conSTable Tutorial

Marco Garieri, Natalia Golini, Luca Pozzi

Consultants ESS Division, FAO

Modified: February 21th, 2014, Compiled: February 26, 2015

Abstract

conSTable is an R package designed to address the reconstruction of unreliable or missing values in partially observed tables using a sampling strategy. The package has been developed within the Statistical Division (ESS) at the Food and Agricultural Organization (FAO) of the United Nations to solve the problem of balancing the Food Balance Sheets (FBSs). FBS are real valued matrices with structural zeros, i.e. entries constrained to be zeroes, whose rows and columns have to satisfy a series of equality constraints. The cell values are only partially reliable due to measurement and reporting error. We will go through a real world example to illustrate the functionalities of the package

1 Introduction

A few initial remarks are important before getting started with the tutorial. This document is written for first-time users of the conSTable package. It is not a reference manual, nor does it contain the technical details. In [1] the sampling strategy for generating plausible (FBSs) is shown. Using of conSTable doesn't require proficiency in R. Nonetheless, some degree of familiarity with R will help with more complex modeling tasks. As pointed out in [1], the algorithm implemented in this package is developed for sampling plausible tables with limited information. This methodology can be applied to any balancing problem, e.g. balancing of Input Output tables, Social Accounting Matrix, etc. The functionalities of conSTable are shown in the contest of two-way tables. The real world example are the FBSs of the 169 countries belonging to Agricultural Market Information System (AMIS) for years 2007-2012 and following the sampling strategy proposed in [1]. A careful reading of the cited paper is strongly recommended.

The package is currently available at https://github.com/mrpozzi/conSTable/tree/develop and, it has been integrated into Statistical Working System (SWS) of FAO and is directly executable by FAO staff. conSTable requires R, version 3.1.2 or higher. R is an open source statistical software which can be downloaded at http://cran.r-project.org/. Another requirement is package msm. Future directions of development include the option of using different prior distribution (i.e., normal skewness), additional constraints on the columns and many others.

2 First Steps with *conSTable*

To install the package from github we need to load the package*devtools*

```
> if(!require(conSTable)){
+   library(devtools)
+   install_github("mrpozzi/constable",ref="develop")
+ }
```

As discussed in [1], the function requires three different files (a sample is included with the package).

```
> file <- system.file("extdata", "Adj.commodityContTab_db.csv",package="conSTable")
> file0 <- system.file("extdata", "Comm.SZ.csv",package="conSTable")
> filef <- system.file("extdata", "Adj.feedrange.csv",package="conSTable")
```

The first is a csv file containing prior information (expected value expressed in Kcal/cap/day and variability) given by experts (FAO staff) for the FBSs of the 169 countries belonging to Agricultural Market Information System (AMIS) in 2007-2012. This is a minimal requirement, which can accommodate the user's needs.

The header of the csv file is:

```
 [1] "AreaName"             "AreaCode"             "Group"
 [4] "Commodity"            "Year"                 "Production"
 [7] "Imports.primary"      "Imports.standardized" "Imports.total"
[10] "Exports.primary"      "Exports.standardized" "Exports.total"
[13] "Domestic.supply"      "Imports.sd"           "Exports.sd"
[16] "Industrial.use"       "Losses"               "Seed.use"
[19] "Feed.use"             "Food.use"             "Stock.change"
```

For these FBSs, the experts (FAO staff) have assumed that the only reliable information (consolidated term) comes from "Production" and that the balance identity to be respected is the following:

$$\begin{aligned}\text{Production}_{i,c,t} = -\text{Total Imports}_{i,c,t} + \text{Total Exports}_{i,c,t} + \text{Feed}\\ \text{Seed}_{i,c,t} + \text{Losses}_{i,c,t} + \text{Industrial}_{i,c,t} - \text{Stock}_{i,c,t}\end{aligned}$$

for each commodity i in a given country c during the period t.

Because for each commodity, the expected value for Stock was obtained as difference between Production and the remaining terms, all the FBSs are balanced in input. See Section 2.2 in [1] and the references cited therein to understand how prior information were generated. The sd variables in the dataset represent the variability of the respective totals, expressed in terms of standard deviation.

The second file cointains data for the structural zeroes, i.e. values constrained to be zero for specified commodity for some terms of the balance identity (1). These values are not country-variant and year-variant.

```
            Group                   Commodity Production Imports.total Stock.change
1 Butter and ghee             butter.and.ghee        num           num          num
2         Cereals         barley.and.products        num           num          num
3         Cereals cereals..other.and.products        num           num          num
4         Cereals          maize.and.products        num           num          num
5         Cereals         millet.and.products        num           num          num
6         Cereals           oats.and.products        num           num          num
  Exports.total Food.use Feed.use Seed.use Losses Industrial.use
1           num      num                      num
2           num      num      num      num    num            num
3           num      num      num      num    num            num
4           num      num      num      num    num            num
5           num      num      num      num    num            num
6           num      num      num      num    num            num
```

Structural zeros data are indicated by blank cells.

The third file contains the expected value of the total column of Feed and the respective lower and upper bound, for each country and year.

```
  AreaCode AreaName Year  FBS.mult EDemand_p EDemand_lb EDemand_ub
1        1  Armenia 2007 0.8529358  635.6697   549.2550   721.5066
2        1  Armenia 2008 0.8923241  612.4662   526.8688   699.5743
3        1  Armenia 2009 0.9262565  669.1666   578.8687   756.4770
4        1  Armenia 2010 0.8814851  664.5670   577.8289   748.0474
5        1  Armenia 2011 0.9054073  630.9346   545.9832   713.7095
6        1  Armenia 2012 0.8628133  656.7158   567.4724   743.4872
```

These data represent constraint column total of Feed coming from subject matter knowledge. Lower and upper bounds delimit the range of values within which the sampled estimate of total column of Feed.use must fall. Information on FBS.mult can be ignored.

Now we can read the data, the package has a function for this task and for formatting the data properly

```
> FBS <- readFBS(file,file0,filef)
```

`FBS` is a nested list indexed by country and year, where:

- `FBS[[country]][[year]]$data` is a commodities by usage matrix with the expected value of the commodity allocated to the specific usage (e.g. Food, Feed, etc..).
- `FBS[[country]][[year]]$feed` is a vector with the lower and upper bounds on Feed.
- `FBS[[country]][[year]]$sd` is a matrix with the variabilities for each value of `FBS[[country]][[year]]$data`
- `FBS[[country]][[year]]$row_Tot` is a vector with the Production of each commodity. This represents a constraint on the total of the row (commodity).

3 Main Functionalities of the Package

The package include functions to balance:

1. A single country for a single year: `balanceFBS`
2. A single country for all the years provided in input: `balanceCountry`
3. all countries in input for all the years: `balanceAll`

3.1 Balancing a Country for a given Year

The function for balancing a single combination CountryYear exploits the principle of function closure to create a function which encapsulates the data and can then be re-used multiple times with different options:

```
> balanceOne <- balanceFBS(FBS)
> oneTab <- balanceOne("9",2012,oset=c(30,30,30,30,30,10000),
+                      nIter=100,verbose=FALSE,checks="none",feedShift=30)
```

Where the input is:

- `sanityCheck`: if `TRUE` the algorithm checks, in the input stage, if the total of Feed in the table `FBS[[country]][[year]]$data` falls in the boundaries given by `FBS[[country]][[year]]$feed`.
- `country`: the country of interest.
- `year`: the year of interest.
- `oset`: a vector of the same length of the number of columns. It defines the variability for each column in absolute terms of the expected values for each column. If `oset=NULL` then the variability of each commodity in each column will be assumed to be 20% of the total of the respective column. See formula (5) in Section 4 in [1] for more details. At the moment, StockVar variability is not relevant, since StockVar is calculated, for each commodity, as residual.
- `prop`: must be a number between 0 and 1 or a vector of length 6 with values in the same range. It expresses the variability on a cell value in terms of a percentage of such value. To be used in an analogous way as `oset`. Either `prop` or `oset` have to be defined.
- `n.iter` is the number of iterations of the sampler. The function will generate a single FBS per iteration and the best will be chosen using the objective function and the constraints.

- `verbose`: if `TRUE` progess messages are printed to screen.
- `writeTab`: if `TRUE` the optimal balanced FBS is written in a csv file in the working directory.
- `checks`: if `"all"` the algorithm checks the following conditions:
  1. Exports < Production + Imports
  2. Stock < 20% of Domestic Supply

  If `"none"` the above checks will not stop the algorithm but will throw warnings.

At the time, the algorithm does not stop if the previous conditions are not respected but returns warnings. The algorithm stops if:

1. Production has negative values in `FBS[[country]][[year]]$row_Tot`
2. Imports has positive values in `FBS[[country]][[year]]$data`, because it can only take negative values in the balance identity (1);
3. Exports has negative values in `FBS[[country]][[year]]$data`, because it can only take positive values in the balance identity (1);
4. null rows have not null row sums.

The function `balanceOne` returns an object with the following fields:

- `bestTab`: the table with the best value of the objective funtion.
- `tables`: list of the all sampled balanced FBSs.
- `iters`: number of iterations that corresponds to the number of sampled balanced FBSs.
- `objective`: value of the objective function for the best table.

We recall that, at the moment, the algorithm requires the input table to be balanced in input. The reason is that, with rows difficult to be sampled, after several attempts it gives the input rows in output.
If the optimal table, after applying the objective function, does not respect the feed ranges given by expert, the algorithm will start again, chaning the oset parameter for the feed column, taking half of the previous attempt. This process will be recursive until the optimal table respect the feed constrains.

3.2 Balancing a Country for every Year

To balance a country for all the years available, type the following

```
> countryTab <- balanceCountry(FBS,"Congo",oset=c(30,30,40,50,50,1e4), nIter=10,verbose=FALSE)
```

This function has the same inputs as `balanceFBS` and provides a list in output with one of the objects described above for every year.

This function implements one further constraint that links the optimal FBS to the one of the year before. The table to be chosen is the one that respects the constraint

$$|\text{Tot. Food}_{t+1} - \text{Tot. Food}_t| \leq 150\text{Kcal/cap/day}$$

If this constraint can not be satisfied by the sampled tables, then the algorithm will choose the table with the best value of the objective function that minimizes $|\text{Tot. Food}_{t+1} - \text{Tot. Food}_t|$.

3.3 Balancing every Country for every Year

To balance all the years and countries in the data we use a command like

```
> allTab <- balanceAll(FBS,oset=c(30,30,40,50,50,10000),feedShift=20,verbose=FALSE)
```

4 An Example: Italy in 2011

Let's now look at the data for Italy for year 2011. Note how the user can see the list o available countries

```
> head(attr(FBS,"countryMap"))
```

```
    Armenia Afghanistan     Albania     Algeria      Angola   Argentina
        "1"         "2"         "3"         "4"         "7"         "9"
```

and the list of years for that country as

```
> head(names(FBS[[attr(FBS,"countryMap")[["Italy"]]]]))
```

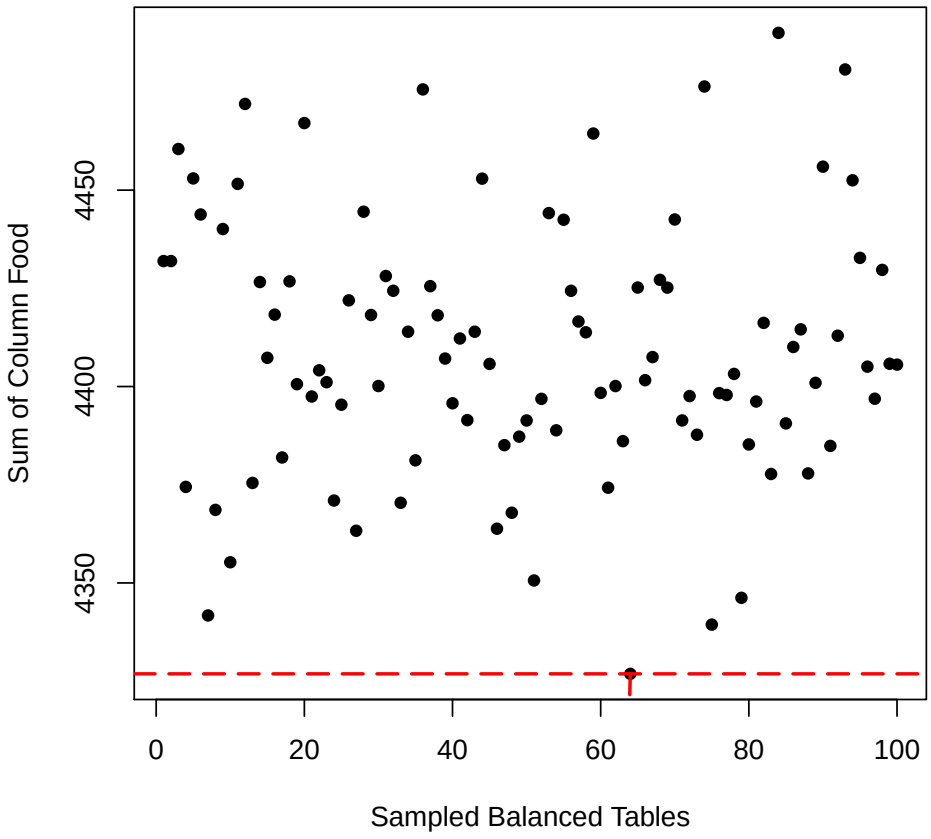
```
[1] "2007" "2008" "2009" "2010" "2011" "2012"
```

Let's run the algorithm for that combination

```
> balanceOne<-balanceFBS(FBS)
> set.seed(181282)
> output<-balanceOne("Italy",2011,oset=c(30,30,40,50,50,10000),nIter = 100,verbose=FALSE,writeTab=FALSE,
```

In this case we sample 100 balanced tables. Because Italy is considered saturated country in term of Dietary Energy Supply (i.e. 3539 Kcal/Cap/Day) by experts, the "optimal balance table" is the one has the lowest total of the Food column.

We can plot the objective function value for each of the sampled tables The best table is accessible at `output$bestTab` and written to the csv `"Italy2011.csv"` if you run the algorithm with option `writeTab=TRUE`


Sum of Column Food
4350
4400
4450
0
20
40
60
80
100
Sampled Balanced Tables

References

[1] Garieri M., Golini, N. and Pozzi L. *Placeholder*. (2015).

SessionInfo

- R version 3.1.2 (2014-10-31), `x86_64-apple-darwin10.8.0`
- Locale: `C/C/C/C/C/en_US.UTF-8`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: conSTable 0.1, msm 1.5
- Loaded via a namespace (and not attached): Matrix 1.1-4, expm 0.99-1.1, grid 3.1.2, lattice 0.20-29, mvtnorm 1.0-2, parallel 3.1.2, splines 3.1.2, survival 2.37-7, tools 3.1.2