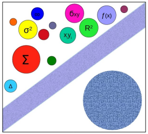
Compind: Composite indicators functions based on frontiers in R

Vidoli Francesco[1], Fusco Elisa[2]

[1]University of Roma Tre
[2]University of Rome La Sapienza



Dealing with Complexity in Society: from Plurality of Data to Synthetic Indicators

17^{th}-18^{th} **September 2015, Padua (Italy)**

Outline

Why a frontier CI package?

The applicative difficulties in applying composite indicators (CI) methods derived from the production frontier analysis (frontier methods, *i.e.* BoD) have often discouraged the adoption of such methods, while having more desirable properties compared to simpler ones.

Why a frontier CI package?

The applicative difficulties in applying composite indicators (CI) methods derived from the production frontier analysis (frontier methods, *i.e.* BoD) have often discouraged the adoption of such methods, while having more desirable properties compared to simpler ones.

Objective: `Compind` package make comparable and easily calculable composite indicators developed with a plurality of methods and supports researcher into robustness analysis through repeated simulations on subsamples of units or variables.

Why a frontier CI package in R?

R is the most comprehensive statistical analysis package available (over 7000 packages);

Why a frontier CI package in R?

R is the most comprehensive statistical analysis package available (over 7000 packages);

R is a programming language (no specific pull-down menu software):

- CI not only as an evaluation tool, but as a part of the main research flow (more general);
- Bootstrap replication (for sensitivity analysis).

Why a frontier CI package in R?

R is the most comprehensive statistical analysis package available (over 7000 packages);

R is a programming language (no specific pull-down menu software):

- CI not only as an evaluation tool, but as a part of the main research flow (more general);
- Bootstrap replication (for sensitivity analysis).

R is free, cross-platform and open source software (open source package code);

Why a frontier CI package in R?

R is the most comprehensive statistical analysis package available (over 7000 packages);

R is a programming language (no specific pull-down menu software):

- CI not only as an evaluation tool, but as a part of the main research flow (more general);
- Bootstrap replication (for sensitivity analysis).

R is free, cross-platform and open source software (open source package code);

R has active user groups (help, documentation, ...).

How design a CI package in R?

The package would have these properties:

- As simple as possible to use;
- The syntax has to be easy and independent (as possible) from the chosen method;
- Package must cover several steps of the CI calculation (not only the weighting and aggregation step).

`Compind` R package contains a plurality of methods; the available methods can be divided into:

- Frontier methods;
- Non frontier methods;
- Utilities.

Outline

Motivation

Compind functionality

- Frontier methods
- Non frontier methods
- Utilities

Conclusion and enhancements

Benefit of the Doubt approach (BoD)

`ci_bod`: Benefit of the Doubt approach (BoD) is the application of Data Envelopment Analysis (DEA) to the field of composite indicators. It was originally proposed by Melyn and Moesen (1991) to evaluate macroeconomic performance.

Reference: *Nardo et al. (2005), "Handbook on constructing composite indicators: Methodology and user guide"*, *OECD*

Benefit of the Doubt approach (BoD)

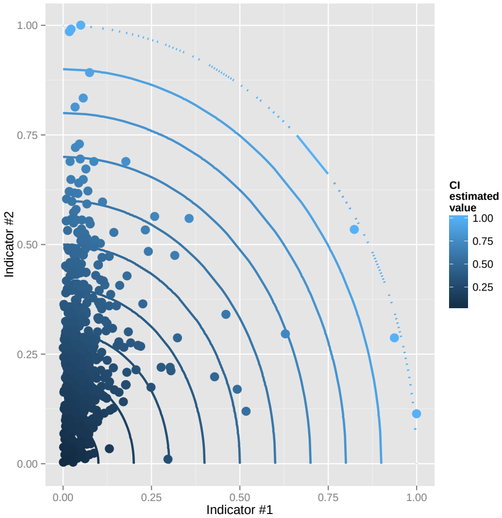
Usage

```
CI_BoD_estimated = ci_bod(x)
```

Arguments

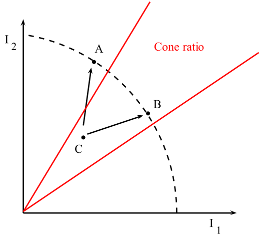
`x`	A data.frame containing score of the simple indicators
`indic_col`	Simple indicators column number.

Two simple indicators



Variance weighted BoD

ci_bod_var_w: Variance weighted Benefit of the Doubt approach (BoD variance weighted) is a particular form of BoD method with additional constraints in the optimization problem.



In particular it has been added weight constraints (in form of an Assurance region type I -AR I) endogenously determined in order to take into account the ratio of the vertical variability of each simple indicator relative to one another.

Reference: *Vidoli, Mazziotta C. (2013)*

Variance weighted BoD

Usage

```
CI_WBoD_estimated = ci_bod_var_w(x)
```

Arguments

`x`	A data.frame containing score of the simple indicators
`indic_col`	Simple indicators column number.
`boot_rep`	The number of bootstrap replicates (default=5000) for the estimates of the nonparametric bootstrap confidence intervals for the variances of the simple indicators.

Robust Benefit of the Doubt approach (RBoD)

`ci_rbod`: Robust Benefit of the Doubt approach (RBoD) is the robust version of the BoD method.
It is based on the concept of the expected minimum input function of order-m, Daraio and Simar (2005).

Reference: *Vidoli, Mazziotta C. (2013)*

Robust Benefit of the Doubt approach (RBoD)

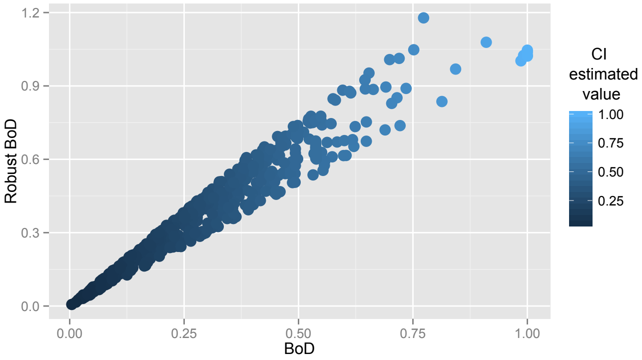
Usage

```
CI_RBoD_estimated = ci_rbod(x,M=20,B=200)
```

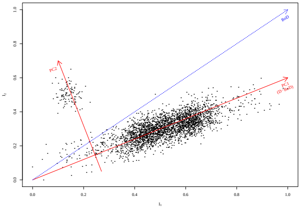
Arguments

`x`	A data.frame containing score of the simple indicators.
`indic_col`	Simple indicators column number.
`M`	The number of elements in each sample.
`B`	The number of bootstap replicates.

Two simple indicators - BoD and RBoD



Directional Benefit of the Doubt (D-BoD)



Reference: *Fusco (2014)*

`ci_bod_dir`: Directional Benefit of the Doubt (D-BoD) model enhance non-compensatory property by introducing directional penalties in a standard BoD model in order to consider the preference structure among simple indicators.

Directional Benefit of the Doubt (D-BoD)

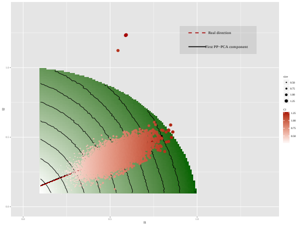
Usage

```
CI_BoD_dir_estimated = ci_bod_dir(x, indic_col, dir)
```

Arguments

`x`	A data.frame containing score of the simple indicators.
`indic_col`	Simple indicators column number.
`dir`	Direction (for example you can set the average rates of substitution).

Directional Robust BoD (D-RBoD)



`ci_rbod_dir`: Directional Robust Benefit of the Doubt approach (D-RBoD) is the robust version of the directional BoD method.

Reference: *Vidoli et al. (2015)*

Directional Robust BoD (D-RBoD)

Usage

```
CI_RBoD_dir_estimated = ci_rbod_dir(x,indic_col,M,B,dir)
```

Arguments

`x`	A data.frame containing score of the simple indicators.
`indic_col`	Simple indicators column number.
`M`	The number of elements in each sample.
`B`	The number of bootstap replicates.
`dir`	Direction (for example you can set the average rates of substitution).

Outline

Weighting method based on Factor Analysis

`ci_factor`: Factor analysis groups together collinear simple indicators to estimate a composite indicator that captures as much as possible of the information common to individual indicators.

Usage

```
CI_Factor_estimated = ci_factor(x,indic_col,method)
```

Arguments

`x`	A data.frame containing score of the simple indicators.
`indic_col`	Simple indicators column number.
`method`	"ONE" (default) first component "ALL" all components multiplied by the relative variance "CH" it can be choose the number of the component.

Weighting method based on geometric aggregation

`ci_mean_geom`: Geometric aggregation lets to bypass the full compensability hypothesis using geometric mean.

Usage

```
CI_Geom_estimated = ci_mean_geom(x, indic_col)
```

Arguments

`x`	A data.frame containing score of the simple indicators.
`indic_col`	Simple indicators column number.

Mazziotta-Pareto Index (MPI) method

`ci_mpi`: Mazziotta-Pareto Index (MPI) is a non-linear composite index method which transforms a set of individual indicators in standardized variables and summarizes them using an arithmetic mean adjusted by a "penalty" coefficient related to the variability of each unit (method of the coefficient of variation penalty).

Mazziotta-Pareto Index (MPI) method

Usage

```
CI_MPI_estimated = ci_mpi(x, indic_col, penalty="POS")
```

Arguments

`x`	A data.frame containing score of the simple indicators.
`indic_col`	Simple indicators column number.
`penalty`	Penalty direction; "POS" (default) in case of increasing or "positive" composite index (e.g., well-being index), "NEG" in case of decreasing or "negative" composite index (e.g., poverty index).

Mean-min function

The Mean-Min Function (MMF) is an intermediate case between arithmetic `mean`, according to which no unbalance is penalized, and `min` function, according to which the penalization is maximum. (Casadio Tarabusi E. & Guarini G., 2013).

Usage

```
CI_mean_min_estimated = ci_mean_min(x, indic_col, alpha, beta)
```

Arguments

`x`	A data.frame containing score of the simple indicators.
`indic_col`	Simple indicators column number.
`alpha`	The intensity of penalisation of unbalance.
`beta`	The intensity of complementarity.

Wroclaw Taxonomic method

`ci_wroclaw`: Wroclaw taxonomy method (also known as the dendric method), originally developed at the University of Wroclaw, is based on the distance from a theoretical unit characterized by the best performance for all indicators considered.

Usage

```
CI_wroclaw_estimated = ci_wroclaw(x,indic_col)
```

Arguments

`x`	A data.frame containing score of the simple indicators.
`indic_col`	Simple indicators column number.

Outline

Normalisation and polarity function

`normalise_ci`: This function lets to normalise simple indicators according to the polarity of each one.

Normalisation and polarity function

Usage

```
new_data = normalise_ci(x, indic_col, polarity, method=1,
            z.mean=0, z.std=1, ties.method ="average")
```

Arguments

`x`	A data.frame containing score of the simple indicators.
`indic_col`	Simple indicators column number.
`method`	Normalisation methods: 1 (default) = standardization or z-scores: $z_{ij} = z.mean \pm \frac{x_{ij} - M_{z_j}}{S_{x_j}} \cdot z.std$ where $\pm$ depends on polarity parameter and $z.mean$ and $z.std$ are the shifting parameters.

Normalisation and polarity function

Usage

```
new_data = normalise_ci(x, indic_col, polarity, method=1,
           z.mean=0, z.std=1, ties.method ="average")
```

Arguments

`x`	A data.frame containing score of the simple indicators.
`indic_col`	Simple indicators column number.
`method`	Normalisation methods:

2 = Min-max method using the following formulation:

if polarity="POS": $\frac{x - min(x)}{max(x) - min(x)}$

if polarity="NEG": $\frac{max(x) - x}{max(x) - min(x)}$

Normalisation and polarity function

Usage

```
new_data = normalise_ci(x, indic_col, polarity, method=1,
            z.mean=0, z.std=1, ties.method ="average")
```

Arguments

`x`	A data.frame containing score of the simple indicators.
`indic_col`	Simple indicators column number.
`method`	Normalisation methods: 3 = Ranking method. If polarity="POS" ranking is increasing, while if polarity="NEG" ranking is decreasing.

Normalisation and polarity function

Usage

```
new_data = normalise_ci(x, indic_col, polarity, method=1,
           z.mean=0, z.std=1, ties.method ="average")
```

Arguments

`polarity`	Polarity vector: "POS" = positive, "NEG" = negative. The polarity of a individual indicator is the sign of the relationship between the indicator and the phenomenon to be measured
`z.mean`	If `method`=1, Average shifting parameter
`z.std`	If `method`=1, Standard deviation parameter
`ties.method`	If `method`=3, a character string specifying how ties are treated.

`Compind` R package contains functions to enhance several approaches to the Composite Indicators methods, focusing, in particular, on the normalisation and weighting-aggregation steps.

`Compind` R package contains functions to enhance several approaches to the Composite Indicators methods, focusing, in particular, on the normalisation and weighting-aggregation steps.

Freely available at:
http://cran.r-project.org/web/packages/Compind/index.html
it allows to build, in a very simple and consistent framework, synthetic indicators according to a plurality of methods based on frontier approach for continuous simple indicators.

Future enhancements

- Graphical functions
- Sensitivity tools

Collaborations in the R package development and improvements are welcome!

Thanks!

Francesco Vidoli *francesco.vidoli@uniroma3.it*

Elisa Fusco *elisa.fusco@uniroma1.it*