*sob*ol *Sen*sitivity *An*alysis
v. 0.99∗

Fosco Mattia Vesely

Indice

1 Introduzione

Il pacchetto R sobSenAn è disegnato per fornire un interfaccia utente interattiva che accompagni l'utilizzatore attraverso i vari passaggi connessi alla stima degli indici si sensitività di un modello ai dati di input. Soluzioni simili sono presenti in **R** ma richiedono che il modello sia richiamabile dall'interno dello stesso ambiente **R**.

Il presente documento è destinato ad illustrare all'utente la modalità d'uso del programma. Per chi intendesse vagliare gli aspetti più tecnici si rimanda alla documentazione del pacchetto, disponibile su GitHub.

2 Installazione

Disponendo di un ambiente R già installato[1] è possibile installare sobSenAn con i comandi forniti in listing 1.

```
install.packages("remotes") # nel caso non sia gia' installato
library(remotes)
remotes::install_github("FoscoV/sobSenAn")
```

Listing 1: Procedura d'installazione entro R

Le istruzioni contenute in listing 1 provvedono automaticamente all'installazione della versione più recente di sobSenAn (od al suo aggiornamento) e delle relative dipendenze. Il pacchetto SobSenAn è basato su strumenti di analisi dati forniti in altri pacchetti. La struttura del pacchetto anzichè uno script è stata adottata, oltre che per la semplificazione nella gestione delle versioni in vista di futuri aggiornamenti, per la più comoda gestione delle dipendenze.

3 Utilizzo

Il funzionamento del pacchetto è divisibile in parti successive:

1. acquisizione di ciascun parametro (3.1)
2. generazione dei valori di input per il modello d'interesse (3.1.1)
3. esecuzione su un modello esterno delle simulazioni secondo i parametri indicati
4. stima degli indici di sensibilità sulla base dei risultati ottenuti (3.2)

Pacchetti **R** disponibili per la stima degli indici di sensitività richiedono la diponibilità di una funzione di classe **predict** per l'iterazione di MonteCarlo sui parametri di input. La strutturazione adottata da sobSenAn genera i dati di input per il modello che li dovrà poi elaborare. Gli output del modello saranno poi analizzati da sobSEnAn per ottenere gli indici. La fase di stima degli indici è quindi asincrona all'esecuzione del modello.

[1] per l'installazione si può fare riferimento al sito ufficiale del progetto CRAN

3.1 Acquisizione dei parametri

sobSenAn contiene l'istruzione biblio2eFast() che avvia una procedura guidata per l'inserimento dei parametri rinvenuti in bibliografia e di seguito elabora la relativa distribuzione.

Dapprima è richiesto il nome del parametro la cui distribuzione si intende valutare. Il nome può essere liberamente attribuito ma non deve essere "Dummy", nè iniziare con una cifra.

Vengono di seguito richiesti i valori disponibili per il parametro[2]. Qualora durante l'elencazione dei valori avvenga un errore di digitazione può essere cancellato prima di premere invio. Se invio è già stato premuto, il valore è ormai letto dal sistema. Per annullare la digitazione è necessario annullare l'operazione[3] e riprendere digitando biblio2eFast(). I tratti già inseriti sono già salvati e non devono essere ripetuti. I valori inseriti per l'ultimo parametro (quello non concluso) sono invece da reinserire.

Premendo "invio" su una linea vuota si conclude la fase di introduzione dei valori e viene richiesto di scegliere la distribuzione più idonea. La scelta è presentata corredata da:

Goodness of Fit (Kolmogorov-Smirnov) che indica quanto la distribuzione corrisponda ai dati indicati

Single Effect che stima l'effetto (medio e massimo) di ciascuno dei parametri forniti nel determinare la distribuzione. Sulla base di questo è stimato quanto un ulteriore parametro possa snaturare la distribuzione (o quanto averne trovato uno di meno avrebbe cambiato l'esito dell'analisi).

Grafico un istogramma riporta la frequenza campionaria indicata mentre le linee colorate riportano la densità di probabilità associata alle distribuzioni in valutazione.

NB: La densità di probabilità e la frequenza sono in ultima istanza due misure diverse; il confronto è utile in termini quantitativi ma non applicabile agli aspetti quantitativi.

Per completezza esplorativa, sono supportate (quindi mostrate) anche distribuzioni che inverosimilmente saranno riscontrate. I criteri su cui fondare la scelta della distribuzione più adatta richiedono l'interpretazione dell'utilizzatore.

Scelta la distribuzione, l'iter automatizzato richiede alcune ulteriori informazioni, che richiedono sempre una risposta y|n. Sono supportate distribuzioni discrete, aventi cioè numeri interi[4], qualora sia il caso, rispondere positivamente a questa richiesta.

Distribuzioni troncate possono essere definite (se richiesto) indicando i valori limite delle probabilità associate. Canonicamente le code inferiori alla probabilità 0.1 e superiori a 0.9. Alternativamente i punti di troncatura possono essere indicati per il loro valore assoluto: alcuni parametri possono avere un limite inferiore pari a 0 (non accettando valori nulli e negativi) e privi di un limite superiore. Il limite di troncatura può essere indicato in termini di probabilità cumulata di frequenza $[0,1]$ od in termini di valore assoluto $(-Inf,+Inf)$.

[2] possono essere inseriti uno alla volta premendo "invio" una volta dopo ciascun inserimento

[3] Esc su MSWindows, ctrl+**C** su linux

[4] Un esempio potrebbero essere livelli di resistenza ad un patogeno

In virtù di una sensibilità di eFAST ai valori estremi della distribuzione dei parametri, sono impiegate le troncature di default a 0.1 e 0.9 della curva di distribuzione di probabilità[5].

Qualora si desideri una distribuzione aperta inferiormente, superiormente od entrambe è necessario superare l'attribuzione di default delle troncature e procedere con l'inserimento dei valori $-Inf$ e $+Inf$ per i valori assoluti di troncamento.

3.1.1 Concludere l'introduzione dei parametri

Rispondendo "n" alla domanda se inserire un ulteriore parametro, si accede all'ultima fase prima dell'esecuzione dei modelli.

All'utilizzatore è richiesto di indicare un file da generare che conterrà una colonna per ciascuno dei parametri introdotti oltre ad una colonna "Dummy" aggiuntiva (requisito per un confronto tra i parametri eseguito entro il programma) che non sarà considerata dal modello di simulazione. Il file dovrà essere impiegato come input al modello di simulazione. I risultati andranno aggiunti in una colonna di modo tale che corrispondano alla riga di input con cui sono stati ottenuti. Possono essere inserite diverse colonne di risultati che saranno tutte analizzate in funzione dei dati di input (interazioni tra output di simulazioni non sono supportate).

Nella cartella[6] "Documenti" (windows) o in "~/" (linux) sarà creato un file "Hyperspace.SAd" da utilizzare per la successiva fase di analisi degli output delle simulazioni. Il nome del file non è vincolante e può essere modificato. In esso sono contenute tutte le informazioni relative all'iperspazio dei parametri generato. Si ricordi che qualora un nuovo iperspazio venga definito, il file verrà sovrascritto.

L'esportazione delle distribuzioni registrate figura tra le modifiche a sessioni precedenti.

3.1.2 Modificare precedenti sessioni

Si noti che tutte le opzioni incluse in questo paragrafo sono disponibili anche per coloro che avessero interrotto la procedura biblio2eFast() forzatamente[7]. Altrimenti una sessione già conclusa può essere ulteriormente modificata impiegando loadSensSession() che richiede di indicare il file di salvataggio .SAd che si intente aggiornare.

Aggiungere parametri Una volta caricati i dati precedenti, la già considerata funzione biblio2eFast() consentirà l'inserimento di ulteriori dati e la generazione dei valori esplorativi dell'iperspazio dei parametri.

Modificare la distribuzione di un parametro Impiegando la funzione SAeditPara() è possibile aggiungere dei valori ed un parametro già inserito. La rimozione di singoli valori non è supportata ed è necessario rimuovere interamente il parametro impiegando SAdelPara() e procedere ad un nuovo inserimento con biblio2eFast().

[5] si ricordi che una proprietà delle curve di distribuzione è che una probabilità non nulla sia associata a tutti i valori compresi tra $-Inf$inito e $+Inf$inito

[6] Si fa di seguito riferimento alle impostazioni di default. Sono personalizzabili utilizzando l'istruzione **setwd**("nuovo␣percorso␣predefinito")

[7] Esc su Windows, ctrl+**C** in linux

Cambiare il numero di Campioni Diverse fonti[8][1] riportano 65 come il numero minimo di campionamenti da effettuare su ogni traiettoria di campionamento. Questo minimo è quello impiegato di default da sobSenAn, per effettuare un campionamento più fitto è possibile, una volta conclusa la fase di generazione dei valori impiegare la funzione SAmorSam() inserendo tra parentesi il numero di campioni che si desidera effettuare. La funzione richiede di selezionare il file .SAd relativo ai parametri da indagare. Autonomamente sostituisce il file dei valori per il modello di simulazione e, di conseguenza, aggiorna il file .SAd.

Impiegare una distribuzione non supportata L'impiego di una distribuzione non supportata non è ovviamente supportato. È tuttavia possibile.

Poichè il parametro deve essere generato armonicamente agli altri, si consiglia di inserirlo con distribuzione uniforme avente range numerico $[0,1]$. Si noti che tale range numerico dovrà essere esplicitato alla domanda relativa alla troncatura. In seguito alla generazione dei parametri, questo potrà essere manualmente modificato nel file di esplorazione dell'iperspazio dei parametri (3.1.1). Il file .SAd non richiede la modifica di questo aspetto.

Esportare le distribuzioni registrate Dopo avere caricato l'iperspazio d'interesse (3.1.2) è sufficiente impiegare SAexport() .

3.1.3 Sessioni concluse

Nel momento in cui viene generato il file con i valori dei parametri ed il file .SAd, la sessione di **R** in corso non rimane altrimenti influenzata dall'inserimento avvenuto. Per recuperare i dati si faccia riferimento al paragrafo 3.1.2. Altrimenti è possibile descrivere direttamente un nuovo iperspazio con la già descritta funzione biblio2eFast().

3.2 Analisi Sensitività

output2Sens() provvede autonomamente a tutte le parti di analisi. Richiede di indicare il file contenente gli input (e gli output), forniti (ed ottenuti) (d)al modello di simulazione. Richiede quindi di segnalare il file .SAd generato contestualmente ai parametri che si stanno analizzando.

Il formato del file deve essere lo stesso generato in uscita (valori separati da tabulazione con intestazioni di colonna nella prima riga). Il programma provvede autonomamente al riconoscimento dei parametri e dei risultati. L'influenza di tutti i parametri viene valutata su tutti i risultati forniti.

Nella cartella SAfast che viene generata in "Documenti" (windows) o in "~/" (linux) si trovano "SAresults.csv" contenente gli indici statistici per ciascun parametro ed un file .pdf per ciascuno parametro analizzato con i relativi grafici.

Il file SAresults.csv contiene, per ogni risultato le seguenti colonne:

_Si Frazione della varianza del risultato spiegata dalla variazione del parametro

_Si_PVal First Order Sensitivity Index p-value

_STi Frazione della varianza attribuita al parametro ed alle sue interazionei con altri parametri

_STi_PVal Total Order Effect Index p-value

_SCi Varianza attribuita a tutti gli altri parametri

_Si_CoEff_of_Var t-test tra p-value del Si del parametro e di Dummy

_STi_CoEff_of_Var t-test tra p-value del STi del parametro e di Dummy

_Si_ErrorBar Errore nella stima del parametro (in base al ricampionamento)

_STi_ErrorBar Errore nella stima del parametro (in base al ricampionamento)

Riferimenti bibliografici

[1] Kieran Alden et al. *spartan: Simulation Parameter Analysis R Toolkit ApplicatioN: Spartan*. R package version 2.3. 2015. URL: https://CRAN.R-project.org/package=spartan.

[2] Dutang Christophe e Savicky Petr. *randtoolbox: Generating and Testing Random Numbers*. R package version 1.17. 2015.

[3] Traci E Clemons e Edwin L Bradley. «A nonparametric measure of the overlapping coefficient». In: *Computational statistics & data analysis* 34.1 (2000), pp. 51–61.

[4] Roberto Confalonieri. «Monte Carlo based sensitivity analysis of two crop simulators and considerations on model balance». In: *European Journal of Agronomy* 33.2 (2010), pp. 89–93.

[5] R Confalonieri et al. «Comparison of sensitivity analysis techniques: a case study with the rice model WARM». In: *Ecological Modelling* 221.16 (2010), pp. 1897–1906.

[6] Marie Laure Delignette-Muller e Christophe Dutang. «fitdistrplus: An R Package for Fitting Distributions». In: *Journal of Statistical Software* 64.4 (2015), pp. 1–34. URL: http://www.jstatsoft.org/v64/i04/.

[7] Tal Galili. *Edfun*. 2016.

[8] Simeone Marino et al. «A methodology for performing global uncertainty and sensitivity analysis in systems biology». In: *Journal of theoretical biology* 254.1 (2008), pp. 178–196.

[9] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2016. URL: https://www.R-project.org/.

[10] Andrea Saltelli et al. «A new sample-based algorithms to compute the total sensitivity index». In: *arXiv preprint arXiv:1703.05799* (2017).

[11] Andrea Saltelli et al. «Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index». In: *Computer Physics Communications* 181.2 (2010), pp. 259–270.

[12] Ilya M Sobol. «Sensitivity estimates for nonlinear mathematical models». In: *Mathematical Modelling and Computational Experiments* 1.4 (1993), pp. 407–414.

[13] W. N. Venables e B. D. Ripley. *Modern Applied Statistics with S*. Fourth. ISBN 0-387-95457-0. New York: Springer, 2002. URL: http://www.stats.ox.ac.uk/pub/MASS4.