MKT 500T

Spring 2018 HW4

Sami Cheong

Feb 19 2018

Q1 Beta-Binomial model

The beta-binomial model is defined as

$$p(X = x|n) = \binom{n}{x} \frac{B(\alpha + x, \beta + n - x)}{B(\alpha, \beta)},$$

which can be expressed as

$$p(X = x|n) = \binom{n}{x} \frac{\Gamma(\alpha + x)\Gamma(\beta + n - x)}{\Gamma(\alpha + \beta + n)} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)}$$

Data overview:

- Total population = 3035, n (choices) = 0, 1,2,3,4,5

```r
library(knitr)
library(ggplot2)
library(dplyr)
library(readxl)

data<-read_excel('data/HW4 data.xlsx',sheet = 1)
n<-sum(data$N_x)
kable(data,align = 'c')
```

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

First, let's define the beta-binomial function and LL function:

```r
LL.count<-function(count.data,pmf,par,debug=F){

  # par[1] = r, # par[2] = alpha
  pmf.val<-pmf(count.data,par)

  LL.val<-sum(count.data$N_x*log10(pmf.val))
  if(debug){
    plot(count.data$x,pmf.val,type = 'l')
    print(pmf)
    print(log10(pmf.val))
    print(LL.val)
  }

  return(LL.val)
}


BB.pmf<-function(count.data,par){
  N<-max(count.data$x)
  N_count<-nrow(count.data)
  #print(N)
  alpha<-par[1]
  beta<-par[2]

  pmf.val = vector("numeric",N_count)
  for (i in 1:N_count){
    x<-count.data$x[i]
    #print(x)
    comb<-choose(N,x)
    pmf.val[i]<-(comb*beta(alpha+x,beta+N-x))/beta(alpha,beta)
    #print(pmf.val)
  }
  return(pmf.val)



}
## BB with spike at 0
BBZ.pmf<-function(count.data,par){
  N<-max(count.data$x)
  N_count<-nrow(count.data)
  alpha<-par[1]
  beta<-par[2]
  p0<-par[3]/(par[3]+1)
  pmfz.val = vector("numeric",N_count)
  pmf.val<-BB.pmf(count.data,c(par[1],par[2]))
  pmfz.val[1]<-p0*pmf.val[1]
  pmfz.val[2:N_count]<-(1-p0)*pmf.val[2:N_count]
  return(pmfz.val)

}



BB1.pmf<-function(count.data,par,N){
  N<-max(count.data$x)
  N_count<-nrow(count.data)
  alpha<-par[1]
  beta<-par[2]
  p0<-par[3]/(par[3]+1)
  pmfz.val = vector("numeric",N_count)
  pmf.val<-BB.pmf(count.data,c(par[1],par[2]))
  pmfz.val[1:(N_count-1)]<-(1-p0)*pmf.val[1:(N_count-1)]
  pmfz.val[N_count]<-p0+(1-p0)*pmf.val[N_count]
  return(pmfz.val)

}
```

1a Using MLE to solve for the optimal parameters, we have the following results:

```
##          method   chi2 df p_val      MAPE
## 1 beta-binomial 36.305  3     0 0.1864792
```

```
##       alpha      beta p0        LL
## 1 0.1418907 0.9881126 NA -1284.963
```

x	N_x	beta-binomial	exp_beta-binomial	chi_ beta-binomial	ape_ beta-binomial
0	2212	0.7316026	2220.41382	0.0318825	0.0038037
1	383	0.1040550	315.80685	14.2964595	0.1754390
2	154	0.0595868	180.84590	3.9851726	0.1743240
3	91	0.0427120	129.63104	11.5123446	0.4245169
4	89	0.0337497	102.43046	1.7609718	0.1509040
5	106	0.0282939	85.87195	4.7179381	0.1898873

```
##            method     chi2 df p_val      MAPE
## 1 beta-binomial-0 1991.843  2     0 0.6346591
```

```
##       alpha      beta        p0        LL
## 1 0.1418843 0.9879959 0.7287654 -2055.276
```

x	N_x	beta-binomial-0	exp_beta-binomial-0	chi_ beta-binomial-0	ape_ beta-binomial-0
0	2212	0.5331620	1618.14671	217.94176	0.2684689
1	383	0.0282225	85.65515	1032.20828	0.7763573
2	154	0.0161619	49.05134	224.54476	0.6814848
3	91	0.0115853	35.16146	88.67501	0.6136103
4	89	0.0091549	27.78508	134.86618	0.6878081
5	106	0.0076758	23.29616	293.60736	0.7802249

```
##            method   chi2 df p_val      MAPE
## 1 beta-binomial-1 16.141  2 3e-04 0.1176386
```

```
##       alpha     beta         p0        LL
## 1 0.1731491 1.507154 0.02196184 -1280.512
```

x	N_x	beta-binomial-1	exp_beta-binomial-1	chi_ beta-binomial-1	ape_ beta-binomial-1
0	2212	0.7300931	2215.8325	0.0066286	0.0017326
1	383	0.1147734	348.3374	3.4492323	0.0905029
2	154	0.0597478	181.3347	4.1204841	0.1774982
3	91	0.0370218	112.3610	4.0609597	0.2347366
4	89	0.0234281	71.1042	4.5040879	0.2010764
5	106	0.0349358	106.0302	0.0000086	0.0002848

Notice that, the p-value from the chi-square statistics is close to 0 for all 3 models, but the beta-binomial model with spike at 0 is the worst (as indicated by MAPE), while moving the spike to x = 5 gives the lowest MAPE and relatively lower chi-squared value. In this case, we would consider that to be the best model, let's denote this as *BB1.*

1b. Determine the implied penetration if $n = 10$ instead of 5.

If the max category is actually 10, that means we need to adjust the formula a bit. Instead of using $n = 5$, we should replace that to $n = 10$ and retrain the model:

```
##              method   chi2 df p_val      MAPE
## 1 beta-binomial-1N 47.009  2     0 0.1194858
```

```
##       alpha    beta         p0       LL
## 1 0.2539063 5.02538 0.02743721 -1293.11
```

x	N_x	beta-binomial-1N	exp_beta-binomial-1N	chi_ beta-binomial-1N	ape_ beta-binomial-1N
0	2212	0.7270921	2206.72462	0.0126113	0.0023849
1	383	0.1316280	399.49102	0.6807507	0.0430575
2	154	0.0570211	173.05901	2.0989720	0.1237598
3	91	0.0284998	86.49679	0.2344465	0.0494858
4	89	0.0147194	44.67345	43.9823314	0.4980511
5	106	0.0349320	106.01861	0.0000033	0.0001755

To estimated expected penetration given n =10, we use the formula $P(X=0)=\frac{B(\alpha,\beta+n)}{B(\alpha,\beta)}$. Plugging in the values trained from the given data ($\alpha = 0.2539, \beta = 5.0254$), the penetration is $1-P(x=0)=0.252$

1c Using the 'means and zeros' estimation scheme to estiamte α and β for the beta-binomial distribution:

This means we need to solve for α^* and β^* such that

$$P(X=0|n)=\frac{B(\alpha^*,\beta^*+n)}{B(\alpha^*,\beta^*)}$$

and

$$\mu = n\frac{\alpha^*}{\alpha^*+\beta^*},$$

so we have a system of two non-linear equations

```r
require(nleqslv)
BB_zm<-function(par,data){
  n=10
  p_x<-data$N_x/(sum(data$N_x))
  p0<-p_x[1]
  #print(p0)
  mu <- sum(p_x*data$x)/n
  #print(mu)
  alpha <- par[1]
  beta <-par[2]

  y<-numeric(2)
  y[1]<-p0-(beta(alpha,beta+n)/beta(alpha,beta))
  y[2]<-mu-(n*alpha/(alpha+beta))

  return(y)
}
# solve for alpha and r
par_BB_solv<-nleqslv(c(1,1),data=data,
                 BB_zm,
                 control = list(ftol = 1e-100, allowSingular =FALSE),
                 method = 'Newton',jacobian = FALSE)
par_BB_zm<-par_BB_solv$x
par_zm<-list()
```

```r
par_zm$par<-c(par_BB_zm[1],par_BB_zm[2])
par_zm$value<-NA

result_bbZM<-BB.pmf(count.data = data,par=par_zm$par)
print(par_zm)
```

```
## $par
## [1]  143.4011 8620.5980
##
## $value
## [1] NA
```

It looks like there the solver provided a local minimum, applying those values to the dataset resulted in a really poor fit.

Q2 Derive the posterior distribution for λ and the conditional expectatino formula for an NBD model

The posterior distribution of λ is

$$g(\lambda|x) \propto \frac{g(x)p(x|\lambda)}{\int g(x)p(x|\lambda)d\lambda},$$

where

$$g(x) \sim \Gamma(r,\alpha)$$
$$p(x|\lambda) \sim Poi(\lambda t)$$
$$\int g(x)p(x|\lambda)d\lambda \sim NBD$$

First, focus on the top part of the equation:

$$g(x)p(x|\lambda) = \frac{\alpha^r\lambda^{r-1}e^{-\alpha\lambda}}{\Gamma(r)}\frac{\lambda^x e^{-\lambda t}}{x!} = \frac{\alpha^r\lambda^{r+x-1}e^{-\lambda(\alpha+t)}}{\Gamma(r)x!}$$

The bottom part of the equation:

$$\int g(x)p(x|\lambda)d\lambda = \frac{\Gamma(r+x)}{x!\Gamma(r)}(\frac{\alpha}{\alpha+t})^r(\frac{1}{\alpha+t})^\lambda$$

Putting the top and bottom together:

$$g(\lambda|x) = \frac{\lambda^{r+x-1}e^{-\lambda(\alpha+t)}(\alpha+t)^{r+x}}{\Gamma(r+x)} \sim \Gamma(r+x,\alpha+t)$$

Conditional Expectation for NBD

$$E(X) = \sum x_i p(x|t=t^*) = \frac{r+x}{\alpha+t^*}$$

Q3 Revisiting the Billboard example

Given $\alpha = 0.218$ and $r = 0.969$, and data from two customers, we can estimate the Bayesian mean $\frac{r+x}{\alpha+t}$ using the cumulative viewing frequency. Notice that for non-unit time NBD, the global mean is $\frac{rt}{\mu}$

```r
bayes.data<-read_excel('data/Bayes_example.xlsx')
bayes.data[,'Freq']<-bayes.data$Johari+bayes.data$Fangyuan
alpha = 0.218
r=0.969

kable(bayes.data)
```

t	Johari	Fangyuan	Freq
1	1	3	4
2	1	0	1
3	1	0	1

```r
bayes.data[,'bayes_est']<- (r+bayes.data$Freq)/(alpha+bayes.data$t)
bayes.data[,'global_bayes_est']<- (r*bayes.data$t)/(alpha)

kable(bayes.data)
```

t	Johari	Fangyuan	Freq	bayes_est	global_bayes_est
1	1	3	4	4.0796388	4.444954
2	1	0	1	0.8877367	8.889908
3	1	0	1	0.6118707	13.334862

```r
print(r/alpha)
```

```
## [1] 4.444954
```

Except for $t = 1$, the Bayesian estimates given by the data are quite different from those given by the global mean. This points to the importance of getting information from the data to provide an updated Bayesian estimate.

Q4. Consider a zero-inflated NBD model with parameters r, α and π, derive the probability that someone who made zero purchase is part of the 'spike at zero' group.

For this question, let's introduce a Bernoulli variable Z, where $p(Z = 1) = \pi$ and $p(Z = 0) = 1 - \pi$. We are trying to find the probability $p(Z = 1|X = 0)$

Using Baye's theorem, we know that: *

$$p(Z = 1|X = 0) = \frac{p(X = 0|Z = 1)p(Z = 1)}{p(X = 0)}$$

- Assuming $X \sim NBD(r, \alpha)$, we can derive p(X=0) to be $(\frac{\alpha}{\alpha+t})^r$.
- $p(X = 0|Z = 1)$ is the probability that someone made a zero purchase given they're from the spike at 0 group, as it was defined previously, that probability is

$$p(X = 0|Z = 1) = \pi + (1 - \pi)p(X = 0)$$

Therefore, putting it together we have a non-standard density:

$$p(Z=1|X=0) = \frac{(\pi + (1-\pi)(\frac{\alpha}{\alpha+t})^r)\pi}{(\frac{\alpha}{\alpha+t})^r}$$

Plugging in the parameter values we estimated from the billboard example, it shows that there is about a 2.2 % chance that someone who made 0 purchase (at time 1) is indeed from the 'zero' group.

```r
r=1.031391516
alpha=0.226778383
pi=0.020231372
px0<-(alpha/(alpha+1))^r

pz1<-((pi +(1-pi)*px0)*pi)/px0
print(pz1)
```

```
## [1] 0.02215676
```

Q5. Questions I would like more clarity on:

1. Can we see more examples of non-unit time models and how to handle estimates that are conditioned on a previous time?
2. For those that are interested in using R for the class, can we go over some of the solvers for different types of optimization problems and non-linear equations?
3. I didn't quite understand the concept of 'shrinkage' for Bayesian estimates, are there a more general source, or mathematical representation we can look into?