DyAlloc-vignette

Disclaimer

Below is the Vignette for my DyAlloc package. The whole package is now on my GitHub and will be submitted on CRAN. This is the link to the package on my GitHub:

https://github.com/ktmiu/DyAlloc/tree/main/Desktop/Cov_project/DyAlloc

Please refer to it as my second deliverable. Thank you!

Overview

DyAlloc is the implementation of Dynamic Allocation/Covariate Adaptive Randomization. In clinical studies, randomized controlled trials are often used to maintain unbiased treatment comparison. When assigning subjects to treatment groups, complete randomization can avoid selection bias. However, by chance alone, simple randomization has been realized to result in imbalances, or unequal allocation, of treatment assignment across study covariates. This is especially true in small trials or in large trials with many strata. Given equal variance of treatment responses, this will compromise the statistical properties of the study and lead to reduced precision of the treatment estimate, and thus increased likelihood of Type II errors (Lebowitsch et al, 2012). Therefore, to achieve unbiasedness in subject allocation while maintaining balance over covariates, this dynamic allocation scheme has been proposed.

Having balanced covariates in clinical trials is crucial. First, it improves the credibility of the trial. When it comes to interpreting the analysis results, minimization of unbalances during randomization is easier to comprehend than having to adjust the analysis after randomization. Second, balance makes conducting interim analysis, subgroup analysis, or secondary endpoints analysis, etc., much easier. Lastly, we are less likely to have power loss due to misspecified models (McEntegart, 2003).

With the advance of personalized medicine and new-found biomarkers, clinical trials often involve multiple centers and an increasingly large number of covariates. This proposes a challenge to balance on a stratified level while accounting for interactions between the covariates. In addition, stratified randomization does not promise overall balance of the entire study or across different factors. For this reason, minimization (dynamic allocation) provides a solution, which balances treatment assignment over covariate margins.

Method

Dynamic Allocation is an unbalance minimization method proposed by Lebowitsch et al. They referred to it as the multidimensional dynamic allocation (MDA) method. Essentially, for each new subject coming into the study, they are assigned a treatment group based on the existing subjects within each treatment arm. This method approaches minimizing unbalances at different levels simultaneously, including the strata, factors, and overall study. The choice of weight for each level is decided by the investigators depending on the study design. Furthermore, this method can be applied to any number of treatment arms.

Function Descriptions

1. Marginal balance as distance measure - measuring imbalance within a population

A low-level function that computes an imbalance score for each treatment given a set of data. This function is used primarily used in the subsequent functions and will most probably not be used directly by a practitioner.

- Function syntax: **MarImb (df, N)**
- INPUT: This function takes two arguments:
  - df: the data frame of the study
  - N: the number of treatment arms
- OUTPUT: a vector of marginal imbalance scores for each treatment arm is produced (a total of N scores)

2. Weighting imbalances - summing imbalances across populations

This function creates an imbalance score for a new subject to be assigned each possible treatment given data from previous subjects.

- Function syntax: **WeiImb (df, N, covars, weight, newsub, site=FALSE)**
- INPUT: This function takes six arguments:
  - df: the data frame of the study
  - N: the number of treatment arms
  - covars: a vector of the name of covariates in the study
  - weight: a vector of weights for overall study, with-in stratum, (site), and factors(covariates), in this particular order
  - newsub: a vector of the factor profile for the new subject
  - site: whether or not to account for site imbalance (default:FALSE)
- OUTPUT: a vector of weighted imbalance scores for each treatment arm is produced (a total of N scores)

NOTE: the default for site here is FALSE, but one can specify site=TRUE in the argument to account for site imbalance. Note that the weight for site has to be placed in the third element of the 'weight' vector (and can be omitted when site=FALSE)

3. Treatment asssignment probabilities

This function uses the imbalance scores created from the function WeiImb to produce a vector of probabilities of a new subject being assigned to each treatment group.

- Function syntax: **Trt_Prob (imbalances, alpha)**
- INPUT: This function takes two arguments:
  - imbalances: imbalance scores across the N treatment groups (generally, the scores we get from WeiImb would be plugged in here)
  - alpha: the "second best probability" parameter, generally alpha is set to 0.2. This means that the treat group a subject assign to gets the probability of 0.8.
- OUTPUT: a vector of probabilities of being assigned to each treatment group

NOTE: if the imbalance scores are the same, then they have equal probability of being assigned to. E.g. if we get two equal least weighted imbalance scores, they each have a probability of 0.5 for a subject to be assigned to.

4. Final treatment assignment - used alone

This function incorporates all the previous functions and gives a final treatment group assignment to a new subject. A practitioner can use this function solely to obtain treatment assignments for the study subjects.

- Function syntax: **CAR (df, N, covars, weight, newsub, site=FALSE)**
- INPUT: This function takes six arguments:
  - df: the data frame of the study
  - N: the number of treatment arms
  - covars: a vector of the name of covariates in the study
  - weight: a vector of weights for overall study, with-in stratum, (site), and factors(covariates), in this particular order
  - newsub: a vector of the factor profile for the new subject
  - site: whether or not to account for site imbalance (default:FALSE)
- OUTPUT: a treatment group assignment

*NOTE: like the WeiImb function, the site default is FALSE.

Examples of implementing the functions

```
library(DyAlloc)
```

Without accounting for site imbalance

Here, we considered a clinical trial with 4 covariates: "gender","age","race","trauma severity", and 4 treatment groups. There are 50 randomized subjects already in the study and we wish to assign the 51st patient to the least imbalanced group. Produced is the total imbalance score across the 4 treatment groups and the probability of the 51st patient being assigned to each treatment.

```
#generate a data frame of 50 subjects in a 4 treatment group study
df1 <- data.frame("gender" = sample(c("female", "male"),50, TRUE),
                  "age" = sample(c("0-35", ">35"),50, TRUE),
                  "race" = sample(c("black", "white"),50,TRUE),
                  "trauma severity"= sample(c("good", "bad"),50,TRUE),
                  "treatment"=sample(c("1","2","3","4"),50, TRUE,c(1,1,5,10)),
                   stringsAsFactors = TRUE)
#covariates of the study
covars <-c("gender","age","race","trauma severity")
#new subject's profile
newsub <-c("male","0-35","black","bad")
#weight of the overall study, the stratum, and the 4 factors
weight <-c(1,1,rep(1,4))
#print out the final weighted imbalance scores
imb <-WeiImb(df1,4,covars,weight,newsub,FALSE)
imb
#> [1] 6.494398 6.661064 6.866947 6.938375
#probabilities for the new subject to be assigned to the each of the treatment group
Trt_Prob(imb,0.2)
#> [1] 0.8 0.2 0.0 0.0

#Using CAR function to get the final treatment assignment
DyAlloc(df1,4,covars,weight,newsub,FALSE)
#> [1] "2"
```

The above example shows that the ratio of treatment 1 to 4 already assigned in the study is 1:1:5:10. Given equal weights to each treatment arm, we can see that the probability of the 51st subject's assignment are most likely to fall into treatment 1 or 2 with a 50/50 chance, which are the least imbalanced ones.

Accounting for site imbalance

The same data frame but also accounting for site in the total imbalance score.

```
df <- data.frame("gender" = sample(c("female", "male"),50, TRUE),
                 "age" = sample(c("0-35", ">35"), 50, TRUE),
                 "race" = sample(c("black", "white"),50,TRUE),
                 "trauma severity"= sample(c("good", "bad"),50,TRUE),
                 "treatment"=sample(c("1","2","3","4"),50, TRUE),
                 "site"=sample(c("1","2"),50, TRUE),
                 stringsAsFactors = TRUE)
covars <-c("gender","age","race","trauma severity")
#new subject's profile with the first element being site
newsub<-c("2","male","0-35","black","bad")
#weight of the overall study, the stratum, the site, and the 4 factors
weight <-c(1,1,1,rep(1,4))
imb<-WeiImb(df,4,covars,weight,newsub,TRUE)
imb
#> [1] 5.658789 5.587360 5.927171 5.731092
Trt_Prob(imb,0.2)
#> [1] 0.2 0.8 0.0 0.0
DyAlloc(df,4,covars,weight,newsub,TRUE)
#> [1] "2"
```

This example shows that when there is random treatment assignment in the study, given equal weights to each treatment arm, the 51st subject's assignemnt can also be random. Under the "second best probability" paradigm, the final treatment assignment would most likely fall into the treatment arm with a probability of 0.8.

Simulation

200 Subject profiles were simulated with 4 covariates considered - gender, age, race, and trauma severity, in a 4-treatment-arms clinical trial. Site imbalance is not considered in this scenario. Each subject is assigned to a treatment group after the simulation using two methods: covariate adaptive randomization and simple randomization. To make this more interesting and to see the effect of balanced allocation, the ratio of female to male is set to be 1:4, and the ratio of Black to White is 7:3.

It can be seen that the covariate adaptive randomization did a very good job in balancing treatment assignment across individual variables as well as across strata defined by interactions between variables, especially compared to simple randomization. For example, no White females were asigned to treatment 1 using simple randomization, while exactly 25% of White females were assigned to treatment 1 using covaraite adaptive randomization, which is the target proportion.

i) Table content for the number of female vs. male
ii) Table for the proportion of each treatment assignment in female and in male using the covariate adaptive randomization
iii) Table for the proportion of each treatment assignment in female and in male using simple randomization

```
#>
#> female   male
#>     48    152
#>
#>                  1         2         3         4
#>   female 0.2291667 0.2291667 0.2708333 0.2708333
#>   male   0.2565789 0.2565789 0.2434211 0.2434211
#>
```

```
#>                  1         2         3         4
#>   female 0.1875000 0.4166667 0.1458333 0.2500000
#>   male   0.2434211 0.2565789 0.2500000 0.2500000
```

i) Table content for the number of Black vs. White
ii) Table for the proportion of each treatment assignment in Black and in White using the covariate adaptive randomization
iii) Table for the proportion of each treatment assignment in Black and in White using simple randomization

```
#>
#> black white
#>   141    59
#>
#>                 1         2         3         4
#>   black 0.2482270 0.2553191 0.2482270 0.2482270
#>   white 0.2542373 0.2372881 0.2542373 0.2542373
#>
#>                 1         2         3         4
#>   black 0.2482270 0.3191489 0.2198582 0.2127660
#>   white 0.1864407 0.2372881 0.2372881 0.3389831
```

i) Table content for the number in each of the 4 combinations of gender and race
ii) Table for the proportion of each treatment assignment in each of the 4 combinations of gender and race using the covariate adaptive randomization
iii) Table for the proportion of each treatment assignment in each of the 4 combinations of gender and race using simple randomization

```
#>
#>         female male
#>   black     36  105
#>   white     12   47
#>                       1         2         3         4
#> black female 0.2222222 0.2500000 0.2500000 0.2777778
#> white female 0.2500000 0.1666667 0.3333333 0.2500000
#> black male   0.2571429 0.2571429 0.2476190 0.2380952
#> white male   0.2553191 0.2553191 0.2340426 0.2553191
#>                       1         2          3         4
#> black female 0.2500000 0.3888889 0.16666667 0.1944444
#> white female 0.0000000 0.5000000 0.08333333 0.4166667
#> black male   0.2476190 0.2952381 0.23809524 0.2190476
#> white male   0.2340426 0.1702128 0.27659574 0.3191489
```