SamplingStrata Modelling Anticipated Variance

Giulio Barcaroli

Created 26 Aug, 2018; Last Update 05 settembre, 2018

Handling Anticipated Variance

When optimizing the stratification of a sampling frame, it is assumed that the values of the target variables Y's are available for the generality of the units in the frame, and thanks to this assumption it is possible to estimate means and standard deviation of Y's in atomic strata.

Of course, this assumption does not hold very often. The situation in which some proxy variables are available in the frame is much more likely to happen.

In these situations, instead of directly indicating the real target variables, proxy ones are named as Y's. By so doing, there is no guarantee that the final stratification and allocation can ensure the compliance to the set of precision constraints.

Handling Anticipated Variance

In order to take into account this problem, and to limit the risk of overestimating the expected precision levels of the optimized solution, it is possible to carry out the optimization by considering, instead of the expected coefficients of variation related to proxy variables, the anticipated coefficients of variation (ACV) that depend on the model that is possile to fit on couples of real target variables and proxy ones.

In the current implementation, only models linking continuous variables can be considered.

Example with dataset 'Nations'

- Data on 207 countries related to demographic variables

```
data(nations)
head(nations)
```

```
##          Country  TFR contraception infant.mortality   GDP  region
## 1    Afghanistan 6.90            63              154  2848    Asia
## 2        Albania 2.60            47               32   863  Europe
## 3        Algeria 3.81            52               44  1531  Africa
## 4 American-Samoa 1.35            71               11  2433 Oceania
## 5        Andorra 1.61            71                7 19121  Europe
## 6         Angola 6.69            19              124   355  Africa
##   Continent
## 1         2
## 2         1
## 3         4
## 4         5
## 5         1
## 6         4
```

Example with dataset 'Nations'

Let us assume that in the sampling frame only variable **GDP** (Gross Domestic Product) is available for all countries, while **contraception rates** and **infant mortality rates** are available only on a subset of countries (about one third).

```
set.seed(1234)
nations_sample <- nations[sample(c(1:207),70),]
```

In this subset we can fit models between GDP and the two variables that we assume are the target of our survey.

Example with dataset 'Nations'

One model for **infant mortality** and **GDP**:

```
mod_logGDP_INFMORT <- lm(log(nations_sample$infant.mortality) ~ log(nations_sample$GDP))
summary(mod_logGDP_INFMORT)
```

```
##
## Call:
## lm(formula = log(nations_sample$infant.mortality) ~ log(nations_sample$GDP))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1292 -0.3765 -0.1455  0.3316  2.6345
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              6.86295    0.33620   20.41  < 2e-16 ***
## log(nations_sample$GDP) -0.46580    0.04389  -10.61 4.52e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6158 on 68 degrees of freedom
## Multiple R-squared:  0.6236, Adjusted R-squared:  0.6181
## F-statistic: 112.7 on 1 and 68 DF,  p-value: 4.523e-16
```

6/14

Example with dataset 'Nations'

and one model for **contraception** and **GDP**:

```
mod_logGDP_CONTRA <- lm(log(nations_sample$contraception) ~ log(nations_sample$GDP))
summary(mod_logGDP_CONTRA)
```

```
##
## Call:
## lm(formula = log(nations_sample$contraception) ~ log(nations_sample$GDP))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.96139 -0.27360 -0.01435  0.45058  1.25143
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.98318    0.30538   3.220  0.00197 **
## log(nations_sample$GDP)  0.34649    0.03986   8.692 1.22e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5593 on 68 degrees of freedom
## Multiple R-squared:  0.5263, Adjusted R-squared:  0.5193
## F-statistic: 75.55 on 1 and 68 DF,  p-value: 1.217e-12
```

Use of SamplingStrata

We define the *sampling frame* in this way:

```
nations$progr <- c(1:nrow(nations))
nations$dom <- 1
frame <- buildFrameDF(nations,
                      id="Country",
                      X="progr",
                      Y=c("GDP","GDP"),
                      domainvalue = "dom")
```

that is, we replicate twice the variable **GDP** because it will be used once for **infant mortality** and once for **contraception**.

We set 10% and 5% precision constraints on these two variables:

```
cv <- as.data.frame(list(DOM=rep("DOM1",1),
                         CV1=rep(0.10,1),
                         CV2=rep(0.05,1),
                         domainvalue=c(1:1)
                    ))
cv
```

```
##    DOM CV1  CV2 domainvalue
## 1 DOM1 0.1 0.05           1
```

Optimization without models

We build the strata without any assumption on the variability of the two target variables, and proceed in the optimization:

```
strata1 <- buildStrataDF(frame, progress = FALSE)
```

```
##
## Computations are being done on population data
##
## Number of strata:  207
## ... of which with only one unit:  207
```

```
solution1 <- optimizeStrata(cv,
                            strata1,
                            iter = 50,
                            suggestions = KmeansSolution(strata1,cv),
                            writeFiles = TRUE,
                            showPlot = FALSE)
```

```
##
## -----------------
##  Kmeans solution
## -----------------
##  *** Domain:  1  ***
##  Number of strata:  7
##  Sample size     :  17
##  *** Domain :  1   1
```

Optimization without models

Then, we evaluate the expected CV's on the three variables:

```
newstrata <- updateStrata(strata1,solution1)
framenew1 <- updateFrame(frame,newstrata)
framenew1 <- framenew1[order(framenew1$ID),]
framenew1$Y2 <- nations$infant.mortality
framenew1$Y3 <- nations$contraception
results1 <- evalSolution(framenew1, solution1$aggr_strata, 50, progress = FALSE)
results1$coeff_var
```

```
##          CV1       CV2       CV3  dom
## 1 0.05332022 0.2326833 0.1431669 DOM1
```

Clearly, the CV's on **infant mortality** and **contraception** are not compliant with the corresponding precision constraints.

Use of models in building strata

We proceed in building the **strata** dataframe using the models:

```
model <- NULL
model$beta[1] <- mod_logGDP_INFMORT$coefficients[2]
model$sig2[1] <- summary(mod_logGDP_INFMORT)$sigma
model$type[1] <- "loglinear"
model$gamma[1] <- 0
model$beta[2] <- mod_logGDP_CONTRA$coefficients[2]
model$sig2[2] <- summary(mod_logGDP_CONTRA)$sigma
model$type[2] <- "loglinear"
model$gamma[2] <- 0
model <- as.data.frame(model)
model
```

```
##         beta      sig2      type gamma
## 1 -0.4658038 0.6157600 loglinear     0
## 2  0.3464857 0.5593031 loglinear     0
```

```
strata2 <- buildStrataDF(frame, model = model, progress = FALSE)
```

```
##
## Computations are being done on population data
##
## Number of strata:  207
## ... of which with only one unit:  207
```

Optimization

We proceed with the optimization

```
strata2 <- buildStrataDF(frame, model = model, progress = FALSE)
```

```
##
## Computations are being done on population data
##
## Number of strata:  207
## ... of which with only one unit:  207
```

```
solution2 <-
  optimizeStrata(
    errors = cv ,
    strata = strata2,
    iter = 20,
    pops = 20,
    suggestions = KmeansSolution(strata2,cv),
    showPlot = TRUE,
    writeFiles = FALSE)
```

Solution

```
newstrata <- updateStrata(strata2,solution2)
framenew2 <- updateFrame(frame,newstrata)
framenew2 <- framenew2[order(framenew2$ID),]
framenew2$Y2 <- nations$infant.mortality
framenew2$Y3 <- nations$contraception
results2 <- evalSolution(framenew2, solution2$aggr_strata, 50, progress = FALSE)
results2$coeff_var
```

```
##            CV1        CV2        CV3  dom
## 1 0.005532387 0.05344947 0.02991657 DOM1
```

This time the expected CV's of all variables are more than compliant with the precision constraints.

Solution

We can decrease the sample size and increase the expected CV's by using the function *adjustSize*:

```
adjustedStrata <- adjustSize(size=45,solution2$aggr_strata)
```

```
##
##  47
##  47
##  Final adjusted size:  47
```

```
results2 <- evalSolution(framenew2, adjustedStrata, 200, progress=FALSE)
results2$coeff_var
```

```
##          CV1       CV2        CV3  dom
## 1 0.02812533 0.1014345 0.04823251 DOM1
```