Sta 531 HW5 (The one due right after spring break)

Daniel Truver

3/18/2018

(1) Some weird Bernoulli stuff

Let Z_i be iid $N(\alpha, \sigma^2)$ and $X_1, \dots, X_n$ be independent Bernoulli random variables given by:

$$X_i = \begin{cases} 0 & Z_i \leq u \\ 1 & Z_i > u \end{cases}$$

(a) Likelihood

$$\begin{aligned} L(p) &= \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} \\ &= p^{\sum x_i}(1-p)^{\sum(1-x_i)} \\ &= p^{\sum x_i}(1-p)^{n-\sum x_i} \end{aligned}$$

And, as for p,

$$p = Pr(X_i = 1) = Pr(Z_i > u) = \int_u^\infty N(x; \alpha, \sigma^2) = \Phi\left(\frac{\alpha - u}{\sigma}\right),$$

where Φ denotes the cdf of the normal distribution.

(b) Complete-Data Log Likelihood

The log-likelihood of the z_i's is the standard normal log-likelihood.

$$l^C(\alpha, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum(Z_i - \alpha)^2$$

We then take the expectation conditional on the x_i's and use the book's notation to denote this as Q.

$$\begin{aligned} Q(\alpha, \sigma^2) &= E\left[-\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum(Z_i - \alpha)^2 \mid x\right] \\ &= -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum E\left[(Z_i - \alpha)^2 \mid x_i\right] \\ &= -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum\left[E(Z_i^2 \mid x_i) - 2\alpha E(Z_i \mid x_i) + \alpha^2\right] \end{aligned}$$

(c) The M-step

We want

$$
\begin{aligned}
\arg\max_\alpha Q &= \arg\max_\alpha \sum \left[2\alpha E(Z_i \mid x_i, \hat{\alpha_{(j)}}, \hat{\sigma^2_{(j)}}) - E(Z_i^2 \mid x_i, \hat{\alpha_{(j)}}, \hat{\sigma^2_{(j)}}) - \alpha^2\right] \\
&= \arg\max_\alpha \sum \left[2\alpha E(Z_i \mid x_i, \hat{\alpha_{(j)}}, \hat{\sigma^2_{(j)}})\right] - n\alpha^2 \\
&= \frac{1}{n} \sum E(Z_i \mid x_i, \hat{\alpha_{(j)}}, \hat{\sigma^2_{(j)}}) \\
&= \hat{\alpha}_{(j+1)}
\end{aligned}
$$

where we obtain the last equality by setting the derivative with respect to α equal to 0. We'll do the same for σ^2.

$$
\begin{aligned}
0 &= -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum (v_i - 2\alpha t_i + \alpha^2) \\
\sigma^2 &= \frac{1}{n}\left[\sum v_i - 2\alpha \sum t_i + n\alpha^2\right] \\
&= \frac{1}{n}\left[\sum v_i - (\frac{1}{n}\sum t_i)^2\right] \quad \text{(from above)}
\end{aligned}
$$

(2) Why we gotta classify the animals? It's 2018, let them be who they want.

(a) Likelihood

$$
\begin{aligned}
L(Y_1 \mid \theta) &= \frac{197!}{125!\ 18!\ 20!\ 34!} \left(\frac{2+\theta}{4}\right)^{125} \left(\frac{1-\theta}{4}\right)^{18} \left(\frac{1-\theta}{4}\right)^{20} \left(\frac{\theta}{4}\right)^{34} \\
L(Y_2 \mid \theta) &= \frac{20!}{14!\ 0!\ 1!\ 5!} \left(\frac{2+\theta}{4}\right)^{14} \left(\frac{1-\theta}{4}\right) \left(\frac{\theta}{4}\right)^{5}
\end{aligned}
$$

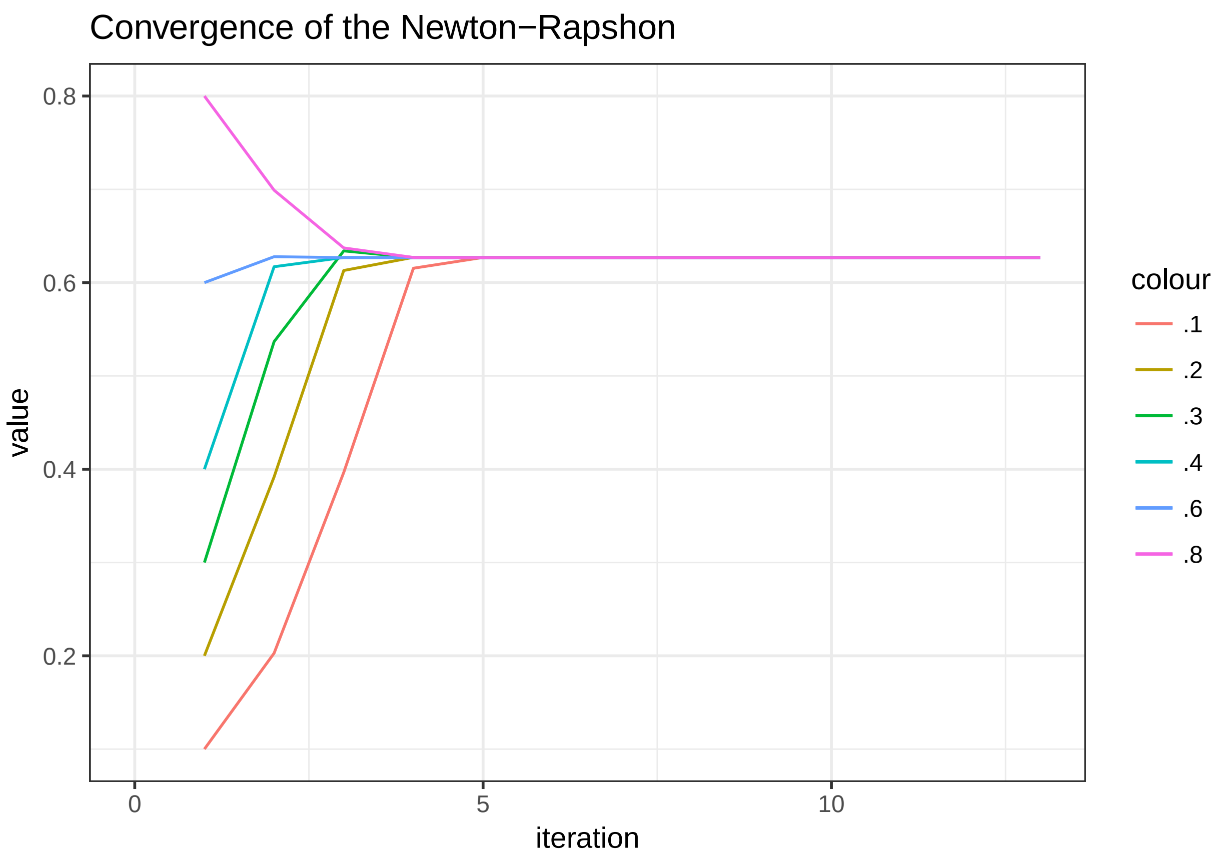
(b) Newton-Raphson

Note that when we take the log-likelihood, we will ignore the constant at the front and instead maximize:

$$
l(Y_1 \mid \theta) = 125 \log\left(\frac{2+\theta}{4}\right) + 18 \log\left(\frac{1-\theta}{4}\right) + 20 \log\left(\frac{1-\theta}{4}\right) + 34 \log\left(\frac{\theta}{4}\right)
$$

```
likeB = function(theta){
  125*log((2+theta)/4) + 18*log((1-theta)/4) + 20*log((1-theta)/4) + 34*log(theta/4)
}
dlikeB = function(theta){
  125/4 * 4/(2+theta) - 18/4 * 4/(1-theta) - 20/4 * 4/(1-theta) + 34/4 * 4/theta
}
d2likeB = function(theta){
  -125/(2+theta)^2 - 18/(1-theta)^2 - 20/(1-theta)^2 - 34/(theta)^2
}
resB = NULL
for (theta_0 in c(.1,.2,.3,.4,.6,.8)){
  x = c(theta_0,rep(NA, 100))
  for (j in 1:100){
    x[j+1] = x[j] - dlikeB(x[j])/d2likeB(x[j])
  }
  resB = cbind(resB, x)
}
```

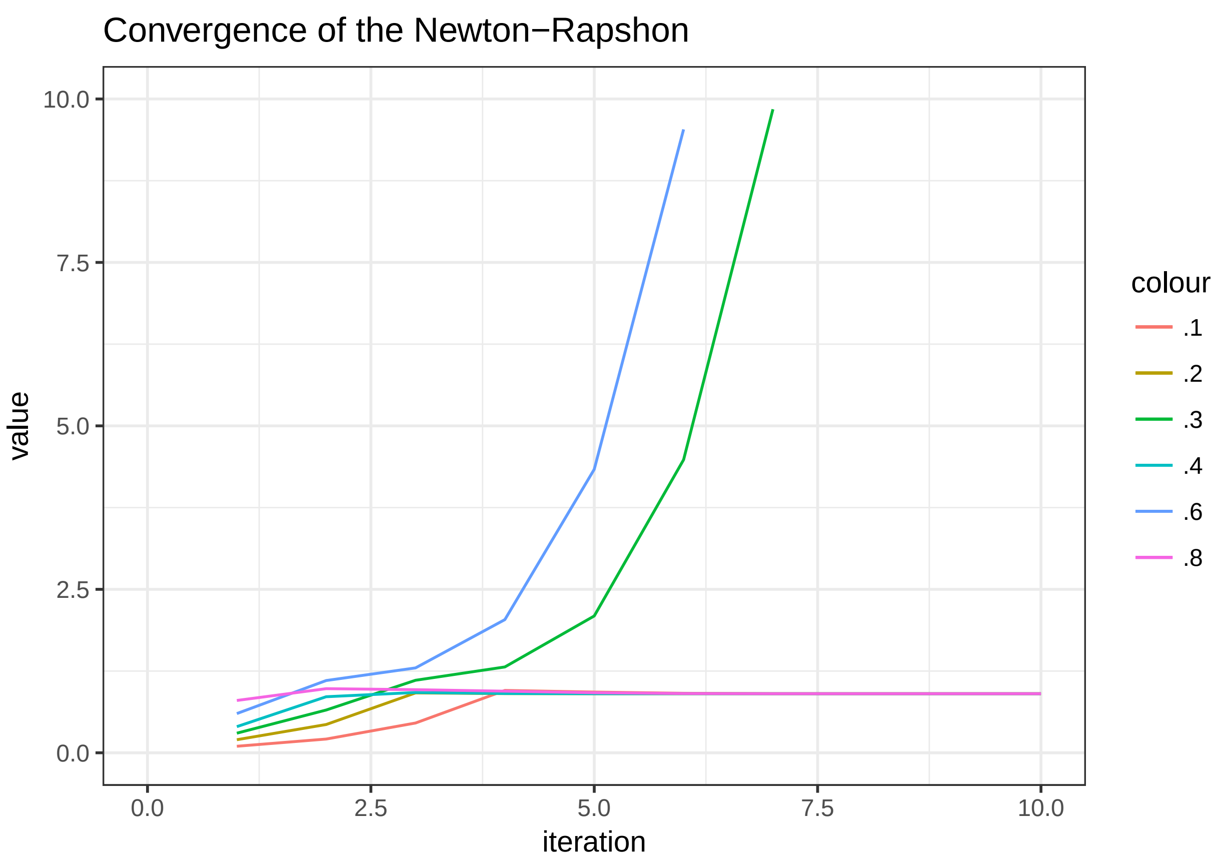
```r
library(dplyr)
resB = data.frame(iteration = 1:101, resB)
colnames(resB) = c("iteration", "x.1", "x.2", "x.3", "x.4", "x.6", "x.8")
library(ggplot2)
ggplot(data = resB, aes(x = iteration)) +
  geom_line(aes(y = x.1, color = ".1")) +
  geom_line(aes(y = x.2, color = ".2")) +
  geom_line(aes(y = x.3, color = ".3")) +
  geom_line(aes(y = x.4, color = ".4")) +
  geom_line(aes(y = x.6, color = ".6")) +
  geom_line(aes(y = x.8, color = ".8")) +
  theme_bw() +
  xlim(0,13) +
  ggtitle("Convergence of the Newton-Rapshon") +
  ylab("value")
```



(c) Newton-Raphson again

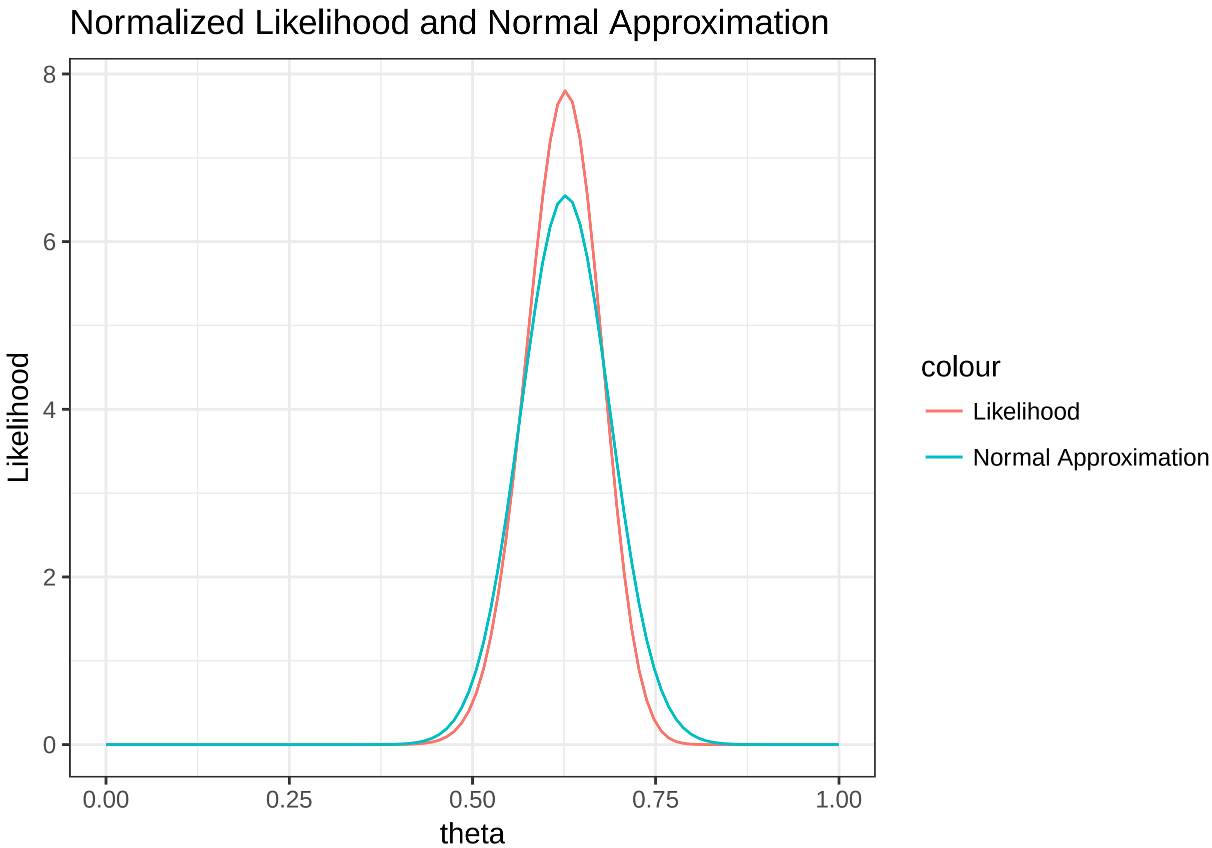
```r
likeC = function(theta){
  14*log((2+theta)/4)  + 1*log((1-theta)/4) + 5*log(theta/4)
}
dlikeC = function(theta){
  14/4 * 4/(2+theta) - 1/4 * 4/(1-theta) + 5/4 * 4/theta
}
d2likeC = function(theta){
  -14/(2+theta)^2 -1/(1-theta)^2 - 5/(theta)^2
```

```r
}
resC = NULL
for (theta_0 in c(.1,.2,.3,.4,.6,.8)){
  x = c(theta_0,rep(NA, 100))
  for (j in 1:100){
    x[j+1] = x[j] - dlikeC(x[j])/d2likeC(x[j])
  }
  resC = cbind(resC, x)
}
library(dplyr)
resC = data.frame(iteration = 1:101, resC)
colnames(resC) = c("iteration", "x.1", "x.2", "x.3", "x.4", "x.6", "x.8")
library(ggplot2)
ggplot(data = resC, aes(x = iteration)) +
  geom_line(aes(y = x.1, color = ".1")) +
  geom_line(aes(y = x.2, color = ".2")) +
  geom_line(aes(y = x.3, color = ".3")) +
  geom_line(aes(y = x.4, color = ".4")) +
  geom_line(aes(y = x.6, color = ".6")) +
  geom_line(aes(y = x.8, color = ".8")) +
  theme_bw() +
  xlim(0,10) +
  ylim(0,10) +
  ggtitle("Convergence of the Newton-Rapshon") +
  ylab("value")
```



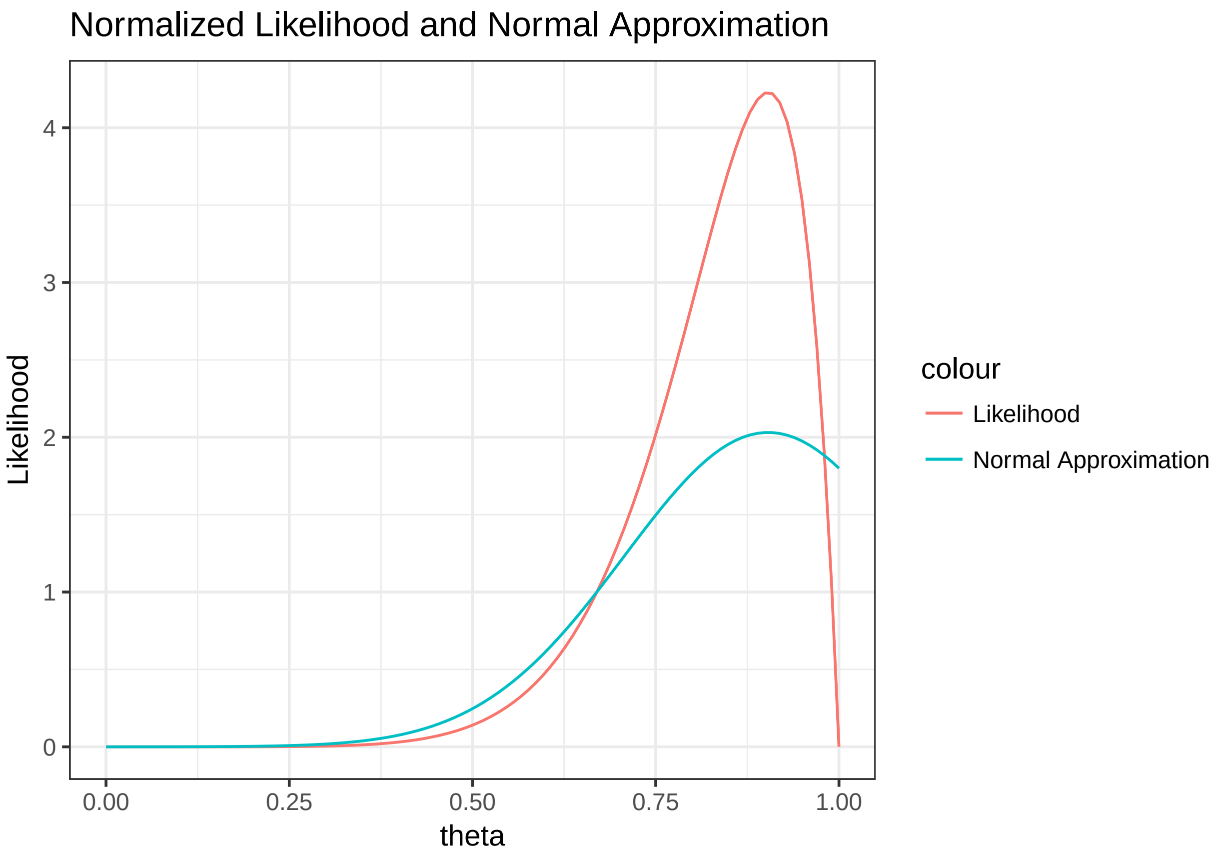
We see that 0.6 and 0.3 starting points do not lead to convergence of the Newton-Raphson algorithm in this case.

(d) Assymptotic Normality



This normal approximation looks pretty good.

(d) Assymptotic Normality round 2



The normal approximation does not look so good here.