Metamodels and displacements inversion
Notes for the R demo

Rodolphe Le Riche, Nicolas Durrande and Valérie Cayol

October 12, 2017

1 Overview

This directory contains demo files for identifying a spherical reservoir from surface displacements. The programming language is R. Some data files are provided that describe the digital terrain model and the targetted surface displacements.

The identification is performed by optimization where a distance between data (either synthetic or measured) and simulation results is minimized on *optimization variables*. The spherical magma chamber is described by its position (variables xs, ys and zs), its radius (variable a) and its overpressure (p), for a total of 5 optimization variables. The Mogi model [] simulates the displacements created by a spherical magma chamber at the surface of a digital terrain.

The optimization problem is tackled with the EGO algorithm [1]. The generic files for the Gaussian processes and their identification can be found in the `../labSessions/` directory. The files provided are an implementation independent from other libraries meant for both completness and simplicity.

2 Prerequisites

1. Have R available on your computer, cf. `https://www.r-project.org/`
2. Optionally (but really helpful) have rstudio installed, cf. `https://www.rstudio.com`

3. Install the `lhs` package (either from Tools / Install Package in rstudio or with the command `install.packages("lhs")`.

4. Optional: if you want to load the data that are in matlab format (`file_name.mat`), install the "R.matlab" package (either from Tools / Install Package in rstudio or with the command `install.packages("R.matlab")`. But you can also load directly the ascii csv file (`file_name.csv`) from R

3 Running the demo step by step

0. Open with rstudio the file `mainInversionPunctualDisplSource.R`, or open the file with any text editor and start R in a console. We will then proceed in steps, where each step is announced by a line of comments (`#### step #####...`), and go from top to bottom. During each step, select sections of code with the mouse and *execute* them, either (rstudio) by hitting `ctrl + return` or, (other text editors) by cutting and pasting in the R console. The main steps are as follows.

1. Execute `load utilities`, `input for variables identification` (where the bounds on the variables are set) and `load data`.

2. Execute `design of experiments`: the hypercube of possible variables is filled with a Latin Hypercube Sampling. The number of points `nbinit`, a measure of how much randomness is devoted to the identification, is set there. The weighted least squares distances (WLS) to the target for each set of variables is calculated. The distances are normalized by $\log(1 + WLS)$ follow by centering and standardization as a remedy against a large spread with outliers.

3. Execute `build a kriging model`: this means selecting a kernel for the Gaussian Process (GP) (default is `kMat52` for the 5/2 Matèrn kernel) and maximizing its log-likelihood with respect to the GP parameters (process variance and length scales).

4. Execute `test the kriging model`, which comprises creating a test set, predicting at the test variables, and measuring the discrepancy between the predictions and the test data (RMSE, Q2 and plots).

5. Execute `model identification`: iteratively maximize the Expected Improvement (EI) with respect to the model variables, update the design of experiments, and update the GP model (maximize its log-likelihood with respect to the GP parameters).

Both the log-likelihood and the EI maximizations are performed by restarting `nbtry` times a BFGS algorithm with bounds on the variables from randomly chosen points. This strategy is meant to avoid getting trapped in the local minima that both these problems have (in particular EI).

4 Files list

- Generic files for building Gaussian Processes and optimizing with them are found in the `../labSessions/` directory. This directory contains the complementary demo files for identifying a spherical reservoir from surface displacements.
- `mainInversionPunctualDisplSource.R` : main script for the demo.
- `mogi_3D.R` : calculate displacements on a digital terrain model from a point-wise spherical source.
- `plots_3d_full_grid.R` : Load a csv file (full grid), and plots its 3d data.
- `process_3d_full_grid_from_matlab.R` : Load a matlab file (full grid), processes it so that it is plotted and (commented out but working) saved in csv format. Displacements are calculated with `mogi_3D.R`.
- `wls_ulos.R` : calculates the weighted least squares distance between the model projected displacements and the target.
- + data files ending in `.mat` (matlab format) or `.csv` (csv format).

References

[1] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.