MKTG776 HW4

Jordan Farrer

2017-02-15

Contents

1 Question 1

We first load the toothpaste dataset:

```
pacman::p_load(tidyverse, pander, ggrepel, stringr)
panderOptions('round', 4)
panderOptions('keep.trailing.zeros', TRUE)
options(scipen = 10, expressions = 10000, digits = 4)

toothpaste_data <- readxl::read_excel("HW toothpaste data.xlsx")

toothpaste_data %>%
  pander(caption = "Raw Toothpaste Data")
```

Table 1: Raw Toothpaste Data

x	N_x
0	2212
1	383
2	154
3	91
4	89
5	106

Then we implement the beta-binomial distribution using the following funcions:

```
fn_bb <- function(x, m, alpha, beta, pi, inflated_at = 0) {
  p_x <- choose(m, x) * beta(alpha + x, beta + m - x) / beta(alpha, beta)
  if(x == inflated_at) {
    return(pi + (1 - pi) * p_x)
  } else {
    return((1 - pi) * p_x)
```

```
  }
}

fn_max_ll <- function(par, inflated = FALSE, x, N, m, inflated_at) {
  alpha <- par[1]
  beta <- par[2]
  if (inflated) {
    pi <- par[3]
  } else {
    pi <- 0
  }

  p_x <- map_dbl(x, .f = fn_bb, m, alpha, beta, pi, inflated_at)

  ll <- sum(N * log(p_x))

  return(-ll)
}

par_bb <- nlminb(c(1, 1), fn_max_ll, lower = c(0, 0), upper = c(Inf, Inf),
                 inflated = FALSE, x = toothpaste_data$x, N = toothpaste_data$N_x,
                 m = 5, inflated_at = 0)
par_bb_zi <- nlminb(c(1, 1, .5), fn_max_ll, lower = c(0, 0, 0), upper = c(Inf, Inf, 1),
                 inflated = TRUE, x = toothpaste_data$x, N = toothpaste_data$N_x,
                 m = 5, inflated_at = 0)
par_bb_onei <- nlminb(c(1, 1, .5), fn_max_ll, lower = c(0, 0, 0), upper = c(Inf, Inf, 1),
                 inflated = TRUE, x = toothpaste_data$x, N = toothpaste_data$N_x,
                 m = 5, inflated_at = 1)


bb_params <-
  data_frame(
    model = c("Beta-Binomial", "Zero-Inflated Beta-Binomial", "One-Inflated Beta-Binomial")
    , alpha = c(par_bb$par[1], par_bb_zi$par[1], par_bb_onei$par[1])
    , beta = c(par_bb$par[2], par_bb_zi$par[2], par_bb_onei$par[2])
    , pi = c(NA, par_bb_zi$par[3], par_bb_onei$par[3])
  ) %>%
  mutate(
    model = factor(model, levels = c("Beta-Binomial", "Zero-Inflated Beta-Binomial", "One-Inflated Beta-Binomial"))
  )
```

Below is a summary of each of model parameters for the 3 beta-binomial models fitted to the data.

Table 2: Model Parameters for 3 variants of Beta-Binomial

model	alpha	beta	pi
Beta-Binomial	0.1419	0.9881	
Zero-Inflated Beta-Binomial	0.1419	0.9881	0.0000
One-Inflated Beta-Binomial	0.1013	0.7523	0.0485

1.1 Model Selection

In order to select the "best" model we will use

1. Graphical review of the results
2. Goodness of Fit test

First we find the expected number of panelists (out of 3,035) that would have purchases the focal brand m times out of 5.

```
bb_expected <-
  bb_params %>%
    replace_na(list(pi = 0)) %>%
    bind_cols(data_frame(inflated_at = c(0,0,1))) %>%
    crossing(toothpaste_data) %>%
    rowwise() %>%
    mutate(p_x = map_dbl(x, .f = fn_bb, m = 5, alpha, beta, pi, inflated_at)) %>%
    group_by(model) %>%
    mutate(expected = p_x * sum(N_x)) %>%
    ungroup() %>%
    mutate(chisq = (N_x - expected)^2 / expected)
```

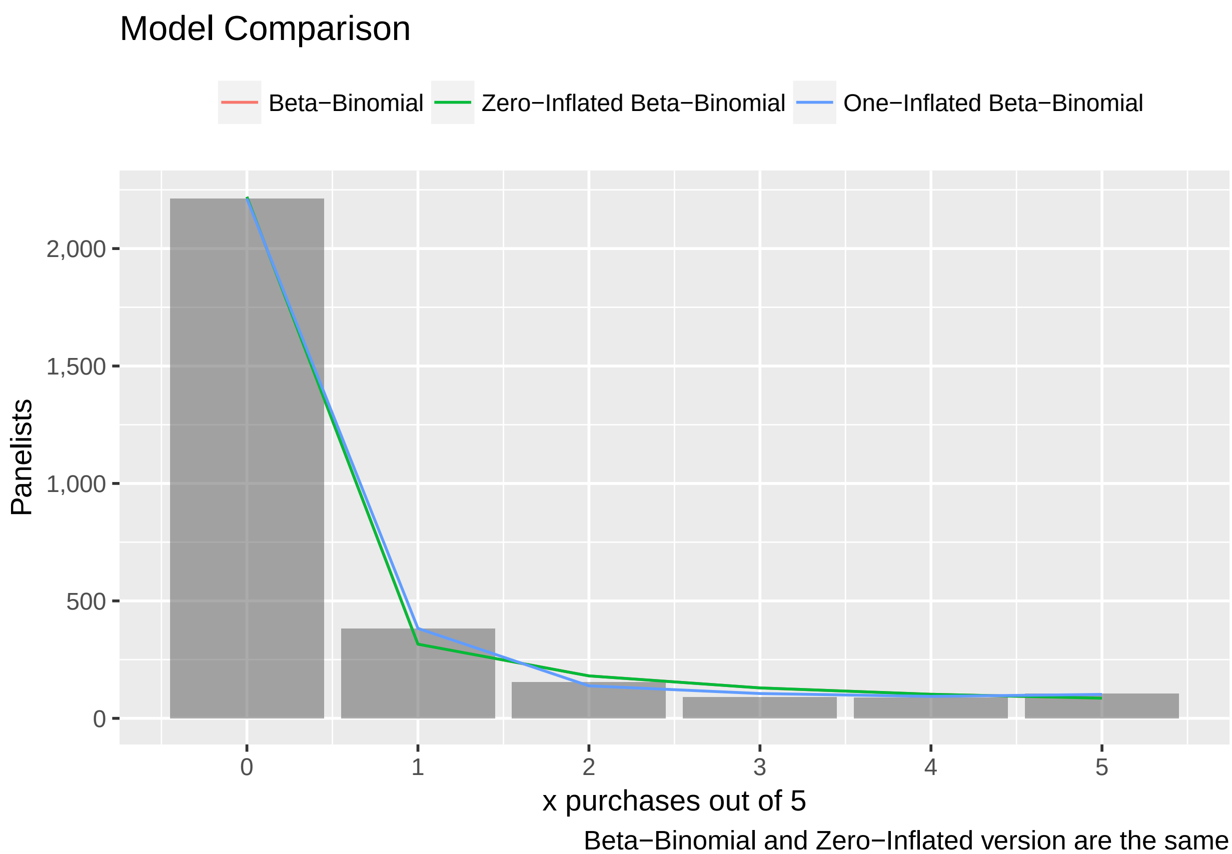
In the table below, we see that the results are quite similar. Noteably, because set a spike at 1 for the 3rd model, the expected number buying 1 out of 5 times matches the actual. Furthermore, the non-buyers for the one-inflated beta-binomial is actually closer to the actual than the regular or the zero-inflated model.

Table 3: Expected number of 3,035 panelist purchasing toothpaste x times out of 5

x	Actual	Beta-Binomial	Zero-Inflated Beta-Binomial	One-Inflated Beta-Binomial
0	2212	2220	2220	2213
1	383	316	316	383
2	154	181	181	138
3	91	130	130	106
4	89	102	102	93
5	106	86	86	102

Below is a graphical display of the results:

```
ggplot() +
  geom_bar(data = bb_expected %>% distinct(x, N_x), aes(x, N_x), stat = 'identity', alpha = 1/2) +
  geom_line(data = bb_expected, aes(x = x, y = expected, colour = model)) +
  theme(legend.position = "top") +
  labs(x = "x purchases out of 5", y = "Panelists", title = "Model Comparison",
       colour = NULL, caption = "Beta-Binomial and Zero-Inflated version are the same") +
  scale_y_continuous(labels = scales::comma) +
  scale_x_continuous(breaks = scales::pretty_breaks())
```



The goodness of fit test shows that beta-binomial and zero-inflated beta-binomial are not good model fits (we reject the null hypotheses that the data comes from either distribution). However, we see that the one-inflated beta-binomial is a good model fit.

```
bb_expected %>%
  group_by(model) %>%
  summarise(chisq = sum(chisq)) %>%
  mutate(p.value = pchisq(chisq, df = 6 - if_else(str_detect(model, "Inflated"), 1L, 0L) - 1, lower.tail = FALSE))
  pander(caption = "Goodness of Fit Test", round = 8)
```

Table 4: Goodness of Fit Test

model	chisq	p.value
Beta-Binomial	36.305	0.00000083
Zero-Inflated Beta-Binomial	36.305	0.00000025
One-Inflated Beta-Binomial	4.172	0.38320752

Using the likehood ratio test we check to see if the larger model (containing $\pi = 1$) is meaningful. We find no reason to believe that the models are the same and thus we select the One-Inflated Beta-Binomial as "best" model of the three.

```
bb_params %>%
  filter(model != "Zero-Inflated Beta-Binomial") %>%
  replace_na(list(pi = 0)) %>%
  bind_cols(data_frame(inflated_at = c(0,1))) %>%
  crossing(toothpaste_data) %>%
  rowwise() %>%
  mutate(p_x = map_dbl(x, .f = fn_bb, m = 5, alpha, beta, pi, inflated_at = 0)) %>%
```

```
  group_by(model) %>%
  summarise(ll = sum(N_x * log(p_x))) %>%
  spread(model, ll) %>%
  mutate(lrt_stat = 2 * (abs(`One-Inflated Beta-Binomial`) - abs(`Beta-Binomial`))) %>%
  mutate(p.value = pchisq(lrt_stat, df = 1, lower.tail = FALSE)) %>%
  pander(caption = "Likelihood Ratio Test")
```

Table 5: Likelihood Ratio Test

Beta-Binomial	One-Inflated Beta-Binomial	lrt_stat	p.value
-2959	-2986	54.2	0

1.2 Implied Penetration

Using the One-Inflated Beta-Binomial model, we find that the implied penetration of the focal brand if the maximum number of purchases were actually 10 is 0.3183 (or 31.83%).

```
bb_params %>%
  filter(model == "One-Inflated Beta-Binomial") %>%
  crossing(x = 0:10) %>%
  rowwise() %>%
  mutate(
    expected = map_dbl(x, .f = fn_bb, m = 10, alpha, beta, pi, inflated_at = 1) * 3035
  ) %>%
  arrange(desc(x)) %>%
  mutate(penetration = cumsum(expected) / sum(expected)) %>%
  mutate(penetration = if_else(x == 0, as.double(NA), penetration)) %>%
  arrange(x) %>%
  select(x, expected, penetration) %>%
  pander(caption = "Implied Penetration", missing = "")
```

Table 6: Implied Penetration

x	expected	penetration
0	2068.89	
1	362.09	0.3183
2	121.64	0.1990
3	87.92	0.1589
4	70.67	0.1300
5	60.46	0.1067
6	54.08	0.0868
7	50.25	0.0689
8	48.62	0.0524
9	49.95	0.0364
10	60.43	0.0199

1.3 Means and Zeros

To implement the “means and zeros” method of the regular beta-binomial method we use the facts that we can compute the actual expectation $E[X]$ and the $P(X = 0)$.

```
actual_expectation <-
  toothpaste_data %>%
  summarise(sum(x *N_x) / sum(N_x)) %>%
```

```
  unlist() %>%
  unname()

actual_p0 <-
  toothpaste_data %>%
  summarise(sum(if_else(x == 0, N_x, as.double(0))) / sum(N_x)) %>%
  unlist() %>%
  unname()
```

Then we use the fact that we use the formula for expectation to solve for beta in terms of alpha

$$E[X] = m\frac{\alpha}{\alpha+\beta} \tag{1}$$

$$\frac{\alpha}{\alpha+\beta} = \frac{E[X]}{m} \tag{2}$$

$$\alpha = \frac{E[X]}{m}(\alpha+\beta) \tag{3}$$

$$\alpha = \frac{E[X]}{m}\alpha + \frac{E[X]}{m}\beta \tag{4}$$

$$\alpha - \frac{E[X]}{m}\alpha = \frac{E[X]}{m}\beta \tag{5}$$

$$\beta = \frac{m}{E[X]}(\alpha - \frac{E[X]}{m}\alpha) \tag{6}$$

$$\beta = \frac{m}{E[X]}\alpha - \alpha \tag{7}$$

$$\beta = \frac{m}{0.6096}\alpha - \alpha \tag{8}$$

We can then minimize the squared error for $P(X = 0)$ using

$$P(X=0) = \binom{m}{0}\frac{\beta(\alpha+0, \frac{m}{E[X]}\alpha - \alpha + m - 0)}{\beta(\alpha, \frac{m}{E[X]}\alpha - \alpha)} \tag{9}$$

```
fn_means_and_zeros <- function(par, m, actual_p0, actual_expectation) {
  alpha <- par[1]
  beta <- (m / actual_expectation  * alpha - alpha)
  f = (fn_bb(0, m, alpha, beta, pi = 0) - actual_p0)^2
  return(f)
}

alpha_maz <- nlminb(c(1), fn_means_and_zeros, lower = c(0), upper = c(Inf), m = 5,
                    actual_p0 = actual_p0, actual_expectation = actual_expectation)$par[1]
beta_maz <- (5 / actual_expectation  * alpha_maz - alpha_maz)
```

Below are the parameters fro the beta-binomial with this dataset using the MLE and means and zeros methods. We see a reasonable difference between the two methods.

Table 7: Comparison of Parameters Based on Estimation Methods

method	alpha	beta
MLE	0.1419	0.9881
Means and Zeros	0.1554	1.1192

2 Question 2

2.1 Posterior Distribution

To derive the posterior distribution of λ for an NBD model for a artbitary period of length t we start in a similar to fashion to a unit time period:

$$
\begin{aligned}
g(\lambda|X(t)=t^*) &= \frac{Poisson \times Gamma}{NBD} && (10)\\
&= \frac{\frac{(\lambda)^x e^{-\lambda t}}{x!} \frac{\alpha^r \lambda^{r-1} e{-\alpha\lambda}}{\Gamma(r)}}{\frac{\Gamma(r+x)}{\Gamma(r)x!}\left(\frac{\alpha}{\alpha+t}\right)^r \left(\frac{t}{\alpha+t}\right)^x} && (11)\\
&= \frac{\lambda^{r+x-1} e^{-\lambda(\alpha+t)}(\alpha+t)^{r+x}}{\Gamma(r+x)} && (12)\\
&= gamma(r+x, \alpha+t) && (13)
\end{aligned}
$$

2.2 Conditional Expectation

We are looking to find the conditional expectation for an NBD for a future period of length t^* applied to a customer who made x purchases over a calibration period of length t. We start with the distribution of $X_2(t^*)$, conditional on $X_1(t) = x_1$, that is

$$
P(X_2(t^*)|X_1(t)=x) = \frac{\Gamma(r+x_1+x_2)}{\Gamma(r)(x_1+x_2)!}\left(\frac{\alpha}{\alpha+t+t^*}\right)^r \left(\frac{t+t^*}{\alpha+t+t^*}\right)^x \tag{14}
$$

Then, the expected value of X_2, conditioned on the fact that $X_1 = x$ (i.e., the conditional expectation of X_2) is

$$
E[X_2(t^*)|X_1(t)=x] = \frac{r+x}{\alpha+t} \tag{15}
$$

3 Question 3

To calculate the posterior estimates of λ, we can use the formula

$$
E[\lambda|X(t)=t] = \frac{r+x}{\alpha+t} \tag{16}
$$

```
billboard_r <- 0.969
billboard_alpha <- 0.218

fn_posterior_lambda <- function(x, r, alpha, t) {
  return((r + x) / (alpha + t))
}

billboard <-
  data_frame(
    customer_name = c(rep("Johari", 3), rep("Fangyuan",3))
    , week = c(1,2,3, 1,2,3)
    , count = c(1,1,1,3,0,0)
    , cumulative_count = c(1,2,3, 3, 3, 3)
```

```
  ) %>%
  rowwise() %>%
  mutate(estimated_lambda = fn_posterior_lambda(cumulative_count, billboard_r, billboard_alpha, t = week))
```

Below are the posterior estimates of lambda.

```
billboard %>%
  pander(caption = "Posterior Estimates of Lambda")
```

Table 8: Posterior Estimates of Lambda

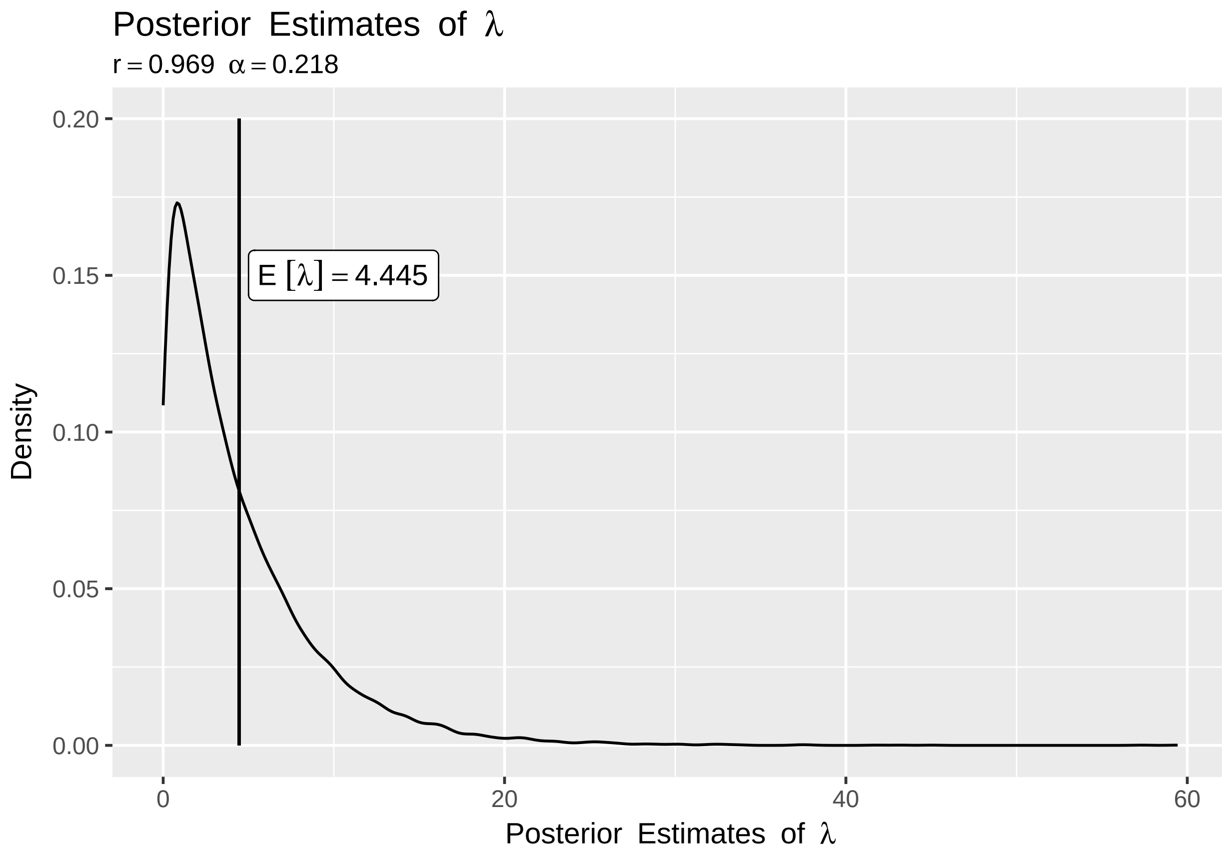
customer_name	week	count	cumulative_count	estimated_lambda
Johari	1	1	1	1.617
Johari	2	1	2	1.339
Johari	3	1	3	1.233
Fangyuan	1	3	3	3.259
Fangyuan	2	0	3	1.789
Fangyuan	3	0	3	1.233

The final estimates make sense. As t increases, we put more weight on what we observed at the individaul-level, rather than population level. If you look at the expanded version of (16) as

$$\begin{aligned} E[\lambda|X(t)=t] &= \frac{r+x}{\alpha+t} && (17) \\ &= \frac{\alpha}{\alpha+1}\frac{r}{\alpha} + \frac{1}{\alpha+t}x && (18) \end{aligned}$$

we see that t gets bigger, x is the primary driver of the posterior estimate rather than the population mean $\frac{r}{\alpha} = 4.445$. The actual distribution gamma distribution of the posterior estimates are shown below:

```
data_frame(estimate = rgamma(10000, billboard_r, billboard_alpha)) %>%
  ggplot(aes(estimate)) +
  geom_line(stat = 'density') +
  labs(x = expression(Posterior~~Estimates~~of~~lambda), y = "Density",
       title = expression(Posterior~~Estimates~~of~~lambda),
       subtitle = expression(r == 0.969~~alpha == 0.218)) +
  geom_segment(aes(x = (billboard_r / billboard_alpha), xend = (billboard_r / billboard_alpha), y = 0, yend = .2))
  geom_label(data = data_frame(x = (billboard_r / billboard_alpha), y = .15, label = paste0('E~group("[",lambda,"]"
             aes(x, y, label = label), hjust = -.05, parse = TRUE)
```



Posterior Estimates of λ
r = 0.969 α = 0.218
E [λ] = 4.445
Density
0.20
0.15
0.10
0.05
0.00
0
20
40
60
Posterior Estimates of λ

4 Question 4