R implementation of robust Multiple Imputation for Compositional Data adjusting for covarites (`MICoDa`)

A. Saha, R. Sundaram, S. Peddada

1.Introduction

In this article, we show how `MICoDa` can be implemented for imputing compositional outcomes adjusting for covariates. To begin with, we generated a data from an Additive log ratio (ALR) model such as that we obtain six activity categories with a total constrained to be 30. The six activity categories include a category called "other_a20", which represents a reference category. This category represents activity hours not accounted for in the first five categories. All categories must have either a strictly positive quantity or a missing value (NA). Among the covarites we have considered two continuous covariates, namely "age", "bmi", ; 2 binary variables, namely "married", "abnorm" indicating marital status or abnormal birth weight status, and one nominal categorical variable, namely education level of parents having four ordinal levels. In the next section, we first load the required R-functions and the supplied data-sets.

2. Loading full data, and partially missing data along with the libraries

```
source("./R_Aux_scripts/MICoDa_Aux.R")  # includes various auxiliary functions including
                                         # the key R function MICoDa
data_full <- readRDS("./metadata/data_full.rds") # data-set without any missing value
data_NA <- readRDS("./metadata/data_with_NA.rds")
data_full[c(1:3,8),]  #printing few rows
```

```
##   total_screen_a20 total_read_a20 total_cogn_a20 total_phys_a20 total_sleep_a20
## 1         2.069573      2.0186056       1.406962      1.1189963       14.658698
## 2         4.782267      1.6078515       1.029694      2.3959613        8.217522
## 3         1.838219      5.3905629       1.976403      3.8322814        7.326951
## 8         2.347313      0.5000859       1.636547      0.9848221       16.266392
##   other_a20       age       bmi married abnorm educ_2 educ_3 educ_4
## 1  8.727165 0.8984977 0.8764558       1      0      0      0      1
## 2 11.966703 1.0020661 0.9113688       1      1      1      0      0
## 3  9.635583 0.9676729 1.2698394       1      1      0      1      0
## 8  8.264839 1.0973451 0.8244510       1      1      1      0      0
```

```
data_NA[c(1:3,8),]  #MICoDa will be applied to this one
```

```
##   total_screen_a20 total_read_a20 total_cogn_a20 total_phys_a20 total_sleep_a20
## 1         2.069573       2.018606       1.406962      1.1189963       14.658698
## 2         4.782267       1.607851       1.029694      2.3959613        8.217522
## 3         1.838219       5.390563             NA      3.8322814        7.326951
## 8               NA             NA             NA      0.9848221       16.266392
##   other_a20       age       bmi married abnorm educ_2 educ_3 educ_4
## 1  8.727165 0.8984977 0.8764558       1      0      0      0      1
## 2 11.966703 1.0020661 0.9113688       1       1       1       0       0
## 3        NA 0.9676729 1.2698394       1       1       0       1       0
## 8        NA 1.0973451 0.8244510       1       1       1       0       0
```

```
sum(is.na(data_NA[,"other_a20"]))/nrow(data_NA) #30% data has at least one misising category
```

```
## [1] 0.3
```

2. R-function, `MICoDa`, and its implementation

$$\textit{function}: \qquad \texttt{MICoDa}\,(train,\, D,\, varlist,\, iter,\, rescaleAll = T, ...,) \tag{1}$$

Description: The function implements `MICoDa` methodology, which starts the procedure by ALR regression on complete data, followed by imputation of missing components either by adjusting all components while sweeping through one component at a time / by restricting to only missing compositions. After first iteration, for next iteration onwards, `MICoDa` updates ALR regression estimates based on whole data including imputed rows followed by re-imputation of originally missing rows. Iteration continues until imputed values converge.

Arguments:

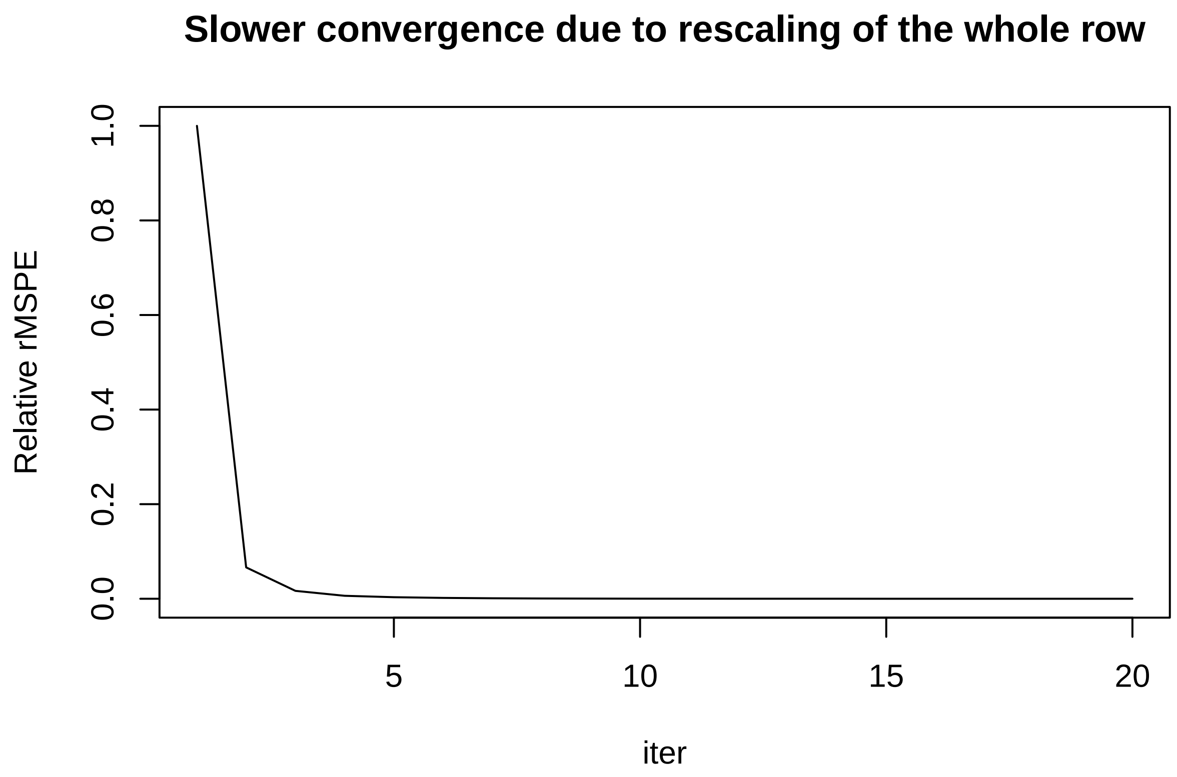
- *train*: a data_frame object representing data with missing values
- D: Dimension of compositional vector. First D variables of the data are assumed to represent the compositions. We further assume that D^{th} composition is the reference variable for ALR regression.
- *varlist*: list of variables indicating covariates to be regressed upon
- *iter*: Number of iteration required for desired tolerance level of convergence criterion based on relative root mean squared prediction error (rMSPE) on the imputed rows between two consecutive iteration. This can be determined using by trial and error, and studying the sequence of $diff$ values as outputted from the function.
- *rescaleAll*: logical variable. By default it is assigned TRUE indicating it will update all components of the rows with missing values. Otherwise, it will only update the missing components.
- ... : optional arguments for *row_partial* function such as the totality constraint given by the argument *Constant*. By default it is assumed to be 24.

Values: The function generates a list of objects including the imputed data (*data_Imp*), ALR regression coefficients at convergence (*Theta_est*), missing rows imputed (*NArows*), and relative rMSPE between consecutive iterations ($diff$)

`MICoDa_V1`: Imputation of missing compositions while re-calibrating the observed compositions.

The following section of codes generates the imputed data, and plots the relative rMSPE while missing rows are fully calibrated.

```
varlist <-  c("age", "bmi", "married","abnorm","educ_2","educ_3","educ_4") # covariate list
# MICoDa_V1: When all components of the missing rows are imputed
rt1<- MICoDa(data_NA, 6, varlist,iter=20, rescaleAll=T,Constant=30)
plot(rt1$diff, type='l',ylab= "Relative rMSPE",xlab="iter",
      main="Slower convergence due to rescaling of the whole row")
```



```r
# Plotting relative rMSPE  between consecutive iterations
rt1$data_Imp[c(1:3,8), ]                                         # Imputed data
```

```
##   total_screen_a20 total_read_a20 total_cogn_a20 total_phys_a20 total_sleep_a20
## 1         2.069573       2.018606      1.4069623       1.118996       14.658698
## 2         4.782267       1.607851      1.0296945       2.395961        8.217522
## 3         2.287969       1.517864      0.9879108       1.908223       14.566264
## 8         2.479044       1.367724      1.3249527       1.721729       15.226039
##   other_a20       age       bmi married abnorm educ_2 educ_3 educ_4
## 1  8.727165 0.8984977 0.8764558       1      0      0      0      1
## 2 11.966703 1.0020661 0.9113688       1      1      1      0      0
## 3  8.731770 0.9676729 1.2698394       1      1      0      1      0
## 8  7.880511 1.0973451 0.8244510       1      1      1      0      0
```

```r
rowSums(rt1$data_Imp[c(1:3,8), 1:6])                             # Checking row total on the imputed data
```

```
##  1  2  3  8
## 30 30 30 30
```

```r
rt1$Theta_est                                 # Estimated coefficients from  ALR model at convergence
```

```
##             total_screen_a20 total_read_a20 total_cogn_a20 total_phys_a20
## (Intercept)      -1.02819540    -1.83571623   -1.284346163    -1.25000358
## age              -0.77344959     0.30486491    0.178181390     0.06050260
## bmi               0.33843840    -0.28899490   -0.425415025    -0.29068827
## married           0.17011061     0.08006133   -0.147384829     0.05642967
## abnorm           -0.04462648     0.06525059   -0.148452176     0.11323046
## educ_2            0.31590673    -0.15712027   -0.047625710    -0.26745469
## educ_3           -0.11790762     0.01270909   -0.231159212    -0.12987199
## educ_4            0.01522290     0.15160532    0.006476687    -0.08584709
##              total_sleep_a20
## (Intercept)      0.32983428
## age              0.13476944
## bmi             -0.28079987
## married          0.26652066
## abnorm           0.17033999
## educ_2          -0.02446356
## educ_3          -0.02879685
## educ_4           0.16923961
```

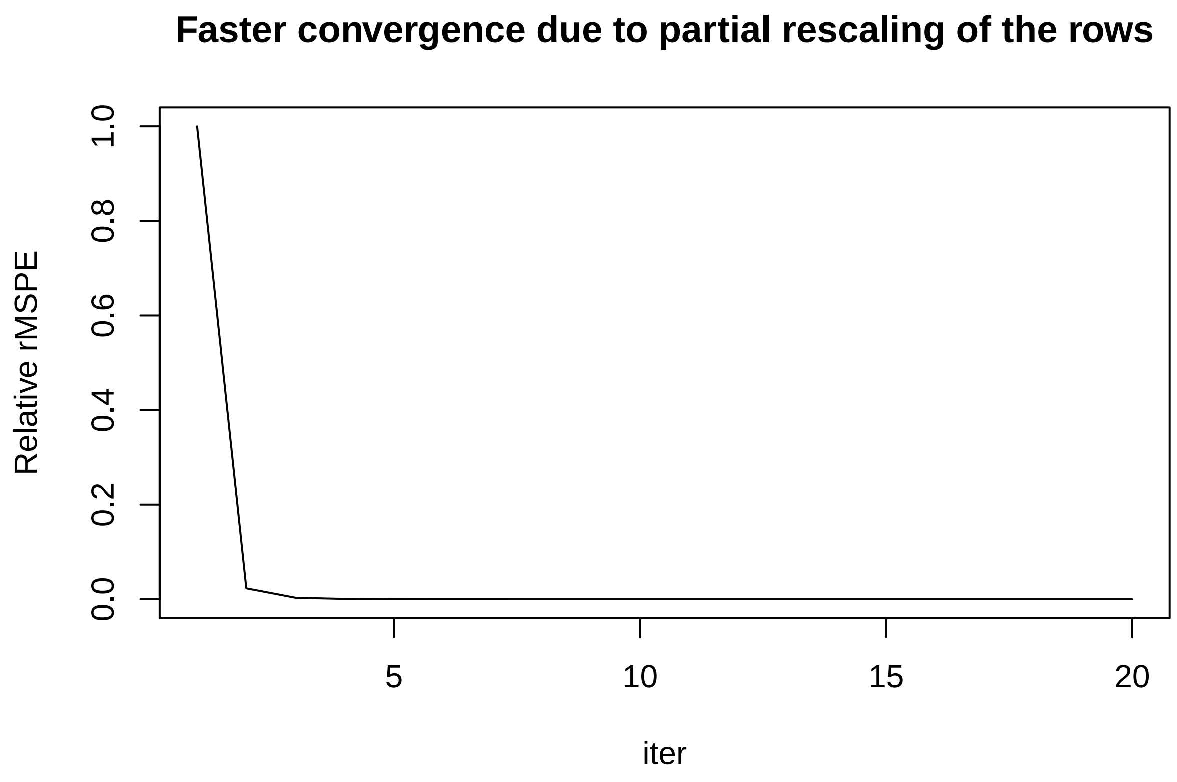
```
rt1$NArows       # Rows imputed
```

```
##  [1]  3  8  9 14 31 32 42 43 45 46 49 53 57 58 59 60 61 64 70 73 75 76 81 83 88
## [26] 92 93 94 95 97
```

`MICoDa_V2`: Imputation of missing compositions without re-calibrating the observed compositions

The following section of codes generates the imputed data, and plots the relative rMSPE where only missing components of the missing rows are imputed only.

```
# MICoDa_V2: When only missing components of the missing rows are imputed
rt2<- MICoDa(data_NA, 6, varlist,iter=20, rescaleAll=F,Constant=30)
plot(rt2$diff, type='l',ylab= "Relative rMSPE",xlab="iter",
     main="Faster convergence due to partial rescaling of the rows")
```



```r
# Plotting relative rMSPE  between consecutive iterations
rt2$data_Imp[c(1:3,8), ]                                          # Imputed data
```

```
##   total_screen_a20 total_read_a20 total_cogn_a20 total_phys_a20 total_sleep_a20
## 1         2.069573       2.018606       1.406962      1.1189963       14.658698
## 2         4.782267       1.607851       1.029694      2.3959613        8.217522
## 3         1.838219       5.390563       1.301679      3.8322814        7.326951
## 8         2.455197       1.325752       1.321434      0.9848221       16.266392
##   other_a20       age       bmi married abnorm educ_2 educ_3 educ_4
## 1  8.727165 0.8984977 0.8764558       1      0      0      0      1
## 2 11.966703 1.0020661 0.9113688       1      1      1      0      0
## 3 10.310307 0.9676729 1.2698394       1      1      0      1      0
## 8  7.646402 1.0973451 0.8244510       1      1      1      0      0
```

```r
rowSums(rt2$data_Imp[c(1:3,8), 1:6])                                 # Checking row total on the imputed dat
```

```
##  1  2  3  8
## 30 30 30 30
```

```r
rt2$Theta_est                              # Estimated coefficients from  ALR model at convergence
```

```
##             total_screen_a20 total_read_a20 total_cogn_a20 total_phys_a20
## (Intercept)      -1.20290166   -2.277224144   -1.446628234    -1.32922899
## age              -0.19125690    0.513670379    0.082970683    -0.03896016
## bmi               0.13188352    0.006991733   -0.242678662    -0.14786849
## married           0.04678506    0.059522980   -0.129639259     0.09146726
## abnorm           -0.06035211    0.101892302   -0.112019183     0.10658079
## educ_2            0.18158373   -0.205884718    0.041798215    -0.29211772
## educ_3           -0.21804932    0.088185017   -0.153327793    -0.10068312
## educ_4           -0.04305067    0.086223252    0.001870992    -0.06725119
##             total_sleep_a20
## (Intercept)      0.74232388
## age             -0.07985568
## bmi             -0.47423470
## married          0.14174814
## abnorm           0.13107070
## educ_2           0.17402434
## educ_3           0.07035728
## educ_4           0.31498211
```

```r
rt2$NArows       # Rows imputed
```

```
##  [1]  3  8  9 14 31 32 42 43 45 46 49 53 57 58 59 60 61 64 70 73 75 76 81 83 88
## [26] 92 93 94 95 97
```