Parameterisation of Food Balance Sheet Uncertainty Distribution

Michael. C. J. Kao
Food and Agriculture Organization
of the United Nations

Abstract

In this paper, we briefly illustrate the use of the **faoswsFlag** package to parameterise the uncertainty distribution based on the confidence about a value.

Keywords: Uncertainty, Distribution.

1. Introduction

In preparing the Food Balance Sheet (FBS), one of the most indispenable yet challenging operation is the balancing mechanism. Due to the imperfection of data collection, estimation and imputation in the real world, it is the norm that the FBS is unbalanced and does not satisfy the equality constraint at first sight. Thus, a balancing mechanism is essential for satisfying the equality between the demand and supply of the FBS.

In current practice, when imbalance exist, a variable is assigned the balancing item and the value of the balancing item is adjusted such that the equality constraint is satisfied. The choice of a variable as a balancing item often reflects the availability of data (or the lack of data), rather than a logical justification and empirical evidence. It is therefore not surprising that different SUA compilers/SUA approaches have chosen different variables as their balancing items. USDA's balances, for instance, use feed (and residual use) as the balancing item, while the FBS often use food to balance supply and demand. Conveniently, the XCBS approach often chooses whatever variable is not explicitly available. Clearly, none of these approaches renders a satisfactory solution to the problem and, no matter what variable is used as the balancing item, this variable is fraught with the measurement errors of all other variables. Given the fact that there is no a priori reason to assume that the measurement errors cancel out, the balancing item is bound to be the most inaccurate variable of the balance. Extending the logic to the Food Balance Sheets, using food as the balancing item would therefore be the least suitable solution.

Problems associated with the current approach motivated the research team to seek a method in which inequality occuring in the FBS to be allocated to various elements based on a sound and logical reasoning rather than arbitrary allocation. One method to handle the problem is to balance the FBS based on a probabilistic basis. Each element is assumed to have a pre-determined level of uncertainty, and the allocation of the imbalance will depends on the uncertainty of each element. That is, the smaller the uncertainty we have with a particular element, the less we should apportion the imbalance or adjust that particular element. In the extreme case where we have perfect certainty about an element, than no imbalance should be added to the element nor adjusted.

In order to proceed with the probabilistic balancing mechanism, formulation of distribution for each of the elements in the FBS is necessary. The specification of the distributions undermines the validity of the balancing mechanism, and thus a consistent and logical construction of the distribution is crucial. The distributions should reflect the underlying uncertainty associated with each elements while respecting the relationship amongst all elements. It is under these conditions, the optimal solution from the probabilistic framework is valid and justified. A framework guiding the specification of the distributions and the parameterisations is thus required and is the focus of this paper. The absence of such framework generates inconsistency and paradox, further the use of the term probability maximisation is a disguise for a procedure which does not bear any proper interpretation.

The paper is organised as follow. A simplified dataset is presented to familiarise the user with the problem. The subsequent section is then devoted to the rationale and theory behind the method for constructing the uncertainty distribution; a simple example with snippets will accompany the theory section to elaborate on the methodology. Finally, an application of the method for the balancing mechanism is presented and end with discussion.

2. The Data

Before we delve into the theory and the descriptions, we will demonstrate some cases of the data for back ground. Further, these data will be later used to demonstrate the method and the package.

First of all, we can load up the package by prompting the following command.

```r
library(faoswsFlag)
```

An example flag table is provided along with the package.

flagObservationStatus	flagObservationWeights
	1.00
I	0.60
E	0.20

Empty flag denotes official data, while "I" stands for imputed value and "E" are manual estimates.

Shown below is an example of the format the data. Each observed value is associated with a flag which indicate the source of the data. Under the proposing framework, the flag contains information about the uncertainty of the value and will be used to parameterise the distribution. There are more elements to the Food Balance Sheet, however, we have selected a handful of variables for illustrative purpose.

production	flag	import	flag	export	flag	food	flag	loss	flag
220	I	10		50		150	E	100	I

3. The Methodology

In this section, we will describe the rationale and provide some background theory followed by an example at the end.

From the example, we can observe that for each element and item in the Food Balance Sheet (FBS), only a single value is observed. Since only a single observation is available, the use of Frequentism method can not be applied here. Rather, we have adopted the subjectivism interpretation of probability in order to come up with a solution.

To construct a probability distribution about the value, one first requires a chosen particular distribution, then parameterise the distribution according to a set of standards and rules.

The choice of the distribution should reflect knowledge and known constaints about the variable. The support, shape and properties of the distribution should be guided by the expertise of the officer. For example, production is strictly positive and thus distributions such as the Normal or the Cauchy should be eliminated from the set. Further, if extreme value are more likely then a log-normal distribution may be more preferable in comparison to the truncated normal distribution which has a higher kurtosis.

After the distribution has chosen, then the distribution need to be parameterised in order to complete the construction of the distribution. Logically, the parameter of the distribution should relate to the observed value and the specified confidence. Moreover, the mode of the constructed distribution should be made to be equivalent to the observed value. This task is in general simple, yet the conversion of the confidence level to the dispersion parameter of the distribution requires several more steps.

Here we propose a method in which the confidence can be converted to a measure of uncertainty we have about a particular value and ultimately lead to the parameterisation of distributions.

3.1. Quantifying Uncertainty

To measure a piece of information, one can use the formula of *self information* which is a measure of the information content and is defined as,

$$I = -ln(P) \qquad P \in [0,1] \tag{1}$$

Where P is the probability or the confidence about the accuracy of the value assigned to the observed value in the first place. The natural logarithm is adopted here, but logarithm of any base can be used. This is a measure of the uncertainty conditioned on the confidence we have provided about the observed value. The greater the I, the larger the associated uncertainty, that is, the lower the confidence we assign to the particular value, the higher the uncertainty. When the confidence is 0, or with 0% certainty, then I is infinite or infinite uncertainty; on the other hand, when the confidence is 1 then the the value is known with certain.

The logarithm also ensures that the uncertainty is additive. Essentially, the sum of the uncertainty is the log of the products of the probabilities assigned to the values. That is, it is the log of the joint probability assuming independence.

The function enable us to convert the confidence about a single value to how much uncertainty is associated with the value.

3.2. Parameterise Distribution Given Uncertainty

Provided that we observe a single value, and at the same time our quantifying the uncertainty about a particular value; any chosen distribution can be parameterised accordingly to reflect the empirical evidence and knowledge about the value.

By setting the observed value to the expected value of the distribution (the expected value here refers to the value with the highest probability, that is, the mode) and the self information to the expected information or the differential entropy of the distribution, the parameters of the chosen distribution can then be obtained by solving the set of equations. Given the level of uncertainty associated with each element, then regardless of the choice of distributions, one can always parametrize the distribution where the uncertainty is held the same. This provides a consistent framework for specifying distributions in which the uncertainty for each element is consistent and relative amongst all elements.

That is, we parameterise the distribution given the following identity.

$$\begin{aligned} Mode(X) &= x \\ H(X) &= I \end{aligned} \tag{2}$$

Where X is the random variable and x is the observed value, I is the self-information or uncertainty computed according to formula 1, and H is the differential entropy of the chosen distribution.

The main reason to use the entropy rather than other dispersion parameters such as the aboslute size of the standard deviation is because it is unit free and does not depends on the size of the value. If we were to impose uncertainty between two values, then the uncertainty associated with both value should be set respectively to the confidence given independent of the magnitude of the value. For example, if we we have observed 2000 tonnes of wheat production and 1000 tonnes of food while the confidence in the two value are identical, then the balance should be 1500 tonnes of production and food. If we were to base the uncertainty on standard deviation or percentage of variation, then the larger value will have large standard deviation of variation based on percentage and thus the final value will be closer to 1000 even though we have equal confidence in both values.

Furthermore, both Normal and the truncated Normal distribution has a standard deviation parameter, however, setting the two distribution with identical standard deviation actually gives the normal distribution a high level of uncertainty.

3.3. A Simple Example

The following illustration provides an example of the method, along with codes to further explain the method. In addition, we will demonstrate how this framework can provide consistent parameterization of various distribution while maintaining the same level of uncertainty with the value.

```
obsValue = 2
confidence = 0.2
```

Let us assume that we have an estimated value of 2 thousand tonnes of wheat production in Australia in 2010.

Then following the flag table, we have a confidence of 20% in the observed value. The amount of uncertainty regarding the wheat production in Australia given the confidence can then be calculated as,

$$I = -ln(0.2) \approx 1.6094379$$

```
(selfInfo = selfInformation(confidence))


## [1] 1.609438
```

In order to preserve this uncertainty associated with this piece of information, we need to preserve the entropy of the distributions. That is, regardless which distribution we choose we need to parameterise the distribution such that the entropy is equivalent to the same nat of information available.

Now for naive reasons that we want to impose a Normal distribution on the wheat production in Australia, we can first set the observed value to the mode of the distribution to first give us the first parameter of the Normal distribution.

$$\mu = 2 \tag{3}$$

In order to solve for the standard deivation σ of the Normal distribution, we first re-arrange the entropy function of the Normal distribution where the parameter σ is a function of the entropy H. Then by substituting the expected information H with the self-information of the observed value I, we can then obtain the standard deviation of the distribution

Starting with the differential entropy of the Normal distribution

$$H = \frac{1}{2}ln(2\pi e\sigma^2)$$

and re-arrange the equation,

$$\sigma = \sqrt{\frac{e^{2H}}{2\pi e}}$$

substituting H with I we obtain the value of the standard deviation as

$$\sigma = \sqrt{\frac{e^{2(-ln(0.2))}}{2\pi e}} \approx 1.2099$$

or simply,

```
parameterise(obsValue = obsValue,
             selfInformation = selfInfo,
             distribution = "normal")


## $mean
## [1] 2
##
## $sd
## [1] 1.209854
```

That is, when the observed value of wheat production in Australia is 2 and a confidence of 20% is imposed, then the associated uncertainty distribution is then:

$$W \sim N(2, 1.2099)$$

However, if we believe that the production is in general rather stable over time but are subject to events such as drought that can create extreme values, then the Cauchy distribution may be a more reasonable distribution. The same method also allow us to parameterise the Cauchy distribution in which the uncertainty remains constant. The choice of distribution should reflect our belief in the probability allocation but it should not alter the uncertainty we have imposed initially.

Following the same procedure, we obtain the following parameter.

$$\begin{aligned} x_0 &= 2 \\ \gamma &= e^{I-ln(4\pi)} = e^{-ln(0.2)-ln(4\pi)} \approx 0.3979 \end{aligned}$$

then,

$$W \sim Cauchy(2, 0.3979)$$

```
parameterise(obsValue = obsValue,
             selfInformation = selfInfo,
             distribution = "cauchy")


## $location
## [1] 2
##
## $scale
## [1] 0.3978874
```

Note, since the standard deviation of the Cauchy distribution is undefined and thus it is impossible to parameterise the distribution if we based our uncertainty measure on the size of the standard deviation.

Moreover, we know production can not be negative and thus distributions such as the Normal or the Cauchy distribution with unbounded support may not be the appropriate distribution. A truncated Normal distribution may incorporate this information by truncating support and allow the variable to be defined strictly on the positive real line.

Following the principle, we will arrive at a distribution which preserves the uncertainty yet re-assign the probability to reflect the physical condition that production can not be negative. In the case of the truncated Normal distribution, analytical solution does not exist, but a numerical solution is provided by the package.

```
parameterise(obsValue = obsValue,
             selfInformation = selfInfo,
             distribution = "truncNorm")


## $mean
## [1] 2
##
## $sd
## [1] 1.502336
```

and the resulting distribution is,

$$W \sim trN(2, 1.5023)$$

When the mean is close to zero, the standard deviation of the truncated normal is marginally larger than the normal distribution above. This is due to the fact to maintain the same level of uncertainty while reducing the support space, one has to increase the standard deviation. However, as the mean increases, the truncated normal becomes more like the normal distribution with very similar standard deviation.

Finally, the log-Normal distributnion is also another distribution which is defined only on the real line that is suitable to describe the probabiblity allocation of the wheat production.

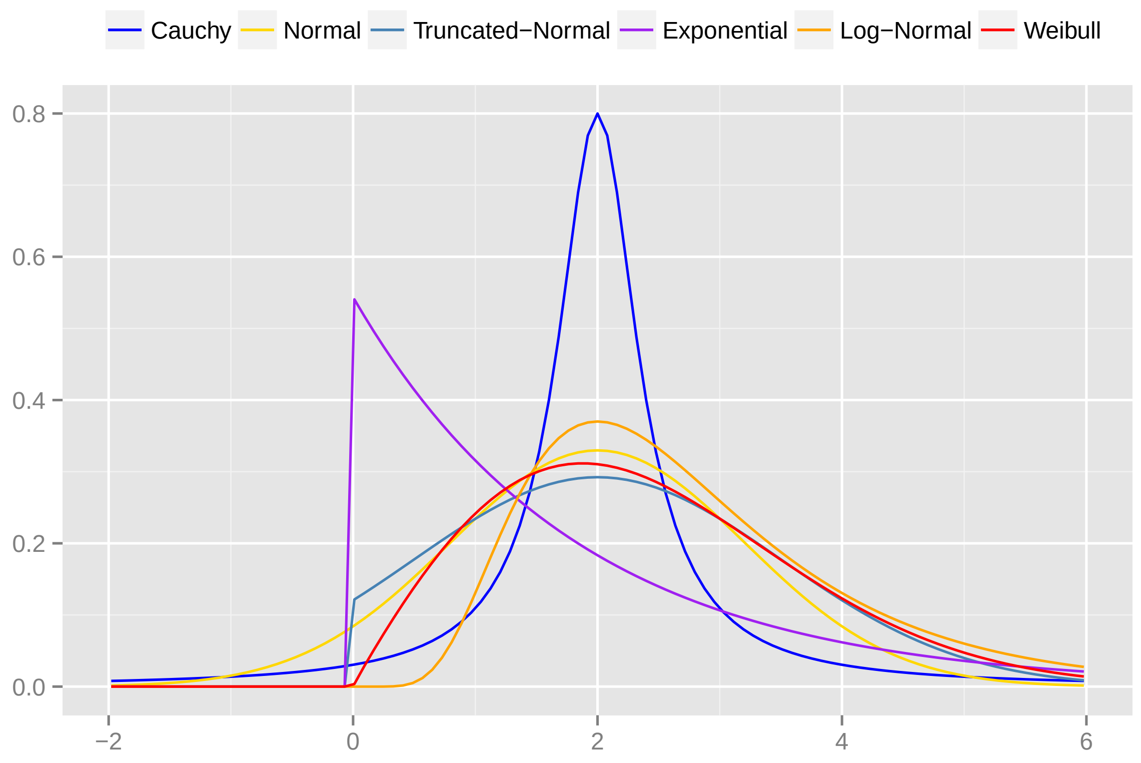
```
parameterise(obsValue = obsValue,
             selfInformation = selfInfo,
             distribution = "logNorm")


## $meanlog
## [1] 0.9238271
##
## $sdlog
## [1] 0.4802915
```

and,

$$W \sim lnN(0.9238, 0.4803)$$

Here is a comparison of the constructed distributions.



4. Illustration

Take the example data, the first step is to construct the uncertainty of each FBS element based on the flags in the FBS.

```
## Select all the flag columns
flagColumns = grep("Flag", colnames(exampleFBS.df), value = TRUE)
valueColumn = grep("Value", colnames(exampleFBS.df), value = TRUE)

## First we convert the flags to weights or confidence
(weightsFBS.df = data.frame(lapply(exampleFBS.df[, flagColumns], flag2weight)))


##   productionFlag importFlag exportFlag foodFlag lossFlag
## 1            0.5          1          1     0.75      0.5


## Then we compute the self-information
(selfInfoFBS.df = data.frame(lapply(weightsFBS.df, selfInformation)))


##   productionFlag importFlag exportFlag  foodFlag  lossFlag
## 1      0.6931472          0          0 0.2876821 0.6931472


## Then we sum up the self-information for each element to obtain
```

```r
## the uncertainty of each element.
totalInfo.df = data.frame(lapply(selfInfoFBS.df, sum))
```

To create the distribution, we simply prodive the function ‘distributionise‘ the observered value, the total information computed from the flag andthe desired distribution.

The function returns a list of two object. The first is the distribution function with the parameters computed, while the second object is a list with the corresponding values of the parameter.

```r
## Parameterise the production element with a Normal distribution
distributionise(obsValue = exampleFBS.df$productionValue,
                selfInformation = totalInfo.df$productionFlag,
                distribution = "normal")


## $pdf
## function (x)
## dnorm(x, mean = mean, sd = sd)
## <environment: 0x44cf450>
##
## $parameters
## $parameters$mean
## [1] 220
##
## $parameters$sd
## [1] 0.4839414


## Parameterise the production element with a Truncated Normal distribution
distributionise(obsValue = exampleFBS.df$productionValue,
                selfInformation = totalInfo.df$productionFlag,
                distribution = "truncNorm")


## $pdf
## function (x)
## dtruncnorm(x, a = 0, b = Inf, mean = mean, sd = sd)
## <environment: 0x42214b0>
##
## $parameters
## $parameters$mean
## [1] 220
##
## $parameters$sd
## [1] 0.4839332
```

Below we show a full process of how to construct each uncertainty distribution and specify the constraints for the balancing of the FBS.

```r
productionDist =
    distributionise(obsValue = exampleFBS.df$productionValue,
                    selfInformation = totalInfo.df$productionFlag,
                    distribution = "truncNorm")


importDist =
    distributionise(obsValue = exampleFBS.df$importValue,
                    selfInformation = totalInfo.df$importFlag,
                    distribution = "truncNorm")


exportDist =
    distributionise(obsValue = exampleFBS.df$exportValue,
                    selfInformation = totalInfo.df$exportFlag,
                    distribution = "truncNorm")


foodDist =
    distributionise(obsValue = exampleFBS.df$foodValue,
                    selfInformation = totalInfo.df$foodFlag,
                    distribution = "truncNorm")
lossDist =
    distributionise(obsValue = exampleFBS.df$lossValue,
                    selfInformation = totalInfo.df$lossFlag,
                    distribution = "truncNorm")


## Create the likelihood function from the distributions
ll = function(x){
    -log(productionDist$pdf(x[1])) -
        log(importDist$pdf(x[2])) -
        log(exportDist$pdf(x[3])) -
        log(foodDist$pdf(x[4])) -
        log(lossDist$pdf(x[5]))
}

## Create the contraint funciton
constraint = function(x){
    ## (1) Productin + Import - Export - Food - Loss = 0
    ## (2) and (3) holding import and export constant
    c(x[1] + x[2] - x[3] - x[4] - x[5], x[2], x[3])
}

## Balance the Food Balance Sheet
library(Rsolnp)
## Double check this solution, the resulting likelihood is infinite
balancedFBS =
    solnp(pars = as.numeric(exampleFBS.df[, valueColumn]),
          fun = ll,
          eqfun = constraint,
          eqB = c(0, exampleFBS.df$importValue, exampleFBS.df$exportValue),
          LB = rep(0, length(valueColumn)))


##
## Iter: 1 fn: 1e+24  Pars:  261.87887  10.00000  50.00000 130.53152  91.34734
## Iter: 2 fn: 1e+24  Pars:  261.87886  10.00000  50.00000 130.53152  91.34734
## solnp--> Completed in 2 iterations


balancedFBS$par


## [1] 261.87886  10.00000  50.00000 130.53152  91.34734
```

5. Discussion

Affiliation:
Michael. C. J. Kao
Economics and Social Statistics Division (ESS)
Economic and Social Development Department (ES)
Food and Agriculture Organization of the United Nations (FAO)
Viale delle Terme di Caracalla 00153 Rome, Italy
E-mail: michael.kao@fao.org
URL: https://github.com/mkao006/sws_r_api/tree/master/faoswsFlag