# purity

February 19, 2019

## 0.1 Try to calculate the purity with generated covariates

Date: 2019-2-20

To calculate the max purity function, first, fit the linear mixed model for the outcome $\mathbf{y_i}$ and time $\mathbf{X_i}$, with baseline covariates $\mathbf{x_i}$:

$$\mathbf{y_i} = \mathbf{X_i}(\mathbf{fi} + \mathbf{b_i} + \mathbf{\Gamma}(\mathbf{ff}'\mathbf{x_i})) + \mathbf{ffl_i}.$$

where,

- $\mathbf{y_i}$ is the vector of outcomes for the $i$th subject, i.e., the dimension of $\mathbf{y_i}$ is ($n_i$, 1). $n_i$ is the number of observations for the $i$th subject.
- $\mathbf{X_i}$ is the covariate matrix for $i$th subject. The dimension is ($n_i$, p). Here in our example, it is ($n_i$,3).
- $\mathbf{fi}$ is the coeffcient vector for the fixed effects of $\mathbf{X_i}$. Dimension is (p,1).
- $\mathbf{x_i}$ is the vector of baseline covariates for the subject. The dimension is (q, 1).
- $\mathbf{b_i}$ is the vector of random effects. Dimension is (p,1).
- $\mathbf{\Gamma}$ is the vector of fixed effects of the baseline covariates. Dimension is (p,1).
- $w_i = \mathbf{ff}'\mathbf{x_i}$ is the combination of the input baseline covariates. $w_i$ is a scalar.

We can define the covariate matrix of $\mathbf{X_i}$ as $\mathbf{z_i}$. The $\mathbf{z_i}$ contains both fixed effects and random effects.

$$\mathbf{z_i} = \mathbf{fi} + \mathbf{b_i} + \mathbf{\Gamma} w_i$$

We can also write the above equation in the matrix version:

$$Y_i = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ ... & ... & ... \\ 1 & t_{n_i} & t_{n_i}^2 \end{bmatrix} \left[ \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} b_{i0} \\ b_{i1} \\ b_{i2} \end{pmatrix} + \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \end{pmatrix} w_i \right] + \mathbf{ffl_i}$$

For different subjects, they have the same $\mathbf{fi}$ vector and the same $\mathbf{\Gamma}$ vector. Their random effect vector $\mathbf{b_i}$ are different.

# 1 Step 1:

### 1.0.1 Estimate the fi ,Γ, $b_i$ in R

**Read in data**

```
In [4]: setwd('/Users/yaolanqiu/Desktop/NYU/rotation/Rotation2/Week3/from dr.tarpey')

        library(lme4)
        source("cvxcluster-0513.R")

        dat = read.table("hcaf.dat", header=T)
        dim(dat) # 3364 7
```

1. 3364 2. 7

**Define the covariates** First let's try Baseline CGI Let the 'newcov' = 'BaselineCGI'

```
In [5]: dat$newcov = dat$BaselineCGI
```

**Fit LME**

```
In [6]: d0 = dat[dat$trt == 0 ,]
        d1 = dat[dat$trt == 1 ,]
        fit_d0 = lmer(y ~ t1 + I(t1^2) + newcov + newcov * t1 +
                        newcov * I(t1^2) + (t1+I(t1^2)|subj),
                      data = d0, REML = FALSE)
        fit_d1 = lmer(y ~ t1 + I(t1^2) + newcov + newcov * t1 +
                        newcov * I(t1^2) + (t1+I(t1^2)|subj),
                      data = d1, REML = FALSE)
```

```
singular fit
```

The estimated beta and gamma is

```
In [7]: beta_d0 = as.matrix(fixef(fit_d0)[1:3])
        gamma_d0 = as.matrix(fixef(fit_d0)[4:6])
        beta_d1 = as.matrix(fixef(fit_d1)[1:3])
        gamma_d1 = as.matrix(fixef(fit_d1)[4:6])
```

```
In [8]: beta_d0
```

| | |
|---|---|
| (Intercept) | 14.0172657 |
| t1 | -5.0133105 |
| I(t1^2) | 0.7171924 |

```
In [9]: gamma_d0
```

| | |
|---|---|
| newcov | 2.04212314 |
| t1:newcov | 0.17005552 |
| I(t1^2):newcov | -0.04868318 |

```
In [10]: beta_d1
```

| | |
|---|---|
| (Intercept) | 13.455594 |
| t1 | -3.400881 |
| I(t1^2) | 0.484097 |

```
In [11]: gamma_d1
```

| | |
|---|---|
| newcov | 2.28686752 |
| t1:newcov | -0.19895847 |
| I(t1^2):newcov | -0.02576792 |

```
In [12]: beta0 = as.matrix(fixef(fit_d0)[1:3])
         gamma0 = as.matrix(fixef(fit_d0)[4:6])

         beta1 = as.matrix(fixef(fit_d1)[1:3])
         gamma1 = as.matrix(fixef(fit_d1)[4:6])
```

## 2 Step 2:

estimate the distribution of $\mathbf{z_i}$ for drug group and placebo, separately.

$$\begin{aligned} f(z_i) &= \int_{w_i} f(z_i, w_i) dw_i \\ &= \int_{w_i} f(z_i|w_i) g(w_i) dw_i \end{aligned}$$

where,

- the conditional distribution $f(z_i|w_i) \sim MVN(\mathbf{fi} + \mathbf{\Gamma} w_i, \mathbf{D_i})$
- $g(w_i)$ is the distribution of the covariates combination. For example, if the covariates combination only contains "sex", which is binary, then the intergal becomes summation.
- $\mathbf{D_i}$ is the covariates matrix of random effects $\mathbf{b_i}$.

We could then fit the $f(\cdot)$ for drug group and placebo group separately, i.e. $f_1(z)$ and $f_2(z)$.

## 3 Step 3: Define the purity

The purity should be a function of $w_i$. First calculate the integral of $z_i$. Then,

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

**calculate the integral and purity**

In [13]:
```
quadratic0 = function(a,b) {
    X = matrix(c(a,b),nrow=2)
    Q = (-1/2)*t(X-mu0)%*%solve(sigma0)%*%(X-mu0)
}
quadratic1 = function(a,b) {
    X = matrix(c(a,b),nrow=2)
    Q = (-1/2)*t(X-mu1)%*%solve(sigma1)%*%(X-mu1)
}

PDF = function(x) {
  f0 = (1/(2*pi))*(1/sqrt(det(sigma0)))*exp(quadratic0(x[1],x[2]))
  f1 = (1/(2*pi))*(1/sqrt(det(sigma1)))*exp(quadratic1(x[1],x[2]))
  if((f0 + f1)!=0){
      res = (f0 - f1)^2 / (f0 + f1)
    }else{res = 0}
  return(res)
}
```

In [16]:
```
library(cubature)
# get unique w value for each subject
W = data.frame(subj = dat$subj, newcov = dat$newcov)
W = unique(W)

P = c() # save the purity value in P
for(i in W$newcov){
    w = i
    m0 = beta0 + gamma0 * w; m0 = m0[2:3]
    m1 = beta1 + gamma1 * w; m1 = m1[2:3]
    D0 = as.matrix(VarCorr(fit_d0)$subj)[2:3, 2:3]
    D1 = as.matrix(VarCorr(fit_d1)$subj)[2:3, 2:3]

    mu0 =  matrix(m0, nrow=2)
    sigma0 = D0
    mu1 = matrix(m1, nrow=2)
    sigma1 = D1

    P = c(P,adaptIntegrate(PDF,
                           lowerLimit= c(-1,-1),
                           upperLimit=c(1,1))$integral)
  }
mean(P)
sum(P)
```

```
0.0303075344326379
16.4569911969224
```

**Define the covariates 2** First let's generate covariates by ourselves

In [20]:
```
# create new covariates
# read in data
dat = read.table("hcaf.dat", header=T)
d0 = dat[dat$trt == 0,]
d1 = dat[dat$trt == 1,]

cov01 = rnorm(length(unique(d0$subj)),5,1)
cov02 = rnorm(length(unique(d0$subj)),10,1)
newcov0 = data.frame(subj = unique(d0$subj), newcov1 = cov01, newcov2 = cov02)
d0 = merge(d0,newcov0, by = 'subj')

# create new covariates
cov01 = rnorm(length(unique(d1$subj)),10,1)
cov02 = rnorm(length(unique(d1$subj)),5,1)
newcov1 = data.frame(subj = unique(d1$subj), newcov1 = cov01, newcov2 = cov02)
d1 = merge(d1,newcov1, by = 'subj')

# new covariate, which is the combination of the two new covariates
# let's make it a simple summation first
d0$newcov = d0$newcov1 + d0$newcov2
d1$newcov = d1$newcov1 + d1$newcov2

dat = rbind(d0, d1)
d0 = dat[dat$trt == 0,]
d1 = dat[dat$trt == 1,]
head(dat)
```

| subj | trt | y | age | BaselineCGI | t1 | responder | newcov1 | newcov2 | newcov |
|---|---|---|---|---|---|---|---|---|---|
| 2497 | 0 | 25 | 29 | 4 | 0 | 0 | 4.726312 | 11.97518 | 16.70149 |
| 2497 | 0 | 18 | 29 | 4 | 1 | 0 | 4.726312 | 11.97518 | 16.70149 |
| 2497 | 0 | 11 | 29 | 4 | 2 | 0 | 4.726312 | 11.97518 | 16.70149 |
| 2497 | 0 | 9 | 29 | 4 | 3 | 0 | 4.726312 | 11.97518 | 16.70149 |
| 2497 | 0 | 19 | 29 | 4 | 4 | 0 | 4.726312 | 11.97518 | 16.70149 |
| 2497 | 0 | 15 | 29 | 4 | 5 | 0 | 4.726312 | 11.97518 | 16.70149 |

In [21]:
```
# get unique w value for each subject
W = data.frame(subj = dat$subj, newcov = dat$newcov)
W = unique(W)

P = c() # save the purity value in P
for(i in W$newcov){
    w = i
    m0 = beta0 + gamma0 * w; m0 = m0[2:3]
    m1 = beta1 + gamma1 * w; m1 = m1[2:3]
    D0 = as.matrix(VarCorr(fit_d0)$subj)[2:3, 2:3]
    D1 = as.matrix(VarCorr(fit_d1)$subj)[2:3, 2:3]

    mu0 =  matrix(m0, nrow=2)
```

```
    sigma0 = D0
    mu1 = matrix(m1, nrow=2)
    sigma1 = D1

    P = c(P,adaptIntegrate(PDF,
                           lowerLimit= c(-1,-1),
                           upperLimit=c(1,1))$integral)
  }
mean(P)
sum(P)
```

0.183757890167255
65.7853246798773

The value is bigger, which means that the new covariate can help classify groups better than baseline CGI

**Define the covariates 3** First let's generate covariates by ourselves

In [22]:
```
# create new covariates
# read in data
dat = read.table("hcaf.dat", header=T)
d0 = dat[dat$trt == 0,]
d1 = dat[dat$trt == 1,]

cov01 = rnorm(length(unique(d0$subj)),5,1)
cov02 = rnorm(length(unique(d0$subj)),10,1)
newcov0 = data.frame(subj = unique(d0$subj), newcov1 = cov01, newcov2 = cov02)
d0 = merge(d0,newcov0, by = 'subj')

# create new covariates
cov01 = rnorm(length(unique(d1$subj)),10,1)
cov02 = rnorm(length(unique(d1$subj)),5,1)
newcov1 = data.frame(subj = unique(d1$subj), newcov1 = cov01, newcov2 = cov02)
d1 = merge(d1,newcov1, by = 'subj')

# new covariate, which is the combination of the two new covariates
# let's make it a simple summation first
d0$newcov = d0$newcov1 #+ d0$newcov2
d1$newcov = d1$newcov1 #+ d1$newcov2

dat = rbind(d0, d1)
d0 = dat[dat$trt == 0,]
d1 = dat[dat$trt == 1,]
head(dat)
```

| subj | trt | y | age | BaselineCGI | t1 | responder | newcov1 | newcov2 | newcov |
|---|---|---|---|---|---|---|---|---|---|
| 2497 | 0 | 25 | 29 | 4 | 0 | 0 | 5.638603 | 8.842928 | 5.638603 |
| 2497 | 0 | 18 | 29 | 4 | 1 | 0 | 5.638603 | 8.842928 | 5.638603 |
| 2497 | 0 | 11 | 29 | 4 | 2 | 0 | 5.638603 | 8.842928 | 5.638603 |
| 2497 | 0 | 9 | 29 | 4 | 3 | 0 | 5.638603 | 8.842928 | 5.638603 |
| 2497 | 0 | 19 | 29 | 4 | 4 | 0 | 5.638603 | 8.842928 | 5.638603 |
| 2497 | 0 | 15 | 29 | 4 | 5 | 0 | 5.638603 | 8.842928 | 5.638603 |

In [23]:
```
# get unique w value for each subject
W = data.frame(subj = dat$subj, newcov = dat$newcov)
W = unique(W)

P = c() # save the purity value in P
for(i in W$newcov){
    w = i
    m0 = beta0 + gamma0 * w; m0 = m0[2:3]
    m1 = beta1 + gamma1 * w; m1 = m1[2:3]
    D0 = as.matrix(VarCorr(fit_d0)$subj)[2:3, 2:3]
    D1 = as.matrix(VarCorr(fit_d1)$subj)[2:3, 2:3]

    mu0 =  matrix(m0, nrow=2)
    sigma0 = D0
    mu1 = matrix(m1, nrow=2)
    sigma1 = D1

    P = c(P,adaptIntegrate(PDF,
                           lowerLimit= c(-1,-1),
                           upperLimit=c(1,1))$integral)
  }
mean(P)
sum(P)
```

```
0.0689950340817801
24.7002222012773
```

In [24]:
```
# create new covariates
# read in data
dat = read.table("hcaf.dat", header=T)
d0 = dat[dat$trt == 0,]
d1 = dat[dat$trt == 1,]

cov01 = rnorm(length(unique(d0$subj)),5,1)
cov02 = rnorm(length(unique(d0$subj)),10,1)
newcov0 = data.frame(subj = unique(d0$subj), newcov1 = cov01, newcov2 = cov02)
d0 = merge(d0,newcov0, by = 'subj')

# create new covariates
cov01 = rnorm(length(unique(d1$subj)),10,1)
cov02 = rnorm(length(unique(d1$subj)),5,1)
```

```
newcov1 = data.frame(subj = unique(d1$subj), newcov1 = cov01, newcov2 = cov02)
d1 = merge(d1,newcov1, by = 'subj')

# new covariate, which is the combination of the two new covariates
# let's make it a simple summation first
d0$newcov = 10 * d0$newcov1 #+ d0$newcov2
d1$newcov = 10 * d1$newcov1 #+ d1$newcov2

dat = rbind(d0, d1)
d0 = dat[dat$trt == 0,]
d1 = dat[dat$trt == 1,]
head(dat)
```

| subj | trt | y | age | BaselineCGI | t1 | responder | newcov1 | newcov2 | newcov |
|---|---|---|---|---|---|---|---|---|---|
| 2497 | 0 | 25 | 29 | 4 | 0 | 0 | 4.703117 | 8.723089 | 47.03117 |
| 2497 | 0 | 18 | 29 | 4 | 1 | 0 | 4.703117 | 8.723089 | 47.03117 |
| 2497 | 0 | 11 | 29 | 4 | 2 | 0 | 4.703117 | 8.723089 | 47.03117 |
| 2497 | 0 | 9 | 29 | 4 | 3 | 0 | 4.703117 | 8.723089 | 47.03117 |
| 2497 | 0 | 19 | 29 | 4 | 4 | 0 | 4.703117 | 8.723089 | 47.03117 |
| 2497 | 0 | 15 | 29 | 4 | 5 | 0 | 4.703117 | 8.723089 | 47.03117 |

In [25]:
```
# get unique w value for each subject
W = data.frame(subj = dat$subj, newcov = dat$newcov)
W = unique(W)

P = c() # save the purity value in P
for(i in W$newcov){
    w = i
    m0 = beta0 + gamma0 * w; m0 = m0[2:3]
    m1 = beta1 + gamma1 * w; m1 = m1[2:3]
    D0 = as.matrix(VarCorr(fit_d0)$subj)[2:3, 2:3]
    D1 = as.matrix(VarCorr(fit_d1)$subj)[2:3, 2:3]

    mu0 =  matrix(m0, nrow=2)
    sigma0 = D0
    mu1 = matrix(m1, nrow=2)
    sigma1 = D1

    P = c(P,adaptIntegrate(PDF,
                           lowerLimit= c(-1,-1),
                           upperLimit=c(1,1))$integral)
  }
mean(P)
sum(P)
```

```
0.0269238927251193
9.63875359559269
```

In [26]:
```
# create new covariates
# read in data
```

```
dat = read.table("hcaf.dat", header=T)
d0 = dat[dat$trt == 0,]
d1 = dat[dat$trt == 1,]

cov01 = rnorm(length(unique(d0$subj)),5,1)
cov02 = rnorm(length(unique(d0$subj)),10,1)
newcov0 = data.frame(subj = unique(d0$subj), newcov1 = cov01, newcov2 = cov02)
d0 = merge(d0,newcov0, by = 'subj')

# create new covariates
cov01 = rnorm(length(unique(d1$subj)),10,1)
cov02 = rnorm(length(unique(d1$subj)),5,1)
newcov1 = data.frame(subj = unique(d1$subj), newcov1 = cov01, newcov2 = cov02)
d1 = merge(d1,newcov1, by = 'subj')

# new covariate, which is the combination of the two new covariates
# let's make it a simple summation first
d0$newcov = d0$newcov1 #+ d0$newcov2
d1$newcov = 10 * d1$newcov1 #+ d1$newcov2

dat = rbind(d0, d1)
d0 = dat[dat$trt == 0,]
d1 = dat[dat$trt == 1,]
head(dat)
```

| subj | trt | y | age | BaselineCGI | t1 | responder | newcov1 | newcov2 | newcov |
|---|---|---|---|---|---|---|---|---|---|
| 2497 | 0 | 25 | 29 | 4 | 0 | 0 | 3.34943 | 8.917454 | 3.34943 |
| 2497 | 0 | 18 | 29 | 4 | 1 | 0 | 3.34943 | 8.917454 | 3.34943 |
| 2497 | 0 | 11 | 29 | 4 | 2 | 0 | 3.34943 | 8.917454 | 3.34943 |
| 2497 | 0 | 9 | 29 | 4 | 3 | 0 | 3.34943 | 8.917454 | 3.34943 |
| 2497 | 0 | 19 | 29 | 4 | 4 | 0 | 3.34943 | 8.917454 | 3.34943 |
| 2497 | 0 | 15 | 29 | 4 | 5 | 0 | 3.34943 | 8.917454 | 3.34943 |

In [27]:
```
# get unique w value for each subject
W = data.frame(subj = dat$subj, newcov = dat$newcov)
W = unique(W)

P = c() # save the purity value in P
for(i in W$newcov){
    w = i
    m0 = beta0 + gamma0 * w; m0 = m0[2:3]
    m1 = beta1 + gamma1 * w; m1 = m1[2:3]
    D0 = as.matrix(VarCorr(fit_d0)$subj)[2:3, 2:3]
    D1 = as.matrix(VarCorr(fit_d1)$subj)[2:3, 2:3]

    mu0 =  matrix(m0, nrow=2)
    sigma0 = D0
    mu1 = matrix(m1, nrow=2)
```

```
    sigma1 = D1

    P = c(P,adaptIntegrate(PDF,
                           lowerLimit= c(-1,-1),
                           upperLimit=c(1,1))$integral)
  }
mean(P)
sum(P)
```

0.0153922200318381
5.51041477139805