dalGen: borrow both arms

Yi Li

08/12/2022

```
setwd("/Users/yili/Desktop/EPIB 635/Final Project")
library(VGAM)
```

```
## Warning: package ’VGAM’ was built under R version 4.1.2
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

Modify the code a bit to borrow treatment arm as well.

```
# posterior weight: corresponds to w_R in Eq. (7) in Schmidli et al. (2014)
w_bar = function(w, y, n, a, b, a0 = 1, b0 = 1) {
  f = beta(a + y, b + n - y)/beta(a, b)
  f0 = beta(a0 + y, b0 + n - y)/beta(a0, b0)
  w_bar = w*f/(w*f + (1 - w)*f0)
  return(w_bar)
}
```

```
options(warn = -1)
trial2 = function(TE, w,
                  ni = c(132, 264),
                  # keep the same total no. of cases at interim and final analyses
                  a0 = 1, b0 = 1) {
  n = c(ni[1], diff(ni, 1))
  theta = (1 - TE)/(2 - TE)
  p_eff = NULL
  p_success = NULL
  y_i = 0
  i = 0

  repeat {
    i = i + 1
    y_i = y_i + rbinom(1, n[i], theta) # y_i ~ Binom(n_i, theta)
    ai = 1 + y_i
    bi = 1 + ni[i] - y_i

    ### Add robust MAP prior
    # posterior weight:
    wb = w_bar(w, y_i, n[i], ai, bi)
    # P(TE > 0|data):
    # In power_and_mixture_priors.html: y1 is # of cases in historical trial (ctrl arm ) , and y is # o

    peff_new = wb*pbeta(0.5,
                        ai + 226 + 180, # add 180 cases in trx to y1 in MAP code
                        bi + (226 + 180) - 226) +
                    (1-wb)*pbeta(0.5,
                                 ai,
                                 bi)
    p_eff = c(p_eff, peff_new)

    p_success = c(p_success,
                   wb*pbetabinom.ab(q = 117,
                                    size = 264,
                                    shape1 = ai + 226 + 180,
                                    # add 180 cases in trx to y1
                                    shape2 = bi + (226 + 180) - 226,
                                    log = FALSE) +
                     (1-wb)*pbetabinom.ab(q = 117,
                                          size = 264,
                                          shape1 = ai,
                                          shape2 = bi,
                                          log = FALSE)
                   )



    if (i <= 1) {
      # P(TE > 0|data) > 0.995;
      if (peff_new > 0.995 | p_success[length(p_success)] < 0.05) break
    } else break
  }
  out = list(round(p_eff, 2), round(p_success, 8))
  return(out)
}

trial2(TE = 0.7, w = 0.7)
```

```
## [[1]]
## [1] 0.06 1.00
##
## [[2]]
## [1] 0.05668654 0.99995963
```

```
trial2(TE = 0.7, w = 0.5) # decrease w because trx arm in historical trial may not be reliable
```

```
## [[1]]
## [1] 0.12 1.00
##
## [[2]]
## [1] 0.1190636 0.9999741
```

3.1 FPR

FPR looks fine if we keep w = 0.7

```
options(warn = -1)
fp = NULL
p_eff <- p_eff_unlist <- check <- list()
for (m in 1:10000) {
  p_eff = trial2(TE = 0, w = 0.7)
  p_eff_unlist = unlist(p_eff[[1]])
  check = ifelse(length(p_eff_unlist) < 2,
                 p_eff_unlist[length(p_eff_unlist)] > 0.995,
                 p_eff_unlist[length(p_eff_unlist)] > 0.986)
  fp = c(fp, check)
}
head(fp)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE
```

```
mean(fp, na.rm = T)
```

```
## [1] 0.0028
```

```
sum(is.na(fp))
```

```
## [1] 0
```

3.2 TPR

w = 0.7: power is not high enough

```
options(warn = -1)
tp = NULL
for (m in 1:10000) {
  p_eff = trial2(TE = 0.7, w = 0.7)
  p_eff_unlist = unlist(p_eff[[1]])
  check = ifelse(length(p_eff_unlist) < 2,
                 p_eff_unlist[length(p_eff_unlist)] > 0.995,
                 p_eff_unlist[length(p_eff_unlist)] > 0.986)
  tp = c(tp, check)
}
mean(tp, na.rm = T)
```

```
## [1] 0.7931
```

```
sum(is.na(tp))
```

```
## [1] 0
```

w = 0.5: power is 1

```
options(warn = -1)
tp = NULL
for (m in 1:10000) {
  p_eff = trial2(TE = 0.7, w = 0.5)
  p_eff_unlist = unlist(p_eff[[1]])
  check = ifelse(length(p_eff_unlist) < 2,
                 p_eff_unlist[length(p_eff_unlist)] > 0.995,
                 p_eff_unlist[length(p_eff_unlist)] > 0.986)
  tp = c(tp, check)
}
mean(tp, na.rm = T)
```

```
## [1] 0.9976
```

```
sum(is.na(tp))
```

```
## [1] 0
```