

# How to Use Absolute-Error-Minimizing Software to Minimize Relative Error: Practitioner’s Guide

Afshin Gholamy<sup>1</sup> and Vladik Kreinovich<sup>2</sup>

<sup>1</sup>Department of Geological Sciences

<sup>2</sup>Department of Computer Science

University of Texas at El Paso

500 W. University

El Paso, TX 79968, USA

afshingholamy@gmail.com, vladik@utep.edu

## Abstract

In many engineering and scientific problems, there is a need to find the parameters of a dependence from the experimental data. There exist several software packages that find the values for these parameters – values for which the mean square value of the absolute approximation error is the smallest. In practice, however, we are often interested in minimizing the mean square value of the *relative* approximation error. In this paper, we show how we can use the absolute-error-minimizing software to minimize the relative error.

## 1 Formulation of the Problem

**Practical problem.** In many practical situations, we know that the dependence between the quantity  $y$  and related quantities  $x_1, \dots, x_n$  has the known form, i.e., the form  $y = f(x_1, \dots, x_n, c_1, \dots, c_m)$  with a known function  $f$  and unknown values of the coefficients  $c_1, \dots, c_m$ .

**Examples.** For example, we may know that  $y$  linearly depends on  $x_i$ , in which case the desired dependence has the form

$$y = c_1 \cdot x_1 + \dots + c_n \cdot x_n + c_{m+1}$$

for some unknown values  $c_i$ .

We may have a quadratic dependence, in which case

$$y = c_1 \cdot x_1 + \dots + c_n \cdot x_n + c_{n+1} + c_{n+2} \cdot x_1^2 + c_{n+3} \cdot x_1 \cdot x_2 + \dots$$

These two cases are particular cases of a more general situation in which the dependence of the expression  $f(x_1, \dots, x_n, c_1, \dots, c_m)$  on the coefficients  $c_j$  is

linear, i.e., in which

$$f(x_1, \dots, x_n, c_1, \dots, c_m) = \sum_{j=1}^m c_j \cdot f_j(x_1, \dots, x_n)$$

for given functions  $f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)$ .

We may also have a more complex dependence, in which the dependence on the coefficients  $c_j$  is nonlinear. For example, in radioactive decay, once we know the initial amount  $y_0$  of the radioactive material, then the amount  $y$  remaining after time  $x_1$  is described by the formula  $y = y_0 \cdot e^{-c_1 \cdot x_1}$ , for an appropriate to-be-determined coefficient  $c_1$ .

**What information we have to solve this problem.** In the above situation, it is necessary to determine the values of these parameters  $c_1, \dots, c_m$  from the experimental data.

In engineering, geosciences, and in many other application areas, this is known as the *inverse problem* – as opposed to the *forward problem*, when we know the values of  $x_i$  and  $c_j$ , and we use the dependence  $y = f(x_1, \dots, x_n, c_1, \dots, c_m)$  to predict the value of the quantity  $y$ .

To solve the inverse problem, we can use the results coming out of several ( $K$ ) situations in which both  $y$  and  $x_i$  have been measured, and we use the results  $y^{(k)}$  and  $x_i^{(k)}$  of these measurements to find the coefficients  $c_i$  for which, for each  $k$  from 1 to  $K$ , we have

$$y^{(k)} \approx f\left(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m\right).$$

It is important to emphasize that since measurements are never absolutely accurate – there is always some measurement error – the measurement result  $y^{(k)}$  is only approximately equal to the quantity  $f\left(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m\right)$ ; see, e.g., [2].

**There exist many software packages that minimize the absolute error.**

The standard approach to solving the above problem is the Least Squares approach (see, e.g., [3]), in which we find the coefficients  $c_1, \dots, c_m$  for which the mean square value of the (absolute) approximation error is minimized, i.e., that minimize the expression  $\sum_{k=1}^K (\Delta y^{(k)})^2$ , where by  $\Delta y^{(k)}$ , we denoted the (absolute) approximation errors

$$\Delta y^{(k)} \stackrel{\text{def}}{=} y^{(k)} - f\left(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m\right).$$

For example, MatLab has such programs – both for the linear and for the nonlinear cases. In addition, neural network packages – in particular, the MatLab neural network toolbox – minimize the above sum of squares; see, e.g., [1].

Some of these packages only deal with situations in which the dependence  $f(x_1, \dots, x_n, c_1, \dots, c_m)$  is linear in terms of the coefficients. In these cases, the

above problem takes the form

$$y^{(k)} \approx \sum_{j=1}^m c_j \cdot f_j \left( x^{(1)}, \dots, x_n^{(k)} \right),$$

and the Least Squares idea means minimizing the sum  $\sum_{j=1}^m (\Delta y^{(k)})^2$ , where

$$\Delta y^{(k)} = y^{(k)} - \sum_{j=1}^m c_j \cdot f_j \left( x^{(1)}, \dots, x_n^{(k)} \right).$$

These packages usually assume that we know the values  $q^{(k)}$  and  $p_1^{(k)}, \dots, p_m^{(k)}$  for all  $k$  from 1 to  $K$ , and they find the coefficients  $c_1, \dots, c_m$  for which the sum

$$\sum_{k=1}^K \left( q^{(k)} - \sum_{j=1}^m c_j \cdot p_j^{(k)} \right)^2$$

attains its smallest possible value. To apply such a package to our problem, it is sufficient to take  $q^{(k)} = y^{(k)}$  and  $p_j^{(k)} = f_j \left( x_1^{(k)}, \dots, x_n^{(k)} \right)$ .

**Often, we need to minimize relative errors instead.** While in many practical situations, minimizing the absolute approximation errors is a reasonable idea, in many other situations, it is more appropriate to minimize *relative* approximation errors

$$\delta y^{(k)} \stackrel{\text{def}}{=} \frac{\Delta y^{(k)}}{y^{(k)}},$$

i.e., to be precise, to minimize the sum  $\sum_{k=1}^K (\delta y^{(k)})^2$ .

For example, if the desired dependence covers a wide range of possible values of  $y$ , e.g., ranging from  $y = 10$  to  $y = 1000$ , it makes more sense to want to approximate all these values with the same relative accuracy – e.g., 5% or 10% – than with the same absolute accuracy, say, 5 – in the case of absolute accuracy, we have a very crude 50% accurate approximation for small values  $y \approx 10$  and an unnecessarily accurate (0.5% accurate) approximation of values  $y \approx 1000$  (unnecessarily accurate since we usually cannot even measure  $y$  with such a high accuracy).

**Problem.** The problem is that, in contrast to minimization of absolute error, for which there are several available software packages, not many packages are available for minimizing the relative error.

In principle, we can write our own code for solving this problem, but it would be much easier if we could simply use the existing software.

**What we do in this paper.** In this paper, we show how we can use absolute-error-minimizing software to minimize relative errors. In Section 2, we describe how to do it for the case when the dependence on the coefficients  $c_j$  is linear. In Section 3, we consider the general case, when the dependence on the coefficients  $c_j$  may be non-linear.

## 2 Case When the Dependence on the Coefficients $c_j$ is Linear

**Description of the case: reminder.** We assume that

$$y = \sum_{j=1}^m c_j \cdot f_j(x_1, \dots, x_n),$$

for known functions  $f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)$ . Based on the results  $y^{(k)}$  and  $x_i^{(k)}$  of measuring the corresponding quantities, we want to find the coefficients  $c_1, \dots, c_m$  for which

$$y^{(k)} \approx \sum_{j=1}^m c_j \cdot f_j(x_1^{(k)}, \dots, x_n^{(k)})$$

for all  $k$  from 1 to  $K$ . To be more precise, we want to find the values  $c_1, \dots, c_m$  for which the sum  $\sum_{k=1}^K (\delta y^{(k)})^2$  is the smallest possible, where  $\delta y^{(k)} \stackrel{\text{def}}{=} \frac{\Delta y^{(k)}}{y^{(k)}}$  and

$$\Delta y^{(k)} \stackrel{\text{def}}{=} y^{(k)} - \sum_{j=1}^m c_j \cdot f_j(x_1^{(k)}, \dots, x_n^{(k)}).$$

**We would like to use a Least Squares package.** To solve our problem, we would like to use a Least Squares package, that, given the values  $q^{(k)}$  and  $p_1^{(k)}, \dots, p_m^{(k)}$  for all  $k$  from 1 to  $K$ , find the values  $c_j$  that minimize the sum

$$\sum_{k=1}^K \left( p^{(k)} - \sum_{j=1}^m c_j \cdot p_j^{(k)} \right)^2.$$

**Analysis of the problem.** Substituting the expression

$$\Delta y^{(k)} = y^{(k)} - \sum_{j=1}^m c_j \cdot f_j(x_1^{(k)}, \dots, x_n^{(k)})$$

into the definition of the relative approximation error  $\delta y^{(k)}$ , we conclude that

$$\delta y^{(k)} = 1 - \sum_{j=1}^m c_j \cdot \frac{f_j(x_1^{(k)}, \dots, x_n^{(k)})}{y^{(k)}}.$$

Thus, the problem of minimizing the sum  $\sum_{k=1}^K (\delta y^{(k)})^2$  of the squares of relative errors is equivalent to the problem of minimizing the sum

$$\sum_{k=1}^K \left( q^{(k)} - \sum_{j=1}^m c_j \cdot p_j^{(k)} \right)^2,$$

with  $q^{(k)} = 1$  and  $p_j^{(k)} = \frac{f_j(x_1^{(k)}, \dots, x_n^{(k)})}{y^{(k)}}$ . So, we arrive at the following recommendation.

**Recommendation.** Our recommendation is to find the coefficients  $c_1, \dots, c_m$  by applying a Least Squares package to the values  $q^{(k)} = 1$  and

$$p_j^{(k)} = \frac{f_j(x_1^{(k)}, \dots, x_n^{(k)})}{y^{(k)}}.$$

### 3 General (Possibly Nonlinear) Case

**First idea: description.** The above approach can be naturally extended to the nonlinear case. Namely, minimizing the sum of relative errors

$$\sum_{k=1}^K (\delta y^{(k)})^2 = \sum_{k=1}^K \left( \frac{y^{(k)} - f(x_1^{(1)}, \dots, x_n^{(k)}, c_1, \dots, c_m)}{y^{(k)}} \right)^2$$

is equivalent to minimizing the sum of the absolute differences

$$\sum_{k=1}^K \left( z^{(k)} - g(x_1^{(1)}, \dots, x_n^{(k)}, y^{(k)}, c_1, \dots, c_m) \right)^2,$$

where  $z^{(k)} = 1$  and

$$g(x_1, \dots, x_n, y, c_1, \dots, c_m) \stackrel{\text{def}}{=} \frac{f(x_1, \dots, x_n, c_1, \dots, c_m)}{y}.$$

Thus, we arrive at the following recommendation.

**First idea: resulting recommendations.** To minimize the relative error, apply the absolute-error-minimizing software to find the coefficients  $c_1, \dots, c_m$  from the condition that

$$z^{(k)} \approx g(x_1^{(1)}, \dots, x_n^{(k)}, y^{(k)}, c_1, \dots, c_m)$$

for  $k = 1, \dots, K$ , where we denoted  $z^{(k)} = 1$  and

$$g(x_1, \dots, x_n, y, c_1, \dots, c_m) \stackrel{\text{def}}{=} \frac{f(x_1, \dots, x_n, c_1, \dots, c_m)}{y}.$$

**Second idea: description.** Alternatively, we can apply the existing software to approximate  $\ln(y)$  by the dependence  $\ln(f(x_1, \dots, x_n, c_1, \dots, c_m))$ , i.e., find

the coefficients  $c_1, \dots, c_m$  for which the sum  $\sum_{k=1}^K (\Delta Y^{(k)})^2$  is the smallest possible, where  $\Delta Y^{(k)} \stackrel{\text{def}}{=} Y^{(k)} - F(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m)$ ,  $Y^{(k)} \stackrel{\text{def}}{=} \ln(y^{(k)})$  and

$$F(x_1, \dots, x_n, c_1, \dots, c_m) \stackrel{\text{def}}{=} \ln(f(x_1, \dots, x_n, c_1, \dots, c_m)).$$

**Second idea: justification.** Measurement errors are usually relatively small, so we can safely ignore terms which are quadratic (or of higher order) in terms of the corresponding error. For example, if we measure with accuracy 10%, then the square of the corresponding measurement error is about 1%, which is much smaller than 10% and can, therefore, be safely ignored.

In our case, by definition of the approximation error  $\Delta y^{(k)}$ , we have

$$y^{(k)} = f(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m) + \Delta y^{(k)}.$$

So,

$$\ln(y^{(k)}) = \ln\left(f(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m) + \Delta y^{(k)}\right).$$

If we expand this expression in Taylor series in terms of  $\Delta y^{(k)}$  and take into account that the derivative of logarithm  $\ln(x)$  is  $\frac{1}{x}$ , we conclude that

$$\begin{aligned} \ln(y^{(k)}) &= \ln\left(f(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m)\right) + \\ &\quad \frac{1}{f(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m)} \cdot \Delta y^{(k)} + \dots \end{aligned}$$

As we have mentioned, in this expansion, we can safely ignore terms which are quadratic (or of higher order) in  $\Delta y^{(k)}$ , thus

$$\begin{aligned} \ln(y^{(k)}) &= \ln\left(f(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m)\right) + \\ &\quad \frac{1}{f(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m)} \cdot \Delta y^{(k)}. \end{aligned}$$

The expression in the denominator is equal to  $y^{(k)} - \Delta y^{(k)}$ , thus

$$\ln(y^{(k)}) = \ln\left(f(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m)\right) + \frac{1}{y^{(k)} - \Delta y^{(k)}} \cdot \Delta y^{(k)}.$$

Expanding again in terms of  $\Delta y^{(k)}$  and ignoring quadratic and higher order terms, we conclude that

$$\ln(y^{(k)}) = \ln\left(f(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m)\right) + \frac{\Delta y^{(k)}}{y^{(k)}},$$

i.e., by definition of the relative error  $\delta y^{(k)}$ , that

$$\ln(y^{(k)}) - \ln\left(f\left(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m\right)\right) = \delta y^{(k)}.$$

Thus,

$$\begin{aligned} \Delta Y^{(k)} &= Y^{(k)} - F\left(x_1^{(1)}, \dots, x_n^{(k)}, c_1, \dots, c_m\right) = \\ &= \ln(y^{(k)}) - \ln\left(f\left(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m\right)\right) = \delta y^{(k)}, \end{aligned}$$

i.e.,  $\Delta Y^{(k)} = \delta y^{(k)}$ .

So, minimizing the sum  $\sum_{k=1}^K (\Delta Y^{(k)})^2$  is indeed equivalent to the minimization of the mean squares value of the relative error, i.e., to the minimization of the sum  $\sum_{k=1}^K (\delta y^{(k)})^2$ . Hence, we arrive at the following recommendation.

**Second idea: recommendation.** To solve the desired relative-error-minimization problem, we compute the values  $Y^{(k)} \stackrel{\text{def}}{=} \ln(y^{(k)})$  and form a new function  $F(x_1, \dots, x_n, c_1, \dots, c_m) = \ln(f(x_1, \dots, c_1, \dots, c_m))$ .

Then, we apply the absolute-error-minimizing package to find the coefficients  $c_1, \dots, c_m$  for which

$$Y^{(k)} \approx F\left(x_1^{(k)}, \dots, x_n^{(k)}, c_1, \dots, c_m\right)$$

for all  $k = 1, \dots, K$ .

## 4 Acknowledgments

This work was supported in part by the National Science Foundation grant HRD-1242122 (Cyber-ShARE Center of Excellence).

## References

- [1] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
- [2] S. G. Rabinovich, *Measurement Errors and Uncertainty: Theory and Practice*, Springer Verlag, Berlin, 2005.
- [3] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.