_i$ as we are not fitting the data directly but rather fitting a function of the estimated mean.

So, for GLMs we have to do something different. One test we could do is a goodness of fit test that looks at the residual deviance, which is the difference between the likelihood and the likelihood of the saturated model. That is:

As long as ϕ is known then the deviance has χ^2 distribution with $n - p$ degrees of freedom

```r
1-pchisq(deviance(modl), df.residual(modl))
```

Now, this doesn't tell us that our model is good, but if our p value is really low it would tell us that our model is poor. Here we have a high p value so there is no evidence of a lack of fit.

Another measure of model fit is the significance of the overall model. This test asks whether the model with predictors fits significantly better than the null model. Again, if we know *phi* (as in a Poisson or logistic regression model) we know the distribution of the difference in likelihood.

```r
anova(modl, test = "Chi")
```

If we add a quadratic term to the model, we can compare the two models and see if the additional term is necessary for the data

```r
modl2 <- glm(cbind(dead, alive) ~ conc + I(conc^2),
             family = binomial, data = bliss)
anova(modl, modl2, test = "Chi")
```

Rememvber what's our null here? Which model is preferable?

Dignostics

Just as in a linear regression we can use the deviance to compute residuals to check for areas of good or poor fit.

We let:

And we define our deviance residuals as:

It is common to standardize the deviance residuals as:

```r
residuals(modl)
```

Let's go through a GLM tooth to tail here:

```r
data(gala)
#help(gala)
head(gala)


gala <- gala[,-2]
modp <- glm(Species ~ ., family = poisson, gala)
```

What model are we fitting?

How would we know that a Poisson is appropriate?

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
g_new = gala %>% mutate(bin_dat =cut(Species,breaks=c(0,50,100,500)))
g_new %>% group_by(bin_dat)%>%summarize(mu=mean(Species),var=var(Species),n=n())
```

Not great, but really hard to tell. Though certainly the variance to mean ratio may be something we dig into more.

Now, look at residuals vs. fitted values. Residuals vs eta-hat

Two things to look at:

1) Any nonlinear relationship between predicted and residuals?

If so, we may need to change our link function, transform thepredictors, change the predictors, etc. Want to try to steer clear of transforming the response to prevent loss of interpretability.

2) Variance of the residuals with respect to the fitted values.We want constant variance in this plot (looks fine here) Maybe a quasi-likelihood or change a different GLM (neg binom) Or use weights for some feature of the data that makes sense.

Relationship between predictors and response

It's helpful to explore how each of the predictors relate to the response. Quick and dirty

```r
pairs(gala)
```

We see Area and Adjacent are very skewed. Luckily, we can do that.

```r
plot(Species ~ log(Adjacent), gala)
```

We can do a transformation that leverages the log link.

```r
mu <- predict(modp, type = "response")
z <- predict(modp) + (gala$Species - mu)/mu
plot(z ~ log(Adjacent), gala,
     ylab ="Linearized Response")
summary(modp)
```

This isn't as pretty as the example in the book? We see Adjacent is significant, so what's wrong? Nothing… while it's sometimes helpful to look at individual predictors, it does not account for the effect of the other predictors on the model.

The Link Function.

Key element here, so we should be able to check it out. I want to use my linearized response from before.

```r
plot(z ~ predict(modp),
     xlab = "Linear Predictor", ylab = "Linear Response")
```

Don't really see an issue here… key word is linear. While we have this link function to help transform our response, we are still working with a linear model… so there should be some evidence of a linear relationship between

Checking for Unusual Points.

For a GLM, we do not expect the residuals to be normally distributed, so using the QQ plot doesn't work. Instead, we use a half-normal plot that compares the sorted absolute residuals and the quantiles.

```r
halfnorm(rstudent(modp))
```

Uses the jack-knife residual… they have a mean near 0 and a variance slightly greater than 1. The key part of this is it leaves out the residual in question when standardizing it (Variance is based on all others). There are no signs of outliers in the plot.

We can use the half-norm for influence and leverage, too.

```r
infl.gal <- influence(modp)
halfnorm(infl.gal$hat)
```

Look at the plot… 12 and 16 may have some influence on the model