RAFF: Robust Algebraic Fitting Function in Julia

11 March 2019

Summary

Let $f : \Re^n \to \Re$ be a function whose mathematical description is not available. This function can be, for example, a black-box, a proprietary computer program or an experiment. Suppose that a data set $S = \{(t_1, f(t_1)), \ldots, (t_m, f(t_m))\}$ of m observations of f is available and we want to approximate f by a known model ϕ. Model ϕ can be defined as $\phi(x; t)$, where t are the n independent variables of f and x represents some parameters of ϕ. `RAFF` (Robust Algebraic Fitting Function) is a Julia package developed to find the optimal parameters x for ϕ in order to adjust it to the observed values S of the unknown function f.

The adjustment of mathematical functions to data is a problem that appears in many areas of science. When data comes from real experiments, non-expected errors may cause the appearance of outliers. Detection of outliers is always regarded as the statistical part of data adjustment. `RAFF` provides automatic detection of outliers using a voting system. It is an optimization-based package, based on algorithms for Lower Order-Value Optimization (LOVO) which were introduced in (Andreani, Dunder, and Martínez 2005) and revisited in (Andreani et al. 2009) to fit the user-provided models ϕ to experimental data. Recently, a good review can be found in (J. M. Martínez 2012). In order to find a robust adjustment, a voting system is used, which is also responsible for the detection of possible outliers.

Functionality

`RAFF` main methods expect as input a data set of the observed data and a model function, whose parameters one intends to adjust. The model function is a regular Julia function with 2 arguments: x represents the parameters of the model and t represents the arguments of function f. The following function is

an example of a model representing a circle

$$\phi(x,t) = (t_1 - x_1)^2 + (t_2 - x_2)^2 - x_3^2.$$

Note that x_1 and x_2 are parameters related with the center of the circle, while x_3 is related to its radius. Therefore, this model contains 3 parameters. Since we think that the unknown function f is a circle, it is obvious that its arguments are a two-dimensional vector t. The observed data can be represented by the following table:

t_1	t_2	$f(t_1, t_2)$
6.11129	1.0000	0.0000
6.12134	1.0038	0.0000
5.73863	2.1833	0.0000
5.35241	3.5871	0.0000
4.76209	4.2487	0.0000
3.57131	5.3119	0.0000
2.75896	5.7070	0.0000
2.07888	5.7852	0.0000
1.65623	6.0197	0.0000
-0.31324	5.7656	0.0000
-2.18214	4.9540	0.0000
-2.45642	4.4935	0.0000
-3.15965	3.8885	0.0000
-3.77514	2.5237	0.0000
-4.03242	1.4327	0.0000
-3.93137 -	0.0512	0.0000
-2.55007 -	2.6777	0.0000
0.16087 -	4.0584	0.0000
1.83171 -	3.9190	0.0000
4.19091 -	2.7741	0.0000
5.67758 -	0.6784	0.0000
5.81633	0.2874	0.0000
5.94573	0.3777	0.0000
-1.12132	3.1213	0.0000
-3.59619 -	3.5961	0.0000

In this example, we are trying to find a circle to fit the observed data. Therefore, all the values of f should be zero. The observed data is shown in Figure 1. The dots represent the observed data, the red ones being the outliers. They were generated as a perturbation of the points lying in the blue dashed circle. Using the Least Squares technique with the model above, the red circle is found. When RAFF is applied to the same problem, it correctly identifies the two outliers. The resulting circle is depicted as the green circle, very close the true circle.

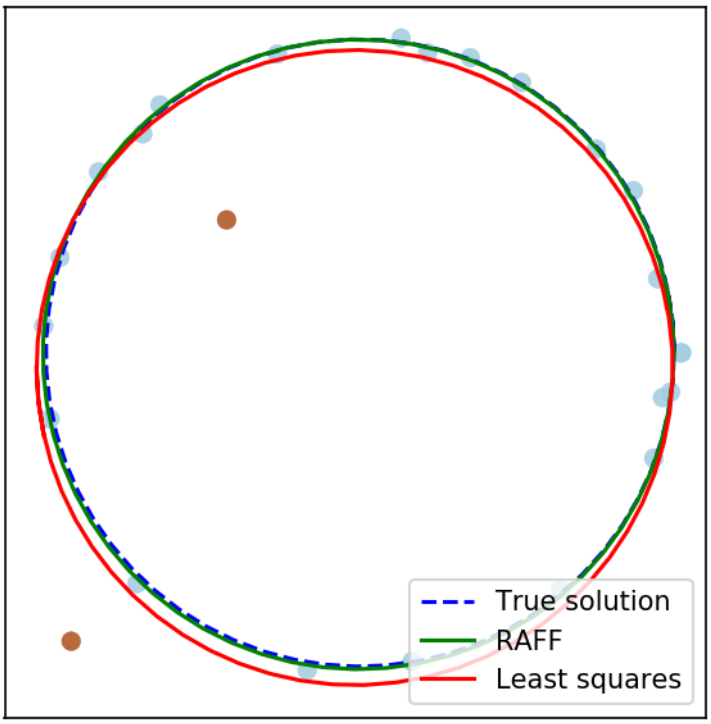

Figure 1: Points representing a circle. The red dots are two outliers that should be ignored. The blue dashed circle is the true one, while the red was obtained by traditional Least Squares techniques and the green one was obtained by RAFF.

Additional features

The user may also provide more information to `RAFF`, such as the expected number of *trusted* observations. Additional methods and options are also available to more advanced users, such as generation of random test data and multistart strategies. Derivatives of the model ϕ can also be provided, what results in a faster executing time. When they are not provided by the user, `RAFF` uses Julia's `ForwardDiff` (Revels, Lubin, and Papamarkou 2016) package.

`RAFF` can be run in serial, parallel and distributed environments. Parallel and distributed methods use the native `Distributed` package. The distributed version is a centralized implementation that does not use shared arrays, therefore, can be run both locally or in a cluster of computers.

This package is intended to be used by all experimental researchers who know a little about mathematical modeling and fitting functions.

References

Andreani, R., C. Dunder, and J. M. Martínez. 2005. "Nonlinear-Programming Reformulation of the Order-Value Optimization Problem." *Mathematical Methods of Operations Research* 61 (3). Springer: 365–84. doi:10.1007/s001860400410.

Andreani, R., J. M. Martínez, L. Martínez, and F. S. Yano. 2009. "Low Order-Value Optimization and Applications." *Journal of Global Optimization* 43 (1): 1–22. doi:10.1007/s10898-008-9280-3.

Martínez, José Mario. 2012. "Generalized Order-Value Optimization." *Top* 20 (1). Springer: 75–98.

Revels, J., M. Lubin, and T. Papamarkou. 2016. "Forward-Mode Automatic Differentiation in Julia." *ArXiv:1607.07892 [Cs.MS]*. https://arxiv.org/abs/1607.07892.