Performance of optimization engines: simulations in R and SAS IML/Plus

Alberto Ferrando

November 27, 2015

Abstract

The purpose of this document is to study the performance of various optimization engines and methods using both SAS IML/Plus and R languages. In particular, we will apply them in Maximum Likelihood Estimation for the parameters of different continuous probability distributions, by varying the sample size. Moreover, those optimization methods will be compared among themselves to determine which method generates the most reliable estimates. From such a study emerged, as statistical theory predicts, an increasing relationship between performance and sample size. In addition, the performance difference between the methods analysed is practically null. Nevertheless, the size of the error grew bigger as the distribution parameters were increased.

1 Introduction

Optimization is one of the cardinal procedures used in almost every branch of applied mathematics. Thanks to this importance, many different methods and algorithms have been developed to cope with this need. In statistics, optimization plays a central role into Maximum Likelihood Estimation of a distribution parameters: in fact, given a set of data and an assumed parametric distribution that well describe those data, we are able to estimate the parameters that most likely generated the data. To accomplish this task we simply have to maximise the *likelihood function* defined as:

$$L(\theta|x) = P(x|\theta) \tag{1}$$

where θ is a vector of parameters and x denotes the vector of data.

However, various optimization techniques have been developed, each with its own drawbacks and strengths. For example, different sample sizes and form of the likelihood function are two factors that influence the behaviour of such methods. We will therefore present the reader with the methodology used, the findings generated, all followed by a short discussion.

2 Methodology

The study was conducted using the following optimization methods:

- the Newton-Raphson: a pure Newton method.
- the Nelder-Mead: a *simplex* search method, where a *simplex* is defined as the convex hull of $n + 1$ vertices $x_0, ..., x_1 \in \mathbb{R}^n$ (Nelder and Mead, 1965).
- a quasi-Newton as developed by Byrd *et al.* (1995).
- golden section with successive parabolic interpolations.

Their relevant algorithms implementations in SAS/IML Plus and R have been used by applying the NLPNRA, optim() and optimize() functions: the NLPNRA for both one and two-parameters distributions, while the optimize() and optim() for one and two parameters respectively.

In order to study their performance, they were applied to estimate the MLE parameters of random samples taken from the Exponential, the Normal and the Gamma distributions. Specifically, the Gamma distribution has been parametrized with shape α and rate β as the inverse of the scale parameter. With the estimations over 1000 repetitions, we calculated the Mean Squared Error which has been chosen as a metric for comparison. The MSE is defined as:

$$MSE(\hat{\theta}) = \frac{1}{M} \sum_{m=1}^{M} (\hat{\theta}_m - \theta)^2 \tag{2}$$

where $\hat{\theta}$ is the vector of ML parameter estimates, M is the number of evaluations for that estimate and θ is the vector of the true parameters.

3 Simulations and findings

The first simulation in our study aimed at comparing the performance of optimization methods as a function of sample size. In other words, we compared the MSE of the parameter estimates obtained by applying ML on 1000 samples of a given sample size and generated from a particular distribution. By changing the sample size, we were able to visualize the behaviour of the optimization engines' performance. The chosen number of observations for each round were: 20, 30, 50, 80, 100 and 200.

Before presenting the results, we have to specify that all values given have been rounded to the 3$^{\text{rd}}$ decimal place for sake of clarity. Nevertheless, the true values generated by the simulations were characterised by practically insignificant differences in the order of 10^{-5}. For this reason, the figures presented will, in most of the cases, appear equal within the different optimization algorithms.

We first concentrated on the exponential distribution with a rate parameter λ of 1. In this case, we used the optimise() algorithm in R which is specifically developed for single-dimensional problems. In SAS IML/Plus on the contrary, we used the NLPNRA subroutine, which can handle both single and multi-dimension optimization. The MSE values for this distribution are summarized in Table 3.

We then focused on the Normal distribution with mean μ of 2 and standard deviation σ of 3. In this case, given the two dimensional optimization problem, the MSE for the parameters is a vector of two components: one for each parameter. For the optim() function in R, we chose to apply the L-BFGS-B method. The results for μ are presented in Table 4, while for σ in Table 5.

Finally, we decided to apply the same rationale on the Gamma distribution, defined with a shape parameter α of 2 and a rate parameter β of 0.5. Also in this case the MSE for the parameter estimates becomes a vector of two elements. Moreover, as for the Normal above we chose to apply the L-BFGS-B method for the optimization engine in R. The relevant findings are given in Table 6 for α and in Table 7 for β.

The second simulation in our study was conceived to compare the performance of optimization methods within themselves. To accomplish this we kept fixed the sample attributes while changing the optimization method. Specifically, the simulation was performed on both the Normal and the Gamma distribution using 1000 repetitions. The two algorithms compared were the Nelder-Mead and the L-BFGS-B. The comparing metric in this case was defined as the difference in MSEs. Formally:

$$\Delta_i = MSE_{i,1} - MSE_{i,2} \qquad i = 1, 2 \tag{3}$$

where the subscript i refers to the parameter, the subscript 1 to L-BFGS-B and the subscript 2 to Nelder-Mead.

By running the simulation we did not find any practically significant difference in estimate accuracy: in the Normal distribution case Δ_μ was $-5.65 * 10^{-6}$ and Δ_σ was $-1.1 * 10^{-5}$. On the contrary, for the Gamma distribution we got opposite results with a Δ_α of $1.53 * 10^{-5}$ and a Δ_β of $4.73 * 10^{-6}$. Nevertheless, we also calculated the average number of function evaluations that each method took until convergence: the findings are summarized in Table 1

	Normal	Gamma
L-BFGS-B	9.26	21.18
Nelder-Mead	55.23	61.90

Table 1: Mean number of objective function evaluations for two optimization methods

Finally we also investigated how those optimization engines would perform as the distribution parameter were changed. To investigate this, we computed 1000 simulations on the MLE for the Exponential distribution parameters, by keeping the sample size fixed

but by increasing the λ. We present the findings for the Exponential distribution in Table 2. Similar results were obtained also with the other two distributions.

λ	1	3	5	7	9	10
NLPNRA (SAS)	0.038	0.339	0.943	1.848	3.055	3.772
optimize (R)	0.038	0.309	0.782	1.455	1.748	1.234

Table 2: MSE of λ of the Exponential for different parameter values

4 Discussion of findings

From the first simulation above a clear trend emerges: the estimation accuracy of the optimizations increases as the sample size increases. In other words, there exist a negative relationship between the chosen metric MSE and sample size. This is well pictured in Graph 1, which is constructed only for the Exponential distribution. However, we can clearly see that this relationship is also valid for the other two distributions. An other important finding relates to the very small size of the MSE error even at the smallest sample which provides proof of the reliability of such optimization engines.

Nevertheless, some odd features appear from the data. From the two simulations on the Normal and the Gamma distributions we see that the MSE for the one of the parameters is constantly greater than the corresponding metric value for the other parameter (see Tables 4 and 5). For example, in the Normal distribution case, the smallest MSE reached for μ is 0.042 while for σ is 0.023.

Moreover, from the data is also clear that both NLPNRA and optim() or optimize() have the same performance: no practically significant difference has been found by comparing the functions. The only visible difference appears in the Exponential case, where the NLPNRA subroutine was less precise, at a sample size of 20, of $2 * 10^{-3}$ if compared to its counterpart (see Table 3). The authors believe that this is clearly not a real issue, given the very small nature of the discrepancy.

From the second simulation no clear difference emerged between the Nelder-Mead and the L-BFGS-B methods: with the Normal, the latter seemed more precise (the $\Delta < 0$) while with the Gamma the result was the opposite ($\Delta > 0$). Therefore we can conclude that, with respect to precision, no real difference exists between those two methods. Nonetheless, as demonstrated by Table 1, the two methods differ significantly in terms of function evaluations, with the L-BFGS-B method being more efficient by requiring less evaluations.

Finally, from the last simulation we also found an interesting relationship existing between the absolute size of the parameter and the ML estimation error. In particular, as it emerges from Table 2, the grater the parameter value, the greater the MSE. Similar findings were also obtained by increasing the parameters for the other two distributions.

At the same time we can also state that the NLPNRA subroutine loses more precision than the optimize() function as the parameter values increase, given that, at $\lambda = 10$, the MSE was 3.772 versus 1.234 respectively.

Sample size	20	30	50	80	100	200
NLPNRA (SAS)	0.062	0.038	0.022	0.014	0.01	0.005
optimize (R)	0.06	0.038	0.022	0.014	0.01	0.005

Table 3: MSE of λ of the Exponential for different sample sizes using SAS IML and R

Sample size	20	30	50	80	100	200
NLPNRA (SAS)	0.433	0.288	0.187	0.118	0.091	0.042
optim (R)	0.433	0.288	0.187	0.118	0.091	0.042

Table 4: MSE of μ of a Normal for different sample sizes using SAS IML and R

Sample size	20	30	50	80	100	200
NLPNRA (SAS)	0.225	0.143	0.086	0.058	0.043	0.023
optim (R)	0.225	0.143	0.086	0.058	0.043	0.023

Table 5: MSE of σ of a Normal for different sample sizes using SAS IML and R

Sample size	20	30	50	80	100	200
NLPNRA (SAS)	0.696	0.423	0.193	0.105	0.084	0.035
optim (R)	0.696	0.423	0.193	0.105	0.084	0.035

Table 6: MSE of α of a Gamma for different sample sizes using SAS IML and R

Sample size	20	30	50	80	100	200
NLPNRA (SAS)	0.057	0.034	0.015	0.008	0.007	0.003
optim (R)	0.057	0.034	0.015	0.008	0.007	0.003

Table 7: MSE of β of a Gamma for different sample sizes using SAS IML and R

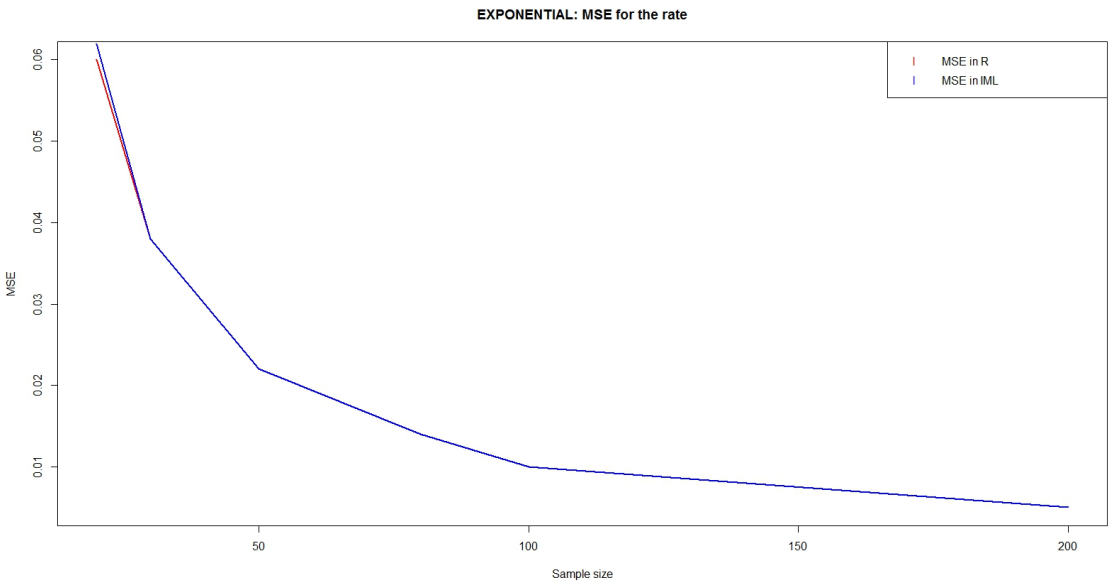

Figure 1: MSE as a function of sample size. Simulation for the Exponential

References

[1] Zhu C, Byrd R.H, Lu P, Nocedal J. (1995), A Limited Memory Algorithm for Bound Constrained Optimization, *IMA J Appl Math*, 6 (1): 76-90.

[2] Nelder J. A, Mead R. (1965), A Simplex Method for Function Minimization, *SIAM Journal on Scientific Computing*, SIAM