Simulation

2019-03-17

Highlight procedures of simulation

- Step 1: Specify true α, where $||\alpha|| = 1$. We could set $\alpha = (sin(\theta_0), cos(\theta_0))$. θ_0 is the true θ to make α equal to the true α
- Step 2: Specify β, Γ, D for drug group (drg) and placebo group (pbo)
- Step 3: Simulate n observations (e.g. n = 100/treatment)
- Step 4: Generate 10000 points from the true model.
- Step 5: Compute "true" purity by

$$p(w_i) = \frac{1}{10000} \sum_{i=1}^{10000} \frac{[f_1(z_i|w_i) - f_2(z_i|w_i)]^2}{f_1(z_i|w_i) + f_2(z_i|w_i)}$$

- Step 6: Compute purity using $\hat{f}_1$, $\hat{f}_2$ from the 100 sample.
- Step 7: Specify $\theta \sim [0, \pi]$, e.g. seq(0,π,100). Calculate αs from θ
- Step 8: Fit model using αs
- Step 9: Compute purity

More details of the simulation

Step 1: Specify true α

We could choose the true θ as $\frac{\pi}{3}$, then $\alpha_{true} = (sin(\frac{\pi}{3}), cos(\frac{\pi}{3})) = (\frac{\sqrt{(}3)}{2}, \frac{1}{2}) \approx (0.866, 0.5)$.

Step 2: Specify β, Γ, D for drug group (drg) and placebo group (pbo)

The outcome follows the formula:

$$\text{the outcome is } Y : \mathbf{Y} = \mathbf{X}(\beta + \mathbf{b} + \mathbf{\Gamma}(\alpha' \mathbf{x})) + \epsilon.$$

where

- The baselines come from the same distributions $\mathbf{x} = [x_1, x_2]$, the baseline covariate x_1, x_2, iid $\sim N(0, 1)$
- A combination of baseline covariate w: $w = \alpha^T [x_1, x_2]$, which is $\alpha_1 * x_1 + \alpha_2 * x2$
- X: the independent covariates, $X = [1, t, t^2]$, where $t = 0, 1, 2, ..., 6$
- $\epsilon \sim N(0, 1)$ for drug, $\epsilon \sim N(0, 1)$ for placebo

The β:

- $\beta_{drg} = \begin{bmatrix} 0 \\ -5 \\ -1 \end{bmatrix}$, $\beta_{pbo} = \begin{bmatrix} 0 \\ 5 \\ 1 \end{bmatrix}$

The Γ: * $\Gamma_{drg} = \begin{bmatrix} 0 \\ -2 \\ -1 \end{bmatrix}$, $\Gamma_{pbo} = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}$

The random effect covariance matrix D:

- $D_{drg} = \begin{bmatrix} 0.5461134 & 0.3479693 & 0.4099377 \\ 0.3479693 & 2.1889306 & 0.6409533 \\ 0.4099377 & 0.6409533 & 0.6297127 \end{bmatrix}$

The values are generated from a uniform distribution UNI[0,1], and then make it to be a positive defined matrix. The codes are below:

```
set.seed(123)
Aa = matrix(runif(3^2)*2-1, ncol=3)
bi_sigma = t(Aa) %*% Aa
bi_sigma
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.5461134 0.3479693 0.4099377
## [2,] 0.3479693 2.1889306 0.6409533
## [3,] 0.4099377 0.6409533 0.6297127
```

Step 3: Simulate n observations (e.g. n = 100/treatment)

For drug group, the baseline covariates are generated from N(0,1):

```
p = 2
baseline = as.matrix(rnorm(p,0,1),p,1)
x1 = baseline[1]; x2 = baseline[2]
```

Then w is calculated as $\alpha' x$

```
alpha = c(sin(pi/3),cos(pi/3))
w = t(alpha) %*% baseline
```

The random effect is generated from MVN, whose mean = 0 and sigma equals to the covariance matrix D

```
bi = mvrnorm(1, c(0,0,0), bi_sigma)
```

Then the outcome is calculated

```
yi = X%*%(beta_drg+bi+gamma_drg*w[1]) + sigma_drg*rnorm(ni,0,1)
```

The placebo group follows the same generation process.

p.s. a larger n, e.g. n = 10000 can make very close estimations but it also runs very slow.

Step 4, 5: Monte carlo simulation

Since we would like to calculate the integral of

$$\int \frac{[f_1(z_i|w_i) - f_2(z_i|w_i)]^2}{f_1(z_i|w_i) + f_2(z_i|w_i)} dz_i$$

To make it run faster, we could just generated 10000 points from the x-axis and calculate the summation of

$$\sum \frac{[\hat{f}_1(z_i|w_i) - \hat{f}_2(z_i|w_i)]^2}{hat f_1(z_i|w_i) + \hat{f}_2(z_i|w_i)}$$

where: $\hat{f}_1$ and $\hat{f}_2$ are multivariate normal distribution, whose means and covariance matrix are estimated by fitting the linear mixed model. $(\mu = \hat{\beta} + \hat{\Gamma} * w)$

The codes are showing below:

```
Xstart = cbind(runif(2000,-1,1),runif(2000,-1,1))

monta_carlo_pdf = function(Xstart, mu1, D1, mu2, D2){
  n_mc = dim(Xstart)[1]
  mu1 = matrix(rep(mu1, n_mc),n_mc,2,byrow = TRUE)
  mu2 = matrix(rep(mu2, n_mc),n_mc,2,byrow = TRUE)

  Q1 = diag((-1/2)*(Xstart-mu1)%*%solve(D1)%*%t(Xstart-mu1))
  Q2 = diag((-1/2)*(Xstart-mu2)%*%solve(D2)%*%t(Xstart-mu2))

  f1 = (1/(2*pi))*(1/sqrt(det(D1)))*exp(Q1)
  f2 = (1/(2*pi))*(1/sqrt(det(D2)))*exp(Q2)

  f3 = (f1 + f2) # deal with the 0s, f1 + f2 cannot be 0
  f3 = ifelse(f3 == 0, 1e-4,f3)

  purity = c((f1 - f2)^2 / f3)
  return(purity)
}
```

Step 6: Compute purity using $\hat{f}_1$, $\hat{f}_2$ from the 100 sample.

- 1: Fit linear mixed model, calculated the estimated β and estimated Γ, estimated covariance matirx D.
- 2: The esimated mu for fitting the multivariate normal distribution is $\mu = \hat{\beta} + \hat{\Gamma} * w$.
- 3: Then calculate the multivariate normal distriubtions
- 4: Calculate the purity. (Monte Carlo simulation)

```
purity_function = function(A, data = dat){
  alpha_est = matrix(A,2,1)
  dat_est = dat
  w_est = cbind(dat$x1,dat$x2) %*% alpha_est
  dat_est$w = w_est
  dat_pbo_est = dat_est[dat_est$trt == 'pbo', ]
  dat_drg_est = dat_est[dat_est$trt == 'drg', ]

  # fit the LMM
  fit_drg_est = lmer(y ~ tt + I(tt^2) + w + w * tt +
                       w * I(tt^2) + (tt+I(tt^2)|subj),
                     data = dat_drg_est, REML = FALSE)
  fit_pbo_est = lmer(y ~ tt + I(tt^2) + w + w * tt +
                       w * I(tt^2) + (tt+I(tt^2)|subj),
                     data = dat_pbo_est, REML = FALSE)

  # estimate the beta and gamma and D
  beta1 = as.matrix(fixef(fit_drg_est))[2:3]
  gamma1 = as.matrix(fixef(fit_drg_est))[5:6]
  D1 = as.matrix(VarCorr(fit_drg_est)$subj)[2:3, 2:3]

  beta2 = as.matrix(fixef(fit_pbo_est))[2:3]
```

```
  gamma2 = as.matrix(fixef(fit_pbo_est))[5:6]
  D2 = as.matrix(VarCorr(fit_pbo_est)$subj)[2:3, 2:3]

  # calculate the purity
  b = purity_calculation(dat_est, beta1, beta2, gamma1, gamma2, D1, D2, p=2)
  return((b))
}
```

Calculate the purity:

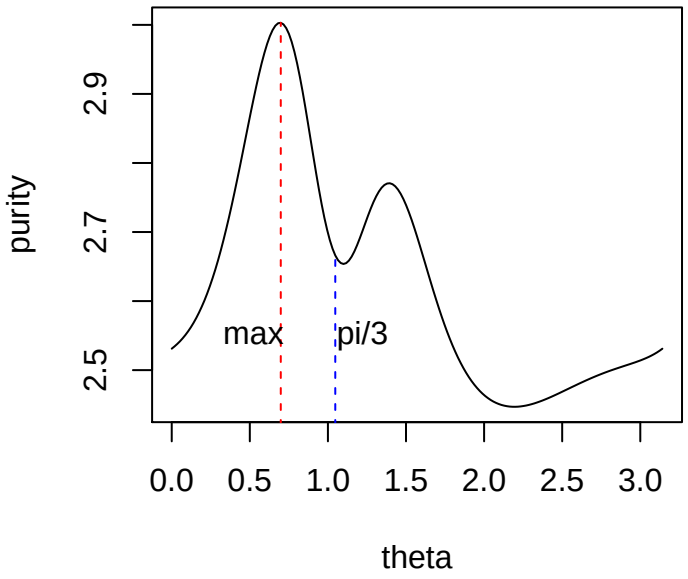
```
purity_calculation = function(dat, beta1, beta2, gamma1, gamma2, D1, D2, p=2){
  unique_dat = unique(dat[,c('subj','w')])
  purity = c()
  for(i in 1:dim(unique_dat)[1]){
    # estimate the mu
    mu1 = beta1 + gamma1 * unique_dat$w[i]
    mu2 = beta2 + gamma2 * unique_dat$w[i]
    # calculate the purity function through monte carlo
    res = monta_carlo_pdf(Xstart, mu1, D1, mu2, D2)/1000
    purity = c(purity, res)
  }
  return(purity)
}
```

Step 7,8,9

With different α values, we could get different w and then different estimations of β, Γ and D. Then follow the above steps we can get new purity values.

Results

The results can be affected by the sample size and Monte Carlo simulation sample size. For example, if we generated 100 subjects in each group and 1000 points for Monte Carlo simulation, the result is:



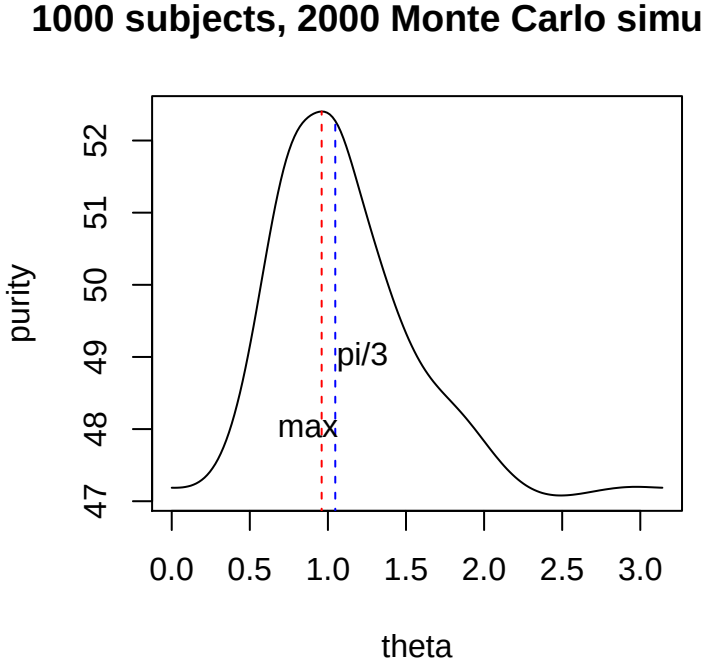
- p.s. I used $\theta = x * \pi/180$ in the plot, since it can be integers and more clear.

The max purity is at $\theta = 40 * \pi/180 \approx 0.698$:

```
data[data$y == max(data$y),]
```

```
##     x        y
## 41 40 3.003013
```

If we increase the sample size, e.g. 1000 subjects in each group and 2000 points for Monte Carlo simulation, the result is:



The max purity is at $\theta = 55 * \pi/180 \approx 0.960$:

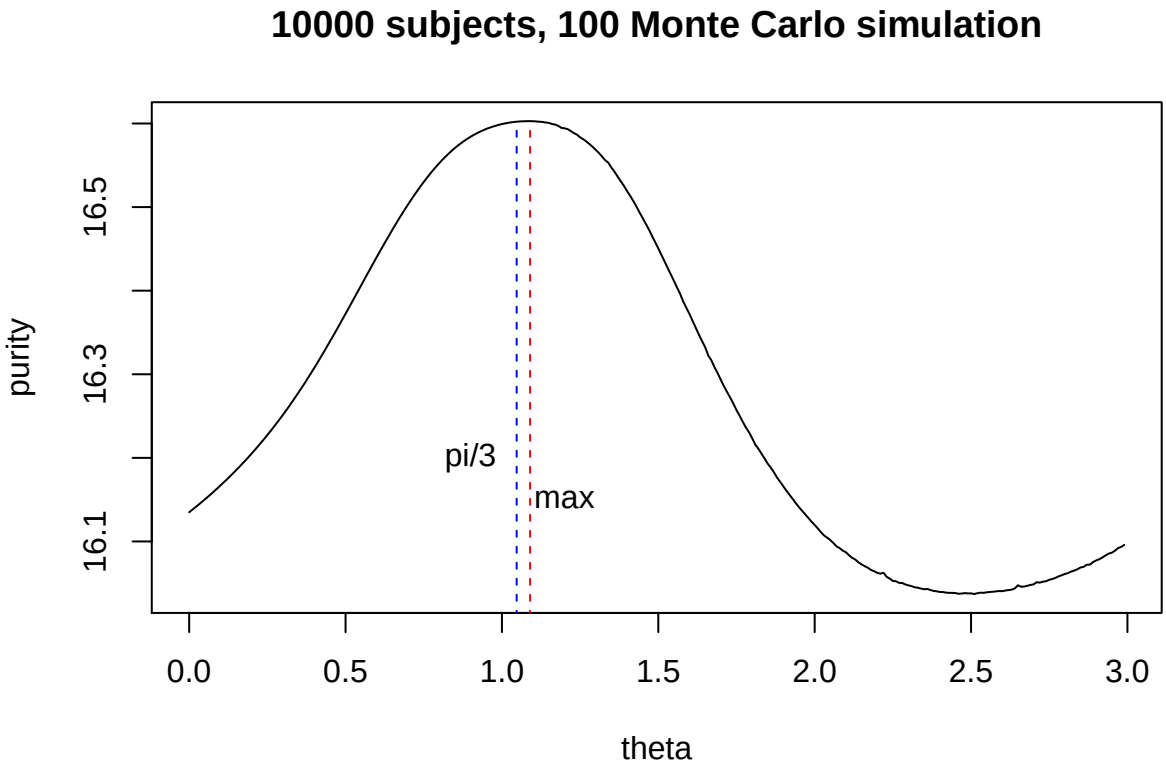
```
data[data$y == max(data$y),]
```

```
##     x        y
## 56 55 52.40269
```

The max purity point is getting closer to the real point.

Maybe if we keep increasing the sample size, we could get the ideal results. (I am still running the 10000 sample size scenario. However, it runs very slowly.)

Larger sample size




```
data[data$y == max(data$y), ]
```

```
##         x        y
## 110 1.09 16.60283
```

```
round(pi/3,2)
```

```
## [1] 1.05
```

P.S.: The true purity

There are two kinds of "true purity":

- True purity 1: the really true purity. Calculated with the true β, Γ and D covariance matrix.
- True purity 2: relatvie true purity. Calulcated with the $\hat{\beta}$, $\hat{\Gamma}$ and $\hat{D}$ covariance matrix estimated by the true θ_0.

We expect that the true purity 1 should be bigger than the true purity 2. However, it may not always true. In some simulations, the true purity 1 is less. (in my simulation, it is much less, about 40, while the true purity 2 got value about 52)

I think this may because of the estimation from LMM, since the purity calculation depends on the two estimated MVNs, and the two MVNs depend on the estimated LMM.

For example, if the $\theta = \theta_0$, which is the true θ, the w should be the same as the original data since we used the same θ. The estimated β and Γ should be very close to our selected values at the very begining, e.g. $\hat{\beta}_{drg} \approx (0, -2, -1), \hat{\beta}_{pbo} \approx (0, 2, 1)$. And the $\hat{\mu} = \beta + \Gamma * w$.

The differences between μ_1 and μ_2 can be larger because of the variation of estimation, e.g. $\hat{\beta}_{drg} = (0, -5.01, -1.01)$, $\hat{\beta}_{drg} = (0, 5.01, 1.01)$, $\hat{\Gamma}_{drg} \approx (0, -2.01, -1.01)$ and $\hat{\Gamma}_{pbo} \approx (0, 2.01, 1.01)$. Then the differences between those two multivariate normal distributions can be larger than the real scenario. And when we increasing the sample size, the differences should be smaller.