nlsCompare: An R package for comparing nonlinear least squares packages in R

by Arkajyoti Bhattacharjee[1], John C. Nash[2], and Heather Turner[3]

Abstract Nonlinear least squares (nls) is extensively used in nonlinear modeling. This paper introduces the package **nlscompare** that attempts to compare existing functions or packages in nls like `nls`(**base**), `nlsLM`(**minpack.lm**) and a new package **nlsj** developed by the authors as a product of Google Summer of Code (GSoC), 2021.

Introduction

While working on the GSoC project **Improvements to nls()** under The R Project for Statistical Computing, the authors realized that to be able to compare the improvements made via `nlsj`, one needs to compare existing packages on nls in R. **nlscompare**:

- contains test data files based on packages like NRAIA and NISTnls
- contains spreadsheets containing the methods to compare
- generates spreadsheets that compare existing or new package functions on nls with `nls` and comments whether the former is an improvement over `nls`

nlscompare draws inspiration from R's testing library **testthat** and uses ideas from packages like **benchmarkme**. Currently, this has been developed keeping the R environment in mind.

Motivation or why nlscompare or the motivation for nlscompare

Regression testing is useful not only to validate correctness of a program but often also for tracking the quality of its output. While working on improving `nls`, author[2] noted that to compare the improvements made over `nls` via **nlsj**, unit testing via **testthat** is not enough. Further, to be able to add more improvements to **nlsj**, it is required to compare it with other existing packages' nls functions. This would allow us to note which methods are supported by one function, but not in another and to check the consistency of same results from different functions.

nlscompare attempts to achieve the above, outputting a simple informative spreadsheet that indicates whether an existing or new package's function provides consistent results with `nls`. Further, it outputs an error log that indicates whether only `nls` or the other funcion being compared or both fail in a particular problem with particular methods (algorithms, starting points, etc.)

Workflow

The **nlscompare** needs to be run in the following hierarchical steps -

1. `setup_dir()`: choose and set the working directory save final csv outputs.
2. `machineId()`: create the machine ID to be used in the output csv files.
3. `create_csvs()`: create dataframes *spreadsheet* and *spreadsheet_error* which contain the comparisons and error log respectively.
4. `csv_exists`: check if the csv-s **nlsDatabase.csv** and **nlsErrorLog.csv** exist already in the directory chosen in 1.
5. `run()`: this uses the problem names from **problems.csv**, the methods from **methods.csv**, does the comparisons and error logging and edits the dataframes accordingly.
6. `write_csvs`: write the dataframes *spreadsheet* and *spreadsheet_error* into **nlsDatabase.csv** and **nlsErrorLog.csv**
7. `clear_nls` : this delete variables declared in the environment due to running of the above functions

Package content details

methods.csv

This csv file contains three columns: ?? add what supports what in the form of a table

- **solver:** This is the name of the function to be used to solve the nonlinear leat squares problem. Currently, it supports `nlsj::nlsj` and `minpack.lm::nlsLM`.
- **algorithm:** This is the name of the algorithm used. Currently, supports `default`, `plinear`, `port`
- **control:** uses the control argument in the nls runline ??One can use what the concerned package's function's control support.

problems.csv

This csv file contains one column:

- **FileName:** This contains the names of files present in *test_files* and is used by the `run()` function.

test_files

This directory contains all the test files on which the various comparisons are made. Each test file has a naming structure: "problemName_x" where x is an integer number. "1" means that it is a parent test file of the "problemName" family of test files. It mainly contains -

- `NLSdata`: A dataframe that contains the data
- `NLSstart`: Starting values for the nls problem
- `NLSformula`: A formula for the nls problem
- `NLSlower`: An lower bound for the nls problem
- `NLSupper`: An upper bound for the nls problem
- `NLSweights`: weights for the nls problem
- `NLSsubset`: for subsetting the dataframe

If x in "_x" is greater than 1, it means that it is a children test file belonging to the same "problemName family of test files. It contains a subset of the contents as mentioned above. For example, *NLSprob_2.R* may contain just a new `NLSstart` or may contain `NLSupper` and `NLSlower` and so on.

An example R code snippet??

Coverage

The idea of hierarchy in test files allows to cover -

- multiple starting values that may be good or bad
- bounded problems having a good or bad bounds
- multiple weights for the same problem that comprise of good or bad weights
- ability to work with subsets of the problem's data

The **methods.csv** allows the possibility to cover existing and new packages' - (??repetitive)

- functions as **solver**s viz. `nlsj::nlsj`, `nlsr::nlxb` and `minpack.lm::nlsLM`
- existing and new **algorithm**s viz. "marquardt", "plinear", "port" and "Gauss-Newton" and its variants

The `run()` function compares all essential estimation quantities like -

- parameters
- residuals
- deviance
- jacobian
- convergence

Future work

A few things we would like to incorporate in **nlscompare** -

- include **nlsr::nlxb** as a solver and use its corresponding algorithms and controls
- Language invariance. It will be useful to compare nls functions not only limited to R but to other popular programming languages like Python, Julia, etc.

Arkajyoti Bhattacharjee[1]
Mathematics and Statistics Department
Indian Institute of Technology, Kanpur
India
arkastat98@gmail.com

John C. Nash[2]
University of Ottawa
line 1 affiliation 1
line 2 affiliation 1
https://journal.r-project.org
ORCiD: 0000-0002-9079-593X
nashjc@uottawa.ca

Heather Turner[3]
Department of Statistics
University of Warwick
Coventry
United Kingdom
ORCiD: 0000-0002-1256-3375
ht@heatherturner.net