Homework 7 - Predictive Modeling in Finance and Insurance

Dennis Goldenberg

2024-03-25

1. Generalized Linear Model

a. Showing pdf belongs to exponential family

I can modify the p.d.f and show:

$$\begin{aligned} f(y;\theta) &= \theta y^{-\theta-1} \\ &= \theta e^{\ln(y^{-\theta-1})} \\ &= \theta e^{-\ln(y)*\theta-\ln(y)} \\ &= e^{\ln(\theta)} * e^{-\ln(y)*\theta-\ln(y)} \\ &= e^{-\ln(y)\theta+\ln(\theta)-\ln(y)} \end{aligned}$$

Thus, we have that $a(y) = -\ln(y)$, $b(\theta) = \theta$, $c(\theta) = \ln(\theta)$, $d(y) = -\ln(y)$.

b. Natural Exponential family or Exponential Dispersion Family

Note that $a(y) \neq y$; thus the Pareto distribution is NOT part of the natural exponential family. However, if you restructure the p.d.f:

$$f(y;\theta) = e^{-\ln(y)\theta+\ln(\theta)-\ln(y)} = e^{-(\theta+1)\ln(y)+(1)\ln(\theta)}$$

You can note that the Pareto distribution is also part of the Exponential dispersion family, with $\lambda = 1$.

c. Score statistic

To find the score statistic, I need log-likelihood function:

$$\ell(\theta;y) = \ln(f(y;\theta)) = -\ln(y)*\theta+\ln(\theta)-\ln(y)$$

The score function is just the derivative of the log-likelihood:

$$S(\theta;y) = \frac{d}{d\theta}\ell(\theta;y) = -\ln(y)+\frac{1}{\theta}$$

As this Pareto is in the EDF, $\mathbb{E}[S(\theta;y)] = 0$.\ The Fischer information is simply the expectation of the second derivative:

$$i(\theta) = -\mathbb{E}\left[\frac{d^2}{d\theta^2}\ell(\theta;y)\right] = -\mathbb{E}\left[-\frac{1}{\theta^2}\right] = \frac{1}{\theta^2}$$

d. Pareto distribution and Exponential Family

i. Does it belong to the exponential family?

ii. Is it a GLM?

This model is **NOT** a generalized linear model. The model is NOT linear in its parameters, so the systematic component does not match the $x^T\beta$ linear format.

2. Logistic Regression

Given the model, I have $\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + 1(\beta_{>1}) + 1(\beta_{[0\%,10\%]})$. Therefore:

$$\begin{aligned}
&\log\left(\frac{\pi}{1-\pi}\right) = 1.53 + 0.735 - 0.031 = 2.234 \\
&\to \frac{\pi}{1-\pi} = e^{2.234} \\
&\to \pi = (1-\pi)e^{2.234} \\
&\to \hat{\pi} = \frac{e^{2.234}}{1+e^{2.234}} = \mathbf{0.9033}
\end{aligned}$$

3. Logistic Regression Part 2

a. Coefficients from Models

I generate the data by creating a data frame, and then run the model:

```r
anther <- c(0,0,0,0,0,0,1,1,1,1,1,1)
treatment <- c(0,0,0,1,1,1,0,0,0,1,1,1)
freq <- c(102-55,99-52,108-57,76-55,81-50,90-50,
          55,52,57,55,50,50)
force <- c(40,150,350,40,150,350,40,150,350,40,150,350)
dataEmb <- data.frame(cbind(anther, treatment, force, freq))
dataEmb$treatment <- factor(dataEmb$treatment)

# for Model 1
model1 <- glm('anther ~ treatment + force + treatment*force',
              data = dataEmb, family = "binomial"(link = 'logit'),
              weights = freq)
coef1 <- summary(model1)$coefficients[,1]

# for Model 2
model2 <- glm('anther ~ treatment + force',
              data = dataEmb, family = "binomial"(link = 'logit'),
              weights = freq)
coef2 <- summary(model2)$coefficients[,1]

# for Model 3
model3 <- glm('anther ~ force',
              data = dataEmb, family = "binomial"(link = 'logit'),
              weights = freq)
coef3 <- summary(model3)$coefficients[,1]
print("Model 1 Coefficients:")
```

```
## [1] "Model 1 Coefficients:"
```

```r
print(coef1)
```

```
##      (Intercept)       treatment1            force treatment1:force
##     0.1456719125     0.7963143307    -0.0001227259    -0.0020493450
```

```r
print("Model 2 Coefficients:")
```

```
## [1] "Model 2 Coefficients:"
```

```r
print(coef2)
```

```
##   (Intercept)    treatment1         force
##  0.3066430701  0.4055543471 -0.0009970257
```

```r
print("Model 3 Coefficients:")
```

```
## [1] "Model 3 Coefficients:"
```

```r
print(coef3)
```

```
##   (Intercept)         force
##  0.4759286430 -0.0009553572
```

b. Probability Estimates from models

To calculate the probability estimates, I note that:

$$\hat{\pi} = \frac{e^{x^T\hat{\beta}}}{1 + e^{x^T\hat{\beta}}}$$

From this, I calculate the probability estimates from the models

```
# for model 1
x_s = dataEmb[1:6,2:3]
x_s$inter <- x_s[,2]*c(0,0,0,1,1,1)
desMat <- data.matrix(cbind(1,x_s))
exTB1 <-exp(desMat %*% matrix(coef1))
pi_1 <- exTB1/(1 + exTB1)
pi_1 <- round(c(pi_1),4)

#for model 2
x_s2 = dataEmb[1:6,2:3]
desMat2 <- data.matrix(cbind(1,x_s2))
exTB2 <-exp(desMat2 %*% matrix(coef2))
pi_2 <- exTB2/(1 + exTB2)
pi_2 <- round(c(pi_2),4)

#for model 3
x_s3 = dataEmb[1:3,3]
desMat3 <- data.matrix(cbind(1,x_s3))
exTB3 <-exp(desMat3 %*% matrix(coef3))
pi_3 <- exTB3/(1 + exTB3)
pi_3 <- round(c(pi_3, pi_3),4)
```

I then create a matrix to show the $\hat{\pi}|\text{force}, \text{treatment}$ under the 6 different possible conditions, where control $= C$, treatment $= T$:

```
pred_pis <- t(cbind(pi_1, pi_2,pi_3))
colnames(pred_pis) <- c("40, C","150, C","350, C",
                        "40, T", "150,T", "350, T")
rownames(pred_pis) <- c("Model 1", "Model 2", "Model 3")
pred_pis
```

```
##          40, C 150, C 350, C  40, T  150,T 350, T
## Model 1 0.7185 0.7158 0.7108 0.8391 0.8042 0.7267
## Model 2 0.6620 0.6371 0.5898 0.7461 0.7248 0.6833
## Model 3 0.6077 0.5824 0.5353 0.6077 0.5824 0.5353
```

c. Expeced Values from 3 Models

```
data.matrix(pred_pis) * c(102,99,108,76,81,90)
```

```
##           40, C  150, C  350, C   40, T   150,T  350, T
## Model 1 73.2870 54.4008 72.5016 63.7716 82.0284 55.2292
## Model 2 65.5380 51.6051 58.3902 60.4341 71.7552 55.3473
## Model 3 65.6316 52.4160 57.8124 54.6930 62.8992 48.1770
```

d. Pearson residuals

e. Goodness of fit statistics

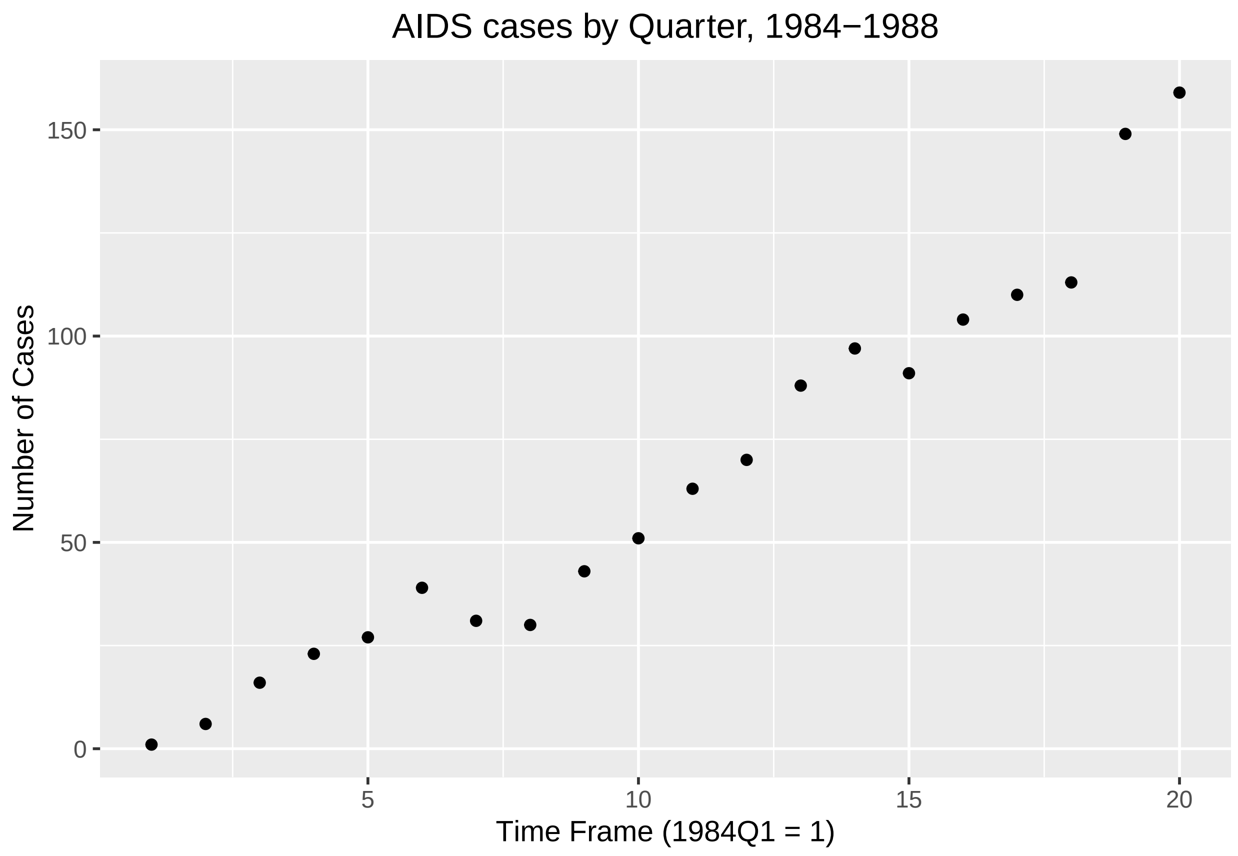
4. Maximum Likelihood Estimation approximation

```r
library(ggplot2)
```

```r
Year <- c(1984, 1985, 1986, 1987, 1988)
Q1 <- c(1,27,43,88,110)
Q2 <- c(6, 39, 51, 97, 113)
Q3 <- c(16, 31, 63, 91, 149)
Q4 <- c(23, 30, 70, 104, 159)
dataAIDS <- data.frame(cbind(Year, Q1,Q2,Q3,Q4))
```

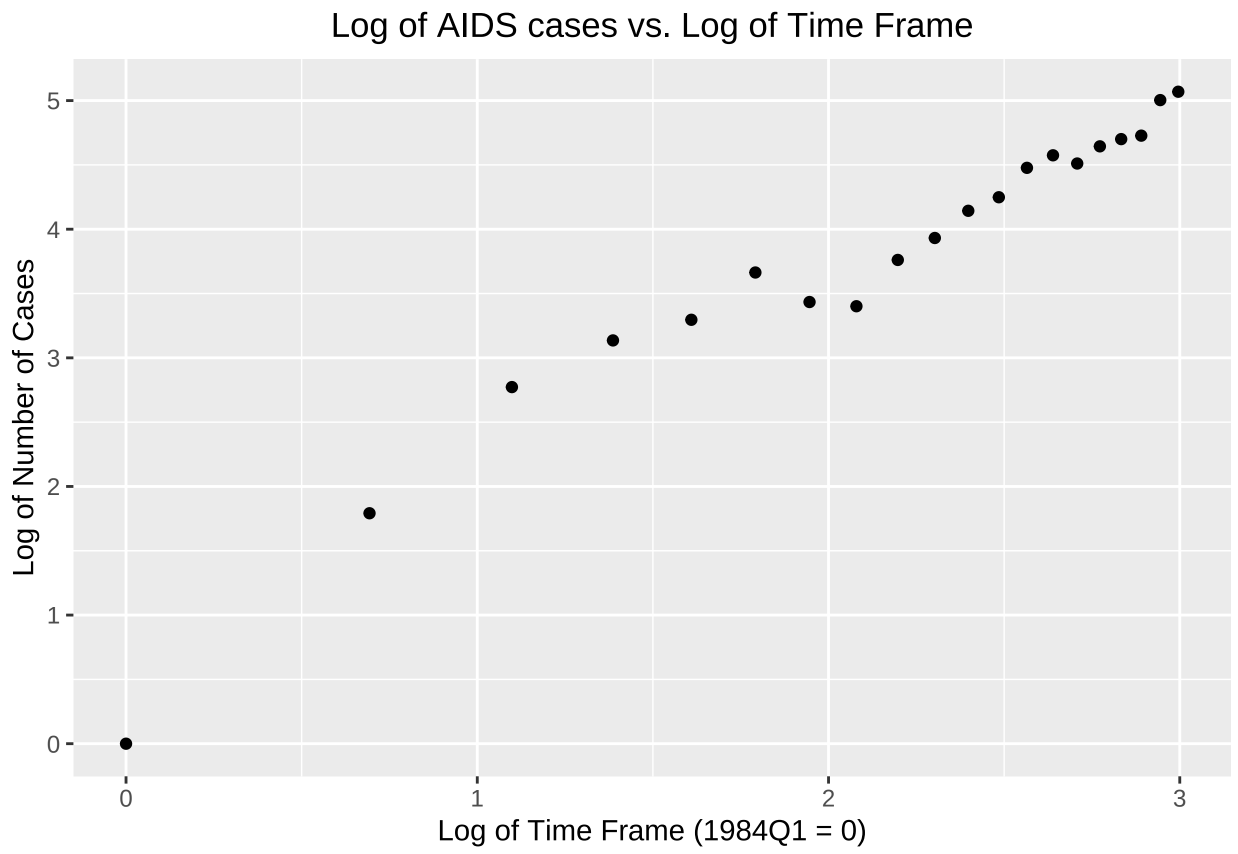
a. Plot of number of cases against time period

```r
qMat <- t(data.matrix(dataAIDS[,2:5]))
timeFrame <- data.frame(cbind(1:(dim(qMat)[1] * dim(qMat)[2]),as.vector(qMat)))
ggplot(data = timeFrame) + geom_point(aes(X1,X2)) +
  xlab("Time Frame (1984Q1 = 1)") + ylab ("Number of Cases") +
  ggtitle("AIDS cases by Quarter, 1984-1988") +
  theme(plot.title = element_text(hjust = 0.5))
```



b. Plot against log i

```r
timeFrameLog <- data.frame(cbind(log(timeFrame$X1),log(timeFrame$X2)))
ggplot(data = timeFrameLog) + geom_point(aes(X1,X2)) +
  xlab("Log of Time Frame (1984Q1 = 0)") + ylab ("Log of Number of Cases") +
  ggtitle("Log of AIDS cases vs. Log of Time Frame") +
  theme(plot.title = element_text(hjust = 0.5))
```



There seems to be a linear relationship between $\log(y_i)$ and $\log(i)$, suggesting that this model may be appropriate.

c. Fitting GLM

Note that, here, we are using the canonical link function for the poisson distribution. Therefore:

$$g'(\mu)V(\mu) = 1 \to \mathbf{W} = \text{diag}\{1\} \text{ and } \theta = \lambda = g^{-1}(x^T\beta) = e^{x^T\beta}$$

I find the log-likelihood and Score function of β:

$$L(\lambda; y) = \prod_{i=1}^{n} \frac{e^{-\lambda_i} * \lambda_i^{y_i}}{y_i!} = e^{-\sum_{i=1}^{n}\lambda_i} * \frac{\prod_{i=1}^{n}\lambda_i^{y_i}}{\prod_{i=1}^{y_i}}$$

$$\to \ell(\lambda; y) = -\sum_{i=1}^{n}\lambda_i + \sum_{i=1}^{n} y_i \ln(\lambda_i) - \sum_{i=1}^{n}\ln(y_i!)$$

$$\to \ell(\beta; y) = -\sum_{i=1}^{n} e^{x_i^T\beta} + \sum_{i=1}^{n} y_i(x_i^T\beta) - \sum_{i=1}^{n}\ln(y_i!)$$

$$\to S(\beta; y) = \begin{bmatrix} \frac{d\ell}{d\beta_0} \\ \frac{d\ell}{d\beta_1} \end{bmatrix} = \begin{bmatrix} -\sum_{i=1}^{n} e^{x_i^T\beta} + \sum_{i=1}^{n} y_i \\ -\sum_{i=1}^{n} x_i e^{x_i^T\beta} + \sum_{i=1}^{n} x_i y_i \end{bmatrix}$$

Further, from the slides, again from the canonical link:

$$i(\beta, y) = x^T \text{diag}\{V(\mu_i)\}\, x = x^T \text{diag}\left\{e^{x_i^T\beta}\right\} x$$

Before beginning, it is important to note:

```r
sum(timeFrame$X2)
```

```
## [1] 1311
```

```r
sum(timeFrameLog$X1 * timeFrame$X2)
```

```
## [1] 3396.379
```

So, $\sum_{i=1}^{20} \log(i) = 1311$ and $\sum_{i=1}^{20} y_i \log(i) = 3396.379$. Therefore:

$$S(\beta; y) = \begin{bmatrix} -\sum_{i=1}^{n} e^{x_i^T \beta} + 1311 \\ -\sum_{i=1}^{n} x_i e^{x_i^T \beta} + 3396.379 \end{bmatrix}$$

Now for the initial step. I do some preliminary calculations:

```r
e_xi <- sum(exp(timeFrameLog$X1))
x_ie_exi <- sum(exp(timeFrameLog$X1) %*% timeFrameLog$X1)
e_xi
```

```
## [1] 210
```

```r
x_ie_exi
```

```
## [1] 529.6022
```

```r
i_0 <- t(data.matrix(timeFrameLog)) %*% diag(exp(timeFrameLog$X1)) %*%
  data.matrix(timeFrameLog)
i_0
```

```
##          X1       X2
## X1 1386.477 2362.725
## X2 2362.725 4037.512
```

- I start with $\beta_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$.
- $S_0(\beta_0; y) = \begin{bmatrix} -\sum_{i=1}^{20} e^{x_i} + 1311 \\ -\sum_{i=1}^{n} x_i e^{x_i} + 3396.379 \end{bmatrix} = \begin{bmatrix} -210 + 1311 \\ -529 + 3396.379 \end{bmatrix} = \begin{bmatrix} 1101 \\ 2866.7768 \end{bmatrix}$.
- $i_0(\beta_0) = x^T \mathrm{diag}\left\{e^{x_i}\right\} x = \begin{bmatrix} 1386.477 & 2362.725 \\ 2362.725 & 4037.512 \end{bmatrix}$
- $W = \mathrm{diag}\left\{1\right\}$ as previously mentioned, and

I iterate once for step 2:

```r
invI_0 <- solve(i_0)
beta_1 <- c(0,1) + invI_0%*%c(-1 * e_xi + sum(timeFrame$X2),
                               -1 * x_ie_exi + sum(timeFrameLog$X1 * timeFrame$X2))
rownames(beta_1) <- c("b_0","b_1")
beta_1
```

```
##           [,1]
## b_0 -150.68715
## b_1   89.89113
```

- $\beta_1 = \begin{bmatrix} -150.687 & 89.891 \end{bmatrix}$.
-

5. Deviance

a. Calculating Deviance

b. Calculating Pearson residual

c. Calculating deviance residual

6. Nominal Regression

a. Odds ratio

b. Probability Calculation

7. Ordinal Regression