STATS245HW2

Lin Lee Cheong

7/27/2017

Problem 1 part a

Which one can control type-I error? Explain why. You can run the codes in the file several times to verify your answer.

The random pick method produces about 0.05 probability of Type I error, while picking zmax method produces ~0.2 probability of Type I error and cannot be controlled. This is because picking zmax is not random and will skew the distribution (which is accounted for by Taylor's method in the next parts)

```
Ffun <- function(a,b,mu,si,x){
  xs <- (x-mu)/si
  as <- (a-mu)/si
  bs <- (b-mu)/si

  f <- function(y) (1/y-1/y^3+3/y^5-15/y^7);

  if (as>4 & bs>4){
    F <- 1-(exp((as^2-bs^2)/2)*f(bs) - exp((as^2-xs^2)/2)*f(xs))/
      ( exp((as^2-bs^2)/2)*f(bs) - f(as) )
  } else if (as< -4 & bs< -4) {
    F <- ( exp((bs^2-xs^2)/2)*f(-xs) - exp((bs^2-as^2)/2)*f(-as))/
      ( f(-bs)-exp((bs^2-as^2)/2)*f(-as) )
  } else {
    denom <- pnorm(bs)-pnorm(as)
    if (denom < 0.00001){
      F <- (xs-as)/(bs-as)
    } else {
      F <- (pnorm(xs)-pnorm(as))/denom
    }
  }
  return(F)
}

numSim <- 500  # number of simulations
Reject_zrand <- rep(0,numSim)
Reject_zmax <- rep(0,numSim)
Reject_taylor <- rep(0,numSim)

for (i in 1:numSim){
  # Generate 5 random variables from N(0,1)
  Zs <- rnorm(5)

  # Reject zrand if zrand>1.645
  Reject_zrand[i] <- (Zs[sample(5,1)]>qnorm(0.95))

  # Reject zmax if zmax>1.645
  jmax <- which(Zs==max(Zs))
  Reject_zmax[i] <- (Zs[jmax]>qnorm(0.95))
```

```
}

print(c(mean(Reject_zrand), mean(Reject_zmax), mean(Reject_taylor)))
```

```
## [1] 0.040 0.236 0.000
```

Problem part b) and c) and d)

Vup, Vlo, Gamma and u are modified below.

```
numSim <- 1000  # number of simulations
Reject_zrand <- rep(0,numSim)
Reject_zmax <- rep(0,numSim)
Reject_taylor <- rep(0,numSim)

for (i in 1:numSim){
  # Generate 5 random variables from N(0,1)
  Zs <- rnorm(5)

  # Reject zrand if zrand>1.645
  Reject_zrand[i] <- (Zs[sample(5,1)]>qnorm(0.95))

  # Reject zmax if zmax>1.645
  jmax <- which(Zs==max(Zs))
  Reject_zmax[i] <- (Zs[jmax]>qnorm(0.95))

  # Taylor method: conditional p-value given that jmax is selected

  # Compute Gamma and u
  # MODIFIED: ====
  # Create 4x5 matrix Gamma (removing jmax column itself)
  Gamma <- matrix(0, nrow = 5, ncol = 5)
  Gamma[, jmax] <- 1
  for(cur_col in seq(1, length(Zs))) {
    if(cur_col != jmax) {
      Gamma[cur_col, cur_col] <- -1
    }
  }
  Gamma <- Gamma[-jmax, ]

  # Create Gamma*Zs = u = 0
  u <-  rep(0, 5)
  u <- u[-jmax]
  # END MODIFIED ====

  # Compute Vup, Vlo
  v <- rep(0,5)
  v[jmax] <- 1
  rho <- (Gamma%*%v)/(sum(v^2))

  if (sum(rho<0)==0){
    Vup <- Inf
  } else {
```

```
    # MODIFIED: ====
    Vup <- min( (u - Gamma %*% Zs + rho %*% t(v) %*% Zs ) / rho )
    # END MODIFIED ====

  }
  if (sum(rho>0)==0){
    Vlo <- -Inf
  } else {

    # MODIFIED: ====
    Vlo <- max( (u - Gamma %*% Zs + rho %*% t(v) %*% Zs ) / rho)
    # END MODIFIED ====

  }
  Reject_taylor[i] <- (Ffun(Vlo, Vup, 0, 1, Zs[jmax])>0.95)
}

print(c(mean(Reject_zrand),mean(Reject_zmax),mean(Reject_taylor)))
```

```
## [1] 0.056 0.220 0.046
```

Problem 3 a

```
GenData <- function(N=200,theta){
  # N: number of samples, by default is 200
  # theta: successful rate

  # Example:
  # data <- GenData(theta=0.5)

  return(rbinom(N, 1, theta))
}


BayesDecision <- function(data,cutoff){
  # data: a length-N vector contains 0 and 1, which corresponds to whther the i-th patient was cured by
  # cutoff: Stopping rule. If the posterior probability of theta>0.5 is over cutoff or below 1-cutoff

  # Example:
  # data <- GenData(theta=0.5)
  # Results <- BayesDecision(data,0.95)
  # NumPatietns <- Results$numPat
  # Decision <- Results$decision
  # Maximum number of patients
  N <- length(data)

  # prior distribution of theta: Beta(1,1)
  a <- 1
  b <- 1

  for (i in 1:N){ # suppose patients are treated one-by-one
    if (data[i] == 1){ # if the i-th is cured by the treatment
```

```
      # MODIFIED: ====
      a <- a + 1
      # END MODIFIED ====
    } else { # if the i-th is NOT cured by the treatment
      # MODIFIED: ====
      b <- b + 1
      # END MODIFIED ====
    }
    # probability of theta>1/2 based on the posterior distribution
    Ptheta <- 1 - pbeta(0.5, a, b)
    if (Ptheta>cutoff | Ptheta<1-cutoff){
      break;
    }
  }
  return(list(numPat=i, decision=(Ptheta>cutoff)))
}
```

Problem 3b

Here I iterated through possible cutoff values and stop by it is ~ <= 0.05

```
mcn <- 3000
for(cur_c in seq(0.95, 1.0, 0.001)){
  mc_decision <- NULL
  mc_numpat <- NULL
  for(cur_mcn in seq(1, mcn)) {
    data <- GenData(theta = 0.5)
    results <- BayesDecision(data, cur_c)
    NumPatients <- results$numPat
    Decision <- results$decision
    mc_decision[cur_mcn] <- Decision
    mc_numpat[cur_mcn] <- NumPatients
  }

  if(mean(mc_decision) < 0.05) {
    print(paste0('Cutoff = ', cur_c, ' when mean(mc_decision) is ', mean(mc_decision)))
    break
  }
}
```

```
## [1] "Cutoff = 0.995 when mean(mc_decision) is 0.0456666666666667"
```

Problem 3c

Cutoff is theta dependent. Need to carefully choose cutoff to control Type 1 error.

```
# simulated cutoff from part I
sim_cutoff <- cur_c
mcn <- 5000
mc_decision <- NULL
mc_numpat <- NULL
mc_mean_prob <- NULL
theta_list <- seq(0.1, 0.9, 0.05)
```

```
for(cur_theta_index in seq(1, length(theta_list))) {
  for(cur_mcn in seq(1, mcn)) {
    data <- GenData(theta = theta_list[cur_theta_index])
    results <- BayesDecision(data, sim_cutoff)
    NumPatients <- results$numPat
    Decision <- results$decision
    mc_decision[cur_mcn] <- Decision
    mc_numpat[cur_mcn] <- NumPatients
  }
  mc_mean_prob[cur_theta_index] <- mean(mc_decision)
}

plot(theta_list, mc_mean_prob)
```