MA 677-Final Report

Xiang Li

5/12/2022

Introduction

This report is for MA677 final project. In this report, I will reproduce 4 examples in Chapter 6 in Computer Age Statistical Inference. Chapter 6 contains four examples – insurance claims (Robbins' Formula), species discovery(The Missing-Species Problem), Shakespeare's vocabulary, and lymph node counts (A Medical Example). In each example, the result of the empirical Bayes analysis is given.

Robbins' Formula (insurance claims)

This example creates a table about number of claims of an insurance company. 7840 of the 9461 policy holders made no claims during the year, 1317 made a single claim, 239 made two claims each, etc., with Ta ble 6.1 continuing to the one person who made seven claims.

```
# Create a data frame
auto <- data.frame(Claims_x=seq(0,7),
           Counts_yx=c(7840,1317,239,42,14,4,4,1))
#auto

# Calculate the expectation of the number of claims for a single customer
n <- 8
robbin1<-round(((auto$Claims_x+1)[1:7]*auto$Counts_yx[2:8]/auto$Counts_yx[1:7]),3)

# Calculate the parametric estimated marginal density and
# then get the maximum likelihood fitting to the counts y_x
f <- function(x,mu,sigma){
  gamma = sigma / (1 + sigma)
  numer = gamma ^ (mu + x) * gamma(mu + x)
  denom = sigma ^ mu * gamma(mu) * factorial(x)
  return(numer/denom)
}
neg_like <-function(param){
  mu=param[1]
  sigma=param[2]
  tmp=-sum(auto$Counts*log(f(auto$Claims_x,mu=mu,sigma=sigma)))
  return(tmp)
}
p <- array(c(0.5, 1), dim = c(2, 1))
ans_auto <- nlm(f = neg_like,p,hessian=T)
mu=ans_auto$estimate[1]
sigma=ans_auto$estimate[2]
```

```
re <- round((seq(0,6)+1)*f(seq(0,6)+1,mu,sigma)/f(seq(0,6),mu,sigma),3)
rbind(robbin1,re)
```

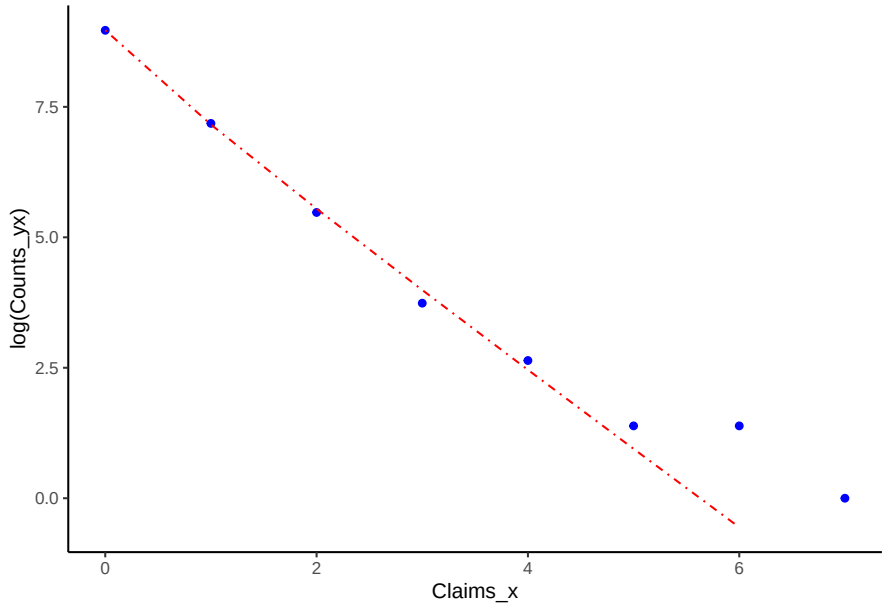
```
##          [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
## robbin1 0.168 0.363 0.527 1.333 1.429 6.000 1.750
## re      0.164 0.398 0.632 0.866 1.100 1.334 1.568
```

```
auto$pred=c(f(seq(0,6),mu,sigma)*9461,NA)

# Visualize the comparison between  log(counts) and claims for 9461 auto insurance policies.
# The dashed line is a gamma MLE fit.

p1 <- ggplot(data=auto) +
  geom_point(aes(x=Claims_x,y=log(Counts_yx)),color='blue')+
  geom_line(aes(x=Claims_x,y=log(pred)),color='red',lty=4)+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), panel.background = eleme

p1
```



The Missing-Species Problem

The very first empirical Bayes success story related to the butterfly data. Even in the midst of World War II Alexander Corbet, a leading naturalist, had been trapping butterflies for two years in Malaysia (then Malaya): 118 species were so rare that he had trapped only one specimen each, 74 species had been trapped twice each.

```
# Create the data set

butterfly <- data.frame(x=seq(1,24),
                    y=c(118,74,44,24,29,22,20,19,20,15,12,14,6,12,6,9,9,6,10,10,11,5,3,3))

# Estimate the expected number of new species seen in the new trapping period E(t)
# with Robbins' formula
Fisher1 <- function(t){
  re <- round(butterfly$y * t^(butterfly$x)* (-1)^(butterfly$x-1),2)
  sd <- round((sum(butterfly$y * (t)^(2)))^(1/2),2)
  return(list('est'=sum(re),'sd'=sd))
}
F1 <- sapply(seq(0,1,0.1),Fisher1)
F1
```

```
##     [,1] [,2] [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]  [,10] [,11]
## est 0    11.1 20.96 29.79 37.81 45.2  52.14 58.94 65.55 71.57 75
## sd  0    2.24 4.48  6.71  8.95  11.19 13.43 15.67 17.91 20.14 22.38
```

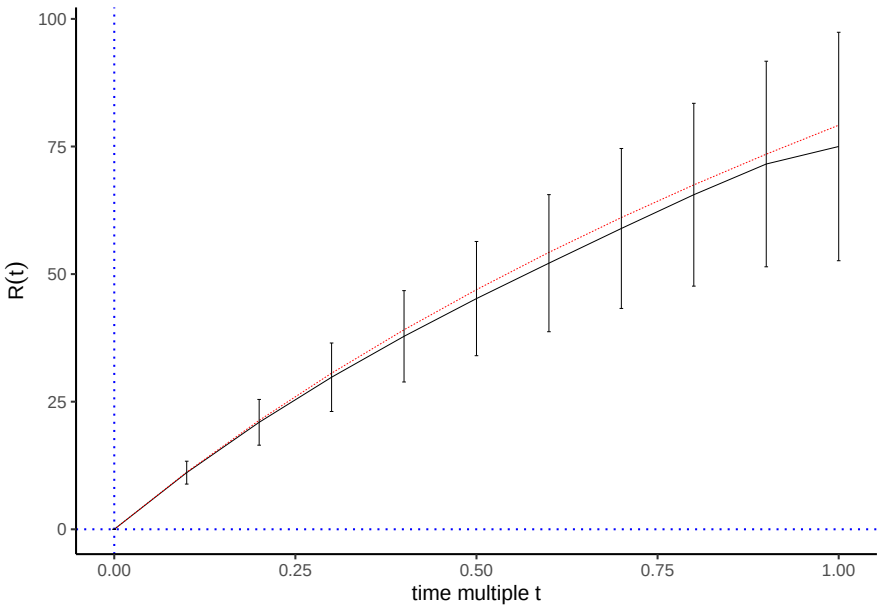
```
# Calculate the parametric estimate of E(t)
v <- 0.104
sigma <-  89.79
gamma <- sigma / (1 + sigma)
e1 <- 118
fisherFn <- function(t){
  re <- e1*((1 - (1+gamma*t)^(-v)) / (gamma * v))
  return(re)
}
EST2 <- sapply(seq(0,1,0.1),fisherFn)
EST2
```

```
##  [1]  0.00000 11.19732 21.33347 30.58504 39.08842 46.95109 54.25922 61.08287
##  [9] 67.47981 73.49817 79.17850
```

```
# Visualize the comparison between nonparametric fit (solid) 1 standard deviation and gamma model

df <- data.frame(time=seq(0,1,0.1),est1=unlist(F1[1,]),sd=unlist(F1[2,]),est2=EST2)
p2 <-  ggplot(data = df) +
  geom_line(mapping = aes(x = time, y = est1), size = 0.25) +
  geom_line(mapping = aes(x = time, y = est2), color = "red", size = 0.1, linetype ="dashed") +
  geom_hline(yintercept = 0.0, color = "blue", linetype="dotted") +
  geom_vline(xintercept = 0.0, color = "blue", linetype="dotted") +
  geom_errorbar(mapping = aes(x = time, ymin = (est1 - sd), ymax = (est1 + sd)), width=0.005, color="bl
  labs(x = "time multiple t", y = expression(R(t)))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.line = element_line(colour = "black"))
```

```
p2
```



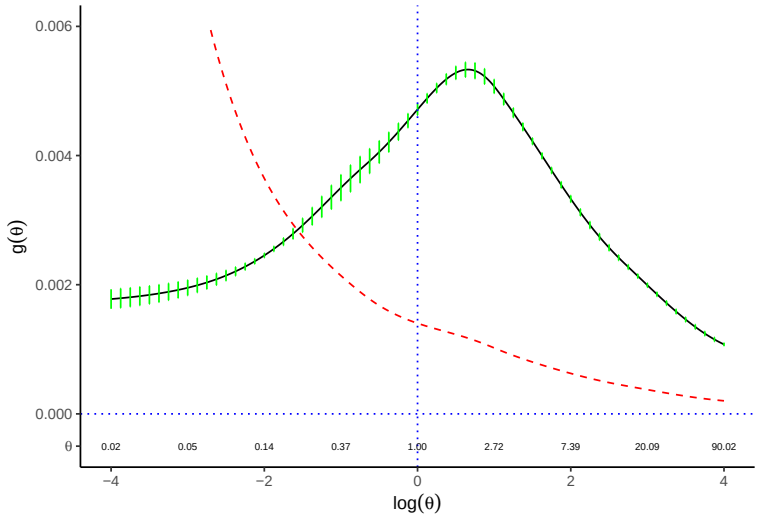
We could see that the expected number of new species in t units of additional trapping time ,nonparametric fit (solid) 1 standard deviation, gamma model which is plotted by dashed line.

Shakespeare's vocabulary

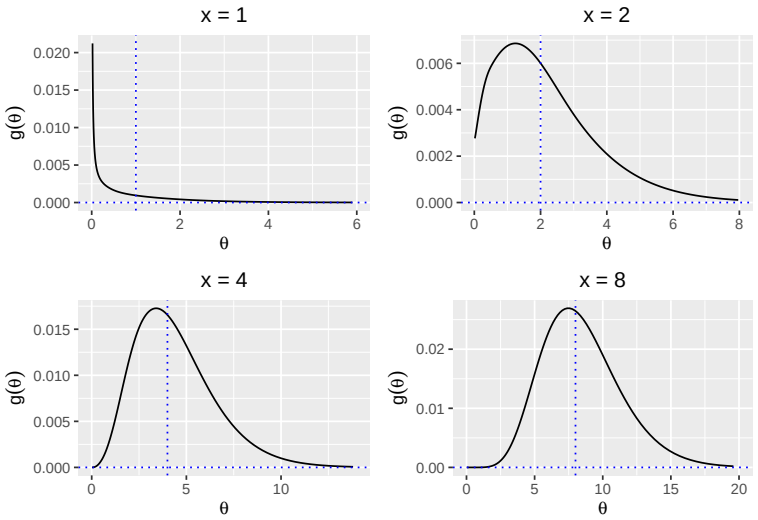
This example gives Shakespeare's word counts. 14,376 distinct words appeared once each in the canon, 4343 distinct words twice each, etc. The canon has 884,647 words in total, counting repeats. 11430+/-178 is for the expected number of distinct new words if t = 1.

```
data("bardWordCount", package = "deconvolveR")
lambda <- seq(-4, 4.5, .025)
tau <- exp(lambda)
result <- deconv(tau = tau, y = bardWordCount, n = 100, c0=2)
stats <- result$stats
d <- data.frame(lambda = lambda, g = stats[, "g"], tg = stats[, "tg"],
                SE.g = stats[, "SE.g"])
indices <- seq(1, length(lambda), 5)

p3 <- ggplot(data = d) +
    geom_line(mapping = aes(x = lambda, y = g)) +
    geom_errorbar(data = d[indices, ],
                  mapping = aes(x = lambda, ymin = g - SE.g, ymax = g + SE.g),
                  width = .01, color = "green") +
    labs(x = expression(log(theta)), y = expression(g(theta))) +
    xlim(-4, 4) +
    geom_vline(xintercept = 0.0, linetype = "dotted", color = "blue") +
    geom_hline(yintercept = 0.0, linetype = "dotted", color = "blue") +
    geom_line(mapping = aes(x = lambda, y = tg),
              linetype = "dashed", color = "red") +
    annotate("text", x = c(-4, -3, -2, -1, 0, 1, 2, 3, 4),
             y = rep(-0.0005, 9),
             label = c("0.02", "0.05", "0.14", "0.37", "1.00", "2.72", "7.39", "20.09", "90.02"), size =
    scale_y_continuous(breaks = c(-0.0005, 0.0, 0.002, 0.004, 0.006),
                       labels = c(expression(theta), "0.000", "0.002","0.004","0.006"),
                       limits = c(-0.0005, 0.006)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.line = element_line(colour = "black"))
p3
```



```
gPost <- sapply(seq_len(100), function(i) local({tg <- d$tg * result$P[i, ]; tg / sum(tg)}))
p3_2 <- lapply(c(1, 2, 4, 8), function(i) {
ggplot() +
geom_line(mapping = aes(x = tau, y = gPost[, i])) +
geom_vline(xintercept = i, linetype = "dotted", color = "blue") +
geom_hline(yintercept = 0.0, linetype = "dotted", color = "blue") +
labs(x = expression(theta), y = expression(g(theta)),
title = sprintf("x = %d", i))
})
p3_2 <- Map(f = function(p, xlim) p + xlim(0, xlim) + theme(plot.title=element_text(hjust=0.5)),
p3_2, list(6, 8, 14, 20))
print(plot_grid(plotlist = p3_2, ncol = 2))
```

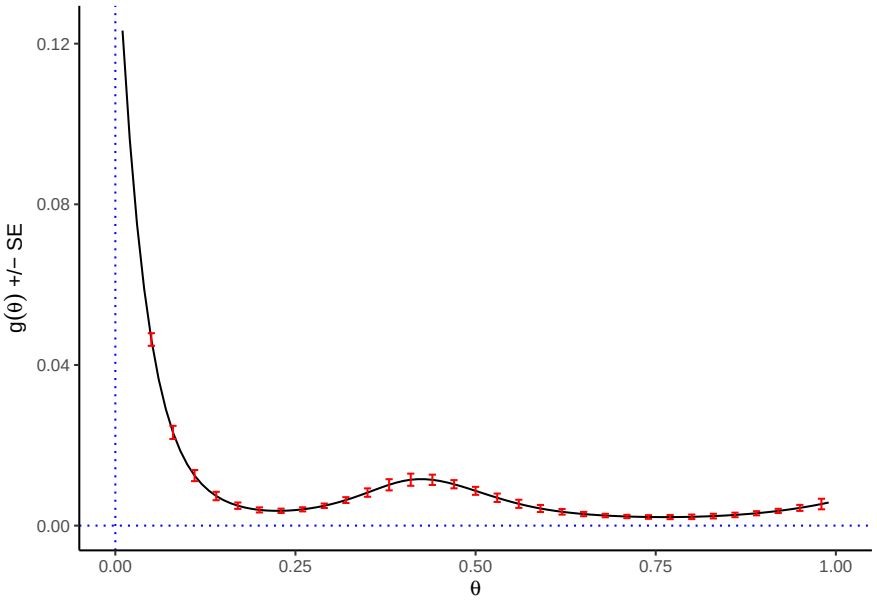


A Medical Example

Cancer surgery sometimes involves the removal of surrounding lymph nodes as well as the primary target at the site. n = # nodes removed and x = # nodes found positive; "positive" meaning malignant.

```
data(surg)
p <- surg$x/surg$n
tau <- seq(from = 0.01, to = 0.99, by = 0.01)
result <- deconv(tau = tau, X = surg, family = "Binomial")
d <- data.frame(result$stats)
indices <- seq(5, 99, 3)
errorX <- tau[indices]
p4 <- ggplot() +
  geom_line(data = d, mapping = aes(x = tau, y = g)) +
  geom_errorbar(data = d[indices, ],
                mapping = aes(x = theta, ymin = g - SE.g, ymax = g + SE.g), width = .01, color = "red")
  geom_vline(xintercept = 0.0, linetype = "dotted", color = "blue") +
    geom_hline(yintercept = 0.0, linetype = "dotted", color = "blue") +
  labs(x = expression(theta), y = expression(paste(g(theta), " +/- SE")))+
  expand_limits(x=c(0,1), y=c(0, 0.12,0.02))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.line = element_line(colour = "black"))

p4
```

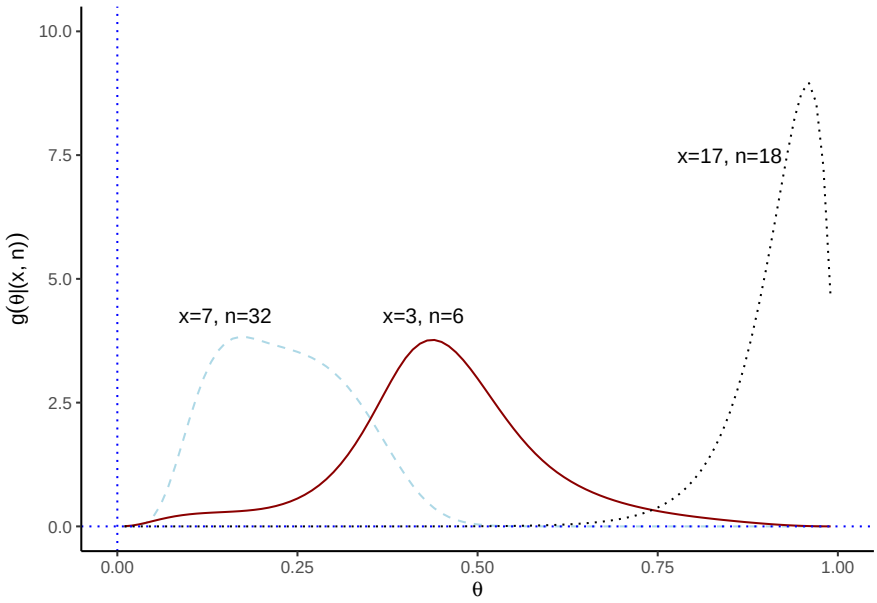



```
theta <- result$stats[, 'theta']
gTheta <- result$stats[, 'g']
f_alpha <- function(n_k, x_k) {
    ## .01 is the delta_theta in the Riemann sum
    sum(dbinom(x = x_k, size = n_k, prob = theta) * gTheta) * .01
}
g_theta_hat <- function(n_k, x_k) {
    gTheta * dbinom(x = x_k, size = n_k, prob = theta) / f_alpha(n_k, x_k)
```

```
}

g1 <- g_theta_hat(x_k = 7, n_k = 32)
g2 <- g_theta_hat(x_k = 3, n_k = 6)
g3 <- g_theta_hat(x_k = 17, n_k = 18)
p4_2 <- ggplot() +
    geom_line(mapping = aes(x = theta, y = g1), col = "lightblue",linetype = "dashed")+
    ylim(0, 10) +
    geom_line(mapping = aes(x = theta, y = g2), col = "darkred",) +
    geom_line(mapping = aes(x = theta, y = g3), col = "black", linetype ="dotted") +
    labs(x = expression(theta), y = expression(g(paste(theta, "|(x, n)")))) +
    geom_vline(xintercept = 0.0, linetype = "dotted", color = "blue") +
    geom_hline(yintercept = 0.0, linetype = "dotted", color = "blue") +
    annotate("text", x = 0.15, y = 4.25, label = "x=7, n=32") +
    annotate("text", x = 0.425, y = 4.25, label = "x=3, n=6") +
    annotate("text", x = 0.85, y = 7.5, label = "x=17, n=18") +
    expand_limits(x=c(0,1,0.2), y=c(0, 6,2))+
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.line = element_line(colour = "black"))

p4_2
```



Discussion

In this final report, I learned the important points of robin's formula and empirical Bayes analysis by reproducing these four examples. What's great about this method is that as long as you have a lot of examples, you don't need to bring in prior expectations. Additionally, I have learned a lot, understood key principles of Empirical Bayes estimation from a post by David Robinson, who is the author of Introduction to Empirical Bayes.

Reference: https://github.com/jrfiedler/CASI_Python/blob/master/chapter06/ch06s01.ipynb

https://github.com/jrfiedler/CASI_Python/blob/master/chapter06/ch06s02.ipynb

Professor Haviland Wright's class note: "File deconvolveR hw.R"

https://github.com/MA615-Yuli

http://varianceexplained.org/r/empirical_bayes_baseball/