'agreement': Analyse the agreement between two measurement methods

by Fabio Frascati, Elia Biganzoli and Bruno Mario Cesana

Assessing agreement between two measurement methods (Y, X) is not an immediate statistical approach. Among the several approaches, *Altman* and *Bland* method (Altman and Bland, 1983) is prone to subjective interpretation leading to not appropriate conclusions.

The Concordance Correlation Coefficient (CCC) has been proposed by *Lin* (Lin, 1989) as an objective agreement index and it is obtained by the product of the Precision (*Pearson*'s correlation coefficient, ρ_{YX}) and *Accuracy* ($C_b = 2\sigma_X\sigma_Y \,/\, [\sigma_Y^2 + \sigma_X^2 + (\mu_X - \mu_Y)^2]$) as $\rho_{YX} \cdot C_b$. Furthermore, a formal agreement decision approach has been proposed by Lin et al. (2002), in the context of a bivariate Gaussian (Y, X) distribution, by using CCC and several other indices such as *Precision*, *Accuracy*, *Total Deviation Index* (TDI) and *Coverage Probability* (CP) of different agreement boundary.

It is worthwhile to stress that in this case the role of the null (H_0) and alternative (H_1) hypotheses has to be reversed: H_0 is of no agreement and H_1 is of agreement. So, the testing procedure is usually based on the confidence interval approach, such as in the equivalence studies and in the non-inferiority controlled clinical trials models.

The aim of this paper is to illustrate the **agreement** package (in **R** language) that allows to calculate the agreement indices, proposed by Lin et al. (2002), and their transformed values for obtaining a better approximation to the Gaussian distribution. Furthermore, and as a more relevant tool, this package investigates their asymptotic properties according to a simulation study under the bivariate Gaussian model. So, having fixed the five parameters of the model (μ_Y, μ_X, σ_Y^2, σ_X^2, ρ_{YX} or σ_{YX}) under the null (H_0) or the alternative (H_1) hypothesis it is possible to investigate, for example, the influence of different sample sizes on the asymptotic properties of the agreement considered indices. In addition, this allows to obtain the different proportions of rejection, given the sample sizes.

In the next section we will introduce the context of the topic in this article. In the section *Analytical Expressions* we will explain analytical formulae when the target values are random (measured with error) while in the sections *The features of lin.simulation()* and *How-to* we will give some details of the current *release* of **agreement** and we will provide an example.

Analytical Expressions

agreement package implements the function `lin.simulation()` which performs:

(a) the simulation study

(b) the calculation of the five agreement indices proposed by Lin et al. (2002) together their approximately Gaussian distributed transformations:

- Precision (Z inverse hyperbolic tangent transformation)
- Accuracy (logit transformation)
- CCC (Z inverse hyperbolic tangent transformation)
- TDI (W logarithmic transformation for mean square error MSD, not directly TDI)
- CPk (logit transformation)

(c) the calculation of their theoretical values under H_0 and under H_1

(d) the rejection proportions of H_0 (upper or lower unilateral confidence interval above or under the pertinent agreement threshold, respectively)

(e) produces a summary of the simulation results

This paper shows the results for CCC and its Z transformation for sake of simplicity. One can easily find the same pattern for the other agreement indices. Particularly, if the lower unilateral $(1-\alpha)\cdot 100\%$ confidence limit of the Z transformed CCC value is greater than its theoretical value under H0 we can conclude for the agreement:

$$Z \geq \zeta_0 + \Phi^{-1}(1-\alpha)\,\sigma_{Z_0} \tag{1}$$

where ζ_0 and σ_{Z_0} are the asymptotic expected value and standard deviation of the **Z** transformation of **CCC** under H_0. It is a one tail test with a right region of rejection. It is clear that we have to anti-transform $\zeta_0 + \Phi^{-1}(1-\alpha)\,\sigma_{Z_0}$ to obtain the threshold in terms of **CCC**. The asymptotic power of accepting agreement by using **Z** is:

$$P_Z = \Phi\left[\frac{\zeta_0 - \zeta_1 + \Phi^{-1}(1-\alpha)\ \sigma_{Z_0}}{\sigma_{Z_1}}\right] \tag{2}$$

The features of lin.simulation()

We illustrate some of the capabilities of the **agreement** package using the `lin.simulation()` function. Seven inputs are available for this command:

- `NUM_CAMP` number of samples to simulate. Its default value is 5000.
- `NUM` sample size. Its default value is 30.
- `matH0` matrix of parameters under null hypothesis (H_0). It has 2 rows and 3 columns:

$$\begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \mu_x \\ \sigma_{xy} & \sigma_y^2 & \mu_y \end{pmatrix}$$

- `matH1` matrix of parameters under alternative hypothesis (H_1). It has exactly the same structure of `matH0`.
- `underH0` logical parameter to determine what condition to simulate. Its default value is `TRUE` (simulation under H_0).
- `ALPHA_CI` leading to a $1-\alpha$ confidence level with a unilateral confidence interval. Its default value is 0.05
- `la_CP1` the threshold used for TDI. Its default value is 0.9

The funcion `lin.simulation()` has a list of eight objects as output (see Figure 1):

- `table` it is a matrix. Each row represents a measure of agreement and each column a summary of the simulation.
- `underH0` see above
- `matH0` see above
- `matH1` see above
- `NUM_CAMP` see above
- `NUM` see above
- `alpha` see above
- `rho` is the value for the correlation coefficient under H_0 (if `underH0 = TRUE`) or H_1 (if `underH0 = FALSE`)

`table` is the most important; its columns are:

- **Th val** theoretical value of each agreement measurement
- **Thr** inverse transformation of the threshold for each transformation
- **Th prob** theoretical value of probability. It should be equal to α
- **Mean of est** antitransformation of the mean estimate of the transformation of each measure of agreement
- **Std of est** standard deviation of the transformation of each measure of agreement
- **Mean of std** mean of the estimates of the standard deviation of each transformation
- **Prop rej** the proportion of samples which lie in the rejection region under H_0

How-to

We must first select a single simulation case. We can choose between null hypothesis (`underH0 = TRUE`) or the alternative (`underH0 = FALSE`). We must set sample size (for example `NUM = 30`) and the number of samples (for example `NUM_CAMP = 10000`). Now we must have two different matrices which represent the parameters of the bivariate normal distribution under H_0 (no agreement) and H_1 (yes agreement). The values tested in literature (Lin et al., 2002, Table 2) are translated in R code by:

```
> sigma2x0   <- 1 / 1.15
> sigma2y0   <- 1.15
> covxy0     <- 0.95 * sqrt(1 / 1.15 * 1.15)
> mux0       <- 0
> muy0       <- 0.15
> matH0      <- matrix(0,nrow = 2,ncol = 3)
> matH0[1,1] <- sigma2x0
> matH0[1,2] <- covxy0
> matH0[1,3] <- mux0
> matH0[2,1] <- covxy0
> matH0[2,2] <- sigma2y0
> matH0[2,3] <- muy0
> matH0
          [,1] [,2] [,3]
[1,] 0.8695652 0.95 0.00
[2,] 0.9500000 1.15 0.15
```

and

```
> sigma2x1   <- 1 / 1.1
> sigma2y1   <- 1.1
> covxy1     <- 0.9662055 * sqrt(1 / 1.1 * 1.1)
> mux1       <- 0
> muy1       <- 0.1
> matH1      <- matrix(0,nrow = 2,ncol = 3)
> matH1[1,1] <- sigma2x1
> matH1[1,2] <- covxy1
> matH1[1,3] <- mux1
> matH1[2,1] <- covxy1
> matH1[2,2] <- sigma2y1
> matH1[2,3] <- muy1
> matH1
          [,1]      [,2] [,3]
[1,] 0.9090909 0.9662055  0.0
[2,] 0.9662055 1.1000000  0.1
```

Let $\alpha = 0.05$ (default value). Now we have set all the parameters to run `lin.simulation()` (see Figure 1).

Conclusions

The first column of `table` object is the theoretical value of the indices (**Th val**) while the second column (**Thr**) represent the threshold used to determine the rejection region. Theoretical values of α and $1-\beta$ are reported in the third column (**Th prob**). The fourth column (**Mean of est**) represents the inverse transformation of the mean estimate of the agreement measure (see above). We expect the first and the fourth columns to be similar in order to consider the estimate robust. The same conclusion is made between the fifth and the sixth columns which represent the standard deviation of the transformation (**Std of est**) and the mean of the standard deviation (**Mean of std**) respectively. In the seventh column (**Prop rej**) it is calculated the proportion between `NUM_CAMP` runs fall in the rejection region. If we simulate under H_0 then we expect that this value is about $\alpha = 0.05$ (type one error probability) while we expect it is about the true value $1-\beta$ (power) if we simulate under H_1.

Summary

In this paper we describe the **agreement** package. This provides the `lin.simulation()` function to simulate and to perform a complete analysis of an agreement measurement study.

Bibliography

D.G. Altman and J.M. Bland. Measurement in Medicine: The Analysis of Method Comparison Studies. *The Statistician*, 32:302–317, 1983.

L. Lin. A Concordance Correlation Coefficient to Evaluate Reproducibility. *Biometrics*, 45:255–258, 1989.

L. Lin and A.S. Heyadat and B. Sinha and M. Yang. Statistical Methods in Assessing Agreement: Models, Issues, and Tools. *JASA*, 97:257–270, 2002.

```
> lin.simulation(matH0 = matH0,matH1 = matH1,NUM = 30,NUM_CAMP = 10000,underH0 = TRUE)
$table
           Th val     Thr Th prob Mean of est Std of est Mean of std Prop rej
Precision 0.95000 0.97283    0.05     0.95160    0.19110     0.19245     0.07
Accuracy  0.97940 0.99254    0.05     0.97847    0.63310     0.64053     0.00
TDI       0.61997 0.09204    0.05     0.62036    0.25665     0.26147     0.11
CCC       0.93043 0.95994    0.05     0.92812    0.17014     0.16935     0.04
Cpk1      0.83026 0.90620    0.05     0.82192    0.39877     0.40945     0.03
CPk3      0.97892 0.99465    0.05     0.97761    0.83295     0.83488     0.04

$underH0
[1] TRUE

$matH0
          [,1] [,2] [,3]
[1,] 0.8695652 0.95 0.00
[2,] 0.9500000 1.15 0.15

$matH1
          [,1]      [,2] [,3]
[1,] 0.9090909 0.9662055  0.0
[2,] 0.9662055 1.1000000  0.1

$NUM_CAMP
[1] 10000

$NUM
[1] 30

$alpha
[1] 0.05

$rho
[1] 0.95
```

Figure 1: The output of the simulation by `lin.simulation()` command.