BOOTSTRAP SIMULATION MODELS FOR INDIRECT MEASUREMENTS ESTIMATION

MIŞCOI GHEORGHE
Hab., Ph.D. Prof., Member of the Academy of Science of Moldova

IOAN RODICA
Faculty of Mathematics and Informatics
Spiru Haret University
r.ioan.mi@spiruharet.ro

ALBEANU GRIGORE
Faculty of Mathematics and Informatics
Spiru Haret University
g.albeanu.mi@spiruharet.ro

Abstract

This paper presents different bootstrap simulation models to be applied for some indirect measurement problems. These approaches are discussed and applied for uncertainty measurement in proficiency testing, time series, and reliability modeling. From computational point of view, the generalized inverse is required when using Quasi Gauss Newton Algorithms for solving nonlinear systems of equations.

Keywords: *bootstrap, Quasi Gauss Newton, generalized inverse*

AMS classification: 62P20

1. Introduction

Statistical methods can be used not only to summarize or describe a collection of data, but also for data modeling in order to identify predictive and inferential aided decisions in various fields. The basic assumption when using any statistical techniques for a data collection is that the data under analyze represent only a sample, sometimes a small sample, of the possible observations, so the predictive / inferential result based only on the sample will be used to predict the result for the population.

Bootstrapping technique, as a way to survive, proved important advantages for small-size samples and in cases when any information about data distribution is missing.

In this paper the authors describe the application of the bootstrap way of thinking for assessing the accuracy when direct or indirect measurements are required for some item. The purpose of the paper is not only methodological. Some types of bootstrapping schemes are presented and applications are considered.

The material is organized as following. The next section presents the mathematical model of direct / indirect measurement process to be study by bootstrapping. The third section is dedicated to bootstrapping algorithms, while different simulation models are described in the fourth section. Finally, some remarks are considered in the concluding section.

2. **Direct and Indirect Measurements**

Direct and indirect measurements are very important in physics, chemistry and biology. Measuring the wind's speed can be done indirectly while the water phase coefficient of permeability is better to be measured directly. The blood pressure can be measured using both direct and indirect approaches with different advantages and disadvantages. Indirect measurement of inner temperature of materials processed by metallurgy and machine industry is necessary. There is a wide range of sensors for surface temperature measurement; however there is no detector which could be placed inside of a material and be able to measure the inner temperature during the heat process in real time. For some cases the measurement is only an approximation; we mention here the estimated number of bugs in some software.

In the indirect measurement is necessary to have, previously, the knowledge of some quantities obtained by direct measurements and, starting from the relationship among those quantities and that one needed to know, it can be obtained the value of the unknown quantity, by solving an equation (polynomial, integral, differential, etc.).

To measure directly the item A consists of measuring exactly some attribute u of A that we are looking to measure. However the indirect measurement of the same attribute u is based on idea that we will find the measure of the attribute u by measuring a different attribute v of an item B. Generally speaking B is a model of A and v is a parameter to be estimated, using different paradigms, and used as a measure of u. A special case appears when also A is a model of a real item. Direct measurement can be applied in laboratory or in situ. Indirect measurement requires some tools, sometimes consisting of models, theoretical results, validation methods, hardware, and simulation software.

Indirect measurements often require the solution of inverse problems, which can be ill-posed or ill-conditioned. Indirect measurement system can be formalized, in general, in a nonlinear regression framework (sometimes, the linear case is appropriate). For such cases we can use the bootstrap to analyze the behavior of the estimation procedure by resampling the environment. In this way, an uncertainty measure can be also obtained.

The model B may belong to linear class or to the nonlinear class. A special class of nonlinear models deals with linear combinations of nonlinear basis functions, including trigonometric shapes.

The measured signal from a linear system, along a scanning parameter t, can be presented in the following form:

$$y(t)=\sum_{k=1}^{m}\alpha_k\phi_k(t), \tag{1}$$

where α_k is an unknown parameter to approximate the measure of some attribute, and $\phi_k(t)$ is the submodel describing an additive component of the output. If the continuous case is considered, the form of the measured signal is given by:

$$y(t)=\int_a^b \alpha(x)\Phi(t,x)dx,\ x\in[c,d], \qquad (2)$$

with the unknown $\alpha(\cdot)$.

Based on (1), if the behaviour of the system A is known for a set of inputs $t_1,t_2,\ldots,t_n$ the unknowns $\alpha_1,\alpha_2,\ldots,\alpha_m$ can be obtained as the solution of the system

$$\Phi\,\alpha = Y, \qquad (3)$$

with

$$\Phi_{ij}=\Phi_j(t_i),\ 1\le i\le n,\ 1\le j\le m,$$

$$Y=(y(t_1),\ldots,y(t_n))^T$$

and

$$\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_m)^T,$$

where the superscript denotes the transpose operation. If the case (2) is considered, also a large linear system of linear equations will be obtained and used to find the function $\alpha(\cdot)$.

A common approach in solving (3) is based on the least square method, namely

$$\alpha = \Phi^{+}Y, \qquad (4)$$

where, V^+ denotes the generalized inverse of the matrix V, α being the minimum Euclidean norm solution [5].

However, the equation (2) describes a homogeneous *Fredholm equation of the first kind*, with $\Phi(t, x)$ being the Kernel. Of course, there are cases when $\alpha(\cdot)$ can be obtained using direct and indirect Fourier transforms.

A special model proved important applications in demography [12], the study of viscoelastic materials [7], and in insurance mathematics through the renewal equation [4]. This is given by:

$$y(t)=\int_a^t \alpha(x)\Phi(t,x)dx, \qquad (5)$$

and called the linear *Volterra equation of the first kind*, that can be solved in some cases using Laplace transform. For "heavy" cases, other methods have to be used, including those described in [12] and [7] to mention only few references.

In general we assume that $y(t)=(Kf)(t)+\varepsilon(t)$, in order to capture imprecise measurement aspects and other sources of noise. From statistical point of view we assume that ε is a random variable uniform distributed having zero mean. In this way, for the n measurements, we have

$$y(t_i)=(Kf)(t_i)+\varepsilon_i, \qquad (6)$$

where $1\le i\le n$.

For the above models K is an integral operator. However, the form given in (6) is a general one suitable for regression operators, differential operators, moving average operators (MA), autoregressive operators (AR), etc. In all cases, f is the unknown to be obtained in order to explain the behaviour of the system under study.

In the next section we consider some bootstrap algorithms and their application to study the behaviour of some particular models for indirect measurement.

3. **Bootstrap Algorithms**

The methods used by researchers are based on imposed assumptions, quite often unverifiable, about the data to be analyzed. In most cases these requirements are not valid making the obtained parameters and related quantities unreliable. The researcher can use bootstrap or other methods (like jackknife, cross validation, etc.) as a tool to eliminate the unreliability generated by the incorrect or incomplete use of theoretical assumptions. The bootstrap method was introduced by Efron [9], in 1979, as a computer-based technique that has become very popular in recent years for estimating such things as standard errors, confidence intervals, biases, and prediction errors. The literature shows various applications of this powerful Monte-Carlo method in assessing statistical accuracy or to estimate different models from data. Without reviewing such developments we mention the references [1, 6, 8, 10] and [14] for time series analysis and applications to risk management, [2, 3, 9] and [11] for the estimation of distribution functions. The method is applicable to samples of different sizes, including to small sets of data [15].

The bootstrap replication can be realized in various ways [9]. The weighted replication, called also, *the importance resampling algorithm*, assumes the generation of sampling values according to a probability distribution $\{(\varepsilon_i, p_i): i = 1, 2, ..., n\}$ such as every p_i is a nonnegative real number, and $p_1 + p_2 + ... + p_n = 1$. When using $p_i = i/n$, i = 1, 2, …, n, the standard bootstrap method is obtained. The weighted replication is justified by the importance of some measurements with increasing accuracy during the measurement process.

Using a method to generate random samples according to a given discrete distribution function, the weighted bootstrap sample $(\varepsilon_1^*, \varepsilon_2^*, ..., \varepsilon_n^*)$ is obtained according to the following steps:

1. Initialise a uniform random number generator *rand*() over (0, 1);
2. For k from 1 to n do the steps (a) to (d).
   (a) Let j = 0, and u = *rand*();
   (b) Increment j by 1;
   (c) If $u > \sum_{i=1}^{j} p_i$ then goto step (b);
   (d) Choose $\varepsilon_k^* = \varepsilon_j$.

The above approach is based on the inverse method for nonuniform discrete random simulation.

In some cases, when stratified sampling is used, the best way of bootstrapping is multistage based. In stratified sampling the population is divided, firstly, in subpopulations (according to considerations depending on the application). Then, for each subpopulation a sample is drawn in an independent manner. The researcher will hope to improve the precision, cover some local aspects, or to improve the data management during statistical analysis.

Block bootstrap will be used when time series with seasonal components, a block corresponding to a segment of the time series.

In the following the application of bootstrap simulation to deal with uncertainty of indirect measurements in proficiency testing, software reliability and time series modeling is considered. In a similar way, the bootstrap algorithms can be used to analyze the behavior of the numerical methods used in solving the Volterra renewal equation [4].

4. **Robust Z-scores in proficiency testing**

In order to apply the bootstrap algorithms for proficiency testing in cement industry, let us introduce some basic definitions and notations [2]. For a set (sample) of uncorrelated N random variables, $x_i, i = 1, 2, ..., N$, let us denote by Mean[X] the sample *mean* given by:

$$Mean[X] = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

The *median* is the middle value of the group for a particular sample, i.e. half of the results for the sample are higher than it and half are lower (Q2). It is calculated from the sorted values (from lowest to highest). If N is even, the median is the average of the two central values. Let us denote this value by Median[X].

Interquartile range (IQR[X]) is the difference between the lower and upper quartiles = Q3 - Q1. The lower quartile is the value below a quarter of the results lie. Similarly, the upper quartile is the value above a quarter of the results lie.

Normalised IQR (NIQR[X]) is a measure of the variability of the results which basically is a robust standard deviation. It is equal to the IQR[X] multiplied by the factor 0.7413.

Robust CV (coefficient of variation) is equal to the NIQR[X] divided by the median, expressed as a percentage (i.e. multiplied by 100), it allows for the variability in different tests to be compared.

An accepted statistical method for analysis test results in proficiency testing is to calculate a Z-score for each laboratory's result. The standard form for the calculation of Z-scores is

$$Z_i = \frac{X_i - A[X]}{B[X]}$$

where A[X] is the assigned value (sample mean), and B[X] is an estimate of the spread of all results (standard deviation). The classical approach based on mean and standard deviation is significantly influenced by the presence of extreme values (outliers). Therefore, a robust approach based on median and interquantile range is better to be used [2].

Robust Z-scores are calculated by replacing A[X] and B[X] in the "classical" Z-score by the median and NIQR, respectively. For proficiency testing both between-laboratory and within laboratory Z-scores can be used.

The standardized sum (S) and the standardized difference (D) for the pair of results are: S = (A+B)/ *sqrt*(2) and D = (A-B)/ *sqrt*(2) (median of A is less than median of B) or D = (B-A)/ *sqrt*(2) , otherwise.

The between-laboratory Z-score (ZB) is the robust Z-score of S and the within-laboratory Z-score (ZW) is the robust Z-score of D:

ZB = (S-*Median*[S])/NIQR(S),

and

ZW = (D - *Median*(D))/NIQR(D).

A methodology based on bootstrap approach can be used to study the robust Z-scores ZB and ZW.

A Z-score can be computed repeatedly by simultaneous resampling in order to obtain a resample mean and standard deviance.

A robust Z-score can be computed repeatedly by simultaneous resampling in order to obtain a resample median and normalized interquantile range.

The process is repeated by a number of steps and the final results can be obtained considering the best performance.

An 80% approach can be used when a strong acceptance is required. Generally speaking, the best test could be based on a 51% approach.

5. **Reliability modelling**

In this part the bootstrap method is applied on software reliability models with important benefits for Software Reliability Engineering, and for Total Quality Software Engineering Management. Firstly, some terminology will be introduced. A counting process, N(t) with N(0)=0, is said to be a nonhomogeneous Poisson Process (NHPP), with intensity function λ if:

- The failure process has an *independent increment*, i.e., the number of failures during the time interval (t, t +Δt] depends on the current time t and the length of time interval Δt, and does not depend on the past history of the process.
- The failure rate of the process is given by: P{exactly one failure in (t, t + Δt)} = P{N(t + Δt) - N(t) = 1}= λ(t) Δt + o(Δt).
- During a small interval Δt, the probability of more than one failure is negligible, that is: P{ two or more failures in (t, t + Δt) }= o(Δt).

It follows that

$$P\{N(t)=n\}=\left[m(t)\right]^n/n!\exp\left(-m(t)\right), n\geq 0,$$

where

$$m(t)=E\left[N(t)\right]=\int_0^t \lambda(s)ds$$

is the expected number of errors detected by time t (the "Mean Value Function", or MVF). The above intensity function is also called "hazard rate" or ROCOF (Rate of occurrence of failures). The template exp(t) stands for the very known function e^t (e = 2.718...).

Let us denote by a(t) the total number of errors in the software product including the initial and introduced errors at time t, impossible to be measured directly, and by b(t) the failure detection rate.

The generalized form of the MVF is a solution of the equation:

$$m(t)+b(t)m(t)=a(t)b(t); m(0)=m_0, \quad (7)$$

and is given by [13]:

$$m(t) = \exp(-B(t))\left[m_0 + \int_{t_0}^{t} a(u)b(u)\exp(B(u))du\right], \tag{8}$$

where

$$B(t) = \int_{t_0}^{t} b(u)du,$$

and t_0 is the time to start the debugging process ($m(t_0) = m_0$).

The reliability R(t), defined as the probability that there are no failures in the time interval (0, t), has the expression:

$$R(t) = PN(t) = 0 = \exp(-m(t)). \tag{9}$$

In general, the reliability R(x | t), the probability that there are no failures in the interval (t, t + x), is given by:

$$R(x \mid t) = PN(t + x) - N(t) = 0 = \exp(-[m(t+x)-m(t)]), \tag{10}$$

having the density function according to:

$$f(x) = \lambda(t + x) \exp(-[m(t+x)-m(t)]). \tag{11}$$

Different assumptions on a(t) and b(t) give rise to particular NHPP software reliability models. For example, if

$$a(t) = \alpha_1$$

and

$$b(t) = \frac{\alpha_2}{[1 + \beta \exp(-\alpha_2 t)]},$$

the Inflection S-shaped NHPP model is obtained (with α_1 - the expected total number of faults that exist in the software before testing (imposible to measure directly), α_2 - the failure detection rate, and β - the inflection factor). Other NHPP models (as Duane, Goel - Okumoto, Musa - Okumoto, Littlewood etc.) start with an assumption on the ROCOF function.

No matter, MVF or ROCOF assumption, the unknown parameters have to be estimated. The mostly used approach is the maximum likelihood method applied to the likelihood function:

$$L(t) = \prod_{k=1}^{n} f_k(t), \tag{12}$$

with f_i being the probability density function after seeing the inter-failure times: $X_1, X_2, ..., X_{i-1}$.

If X_k is the time interval between the (k-1)st and kth failure, and T_k is the random variable of the occurring time for the kth failure, then $X_k = T_k - T_{k-1}, k \geq 1$ with $T_0 = 0$. The probability density function of T_k, according to [13], is given by:

$$f_{T_k}(t) = \frac{\lambda(t) e^{-m(t)} [m(t)]^{k-1}}{(k-1)!} \tag{13}$$

with

$$\lambda(t) = \frac{d}{dt} m(t). \tag{14}$$

Assuming that $m_0 = 0$, m(t) is a strictly increasing function and uniformly continuous on any closed interval, and differentiable then:

$$E^*[T_k] = \frac{\int_0^a m^{-1}(z) z^{k-1} e^{-z} dz}{\int_0^a z^{k-1} e^{-z} dz}, \qquad (15)$$

where m(∞) = a, and $E^*[T_k] = E[T_k \mid T_k < \infty]$ is the conditional expectation.

If the expectation of T_k is available, the mean time between failures (MTBF) are given by

$$E^*[X_k] = E^*[T_k] - E^*[T_{k-1}], \qquad (16)$$

However, it is difficult to derive the inverse value function analytically for the most mean value functions. When inversion is possible, a very accurate estimation of the functional m is required. Hence, an accuracy assessment of the estimation is mandatory, and the bootstrap approach can be used, as described in the following.

Let us suppose that n parameters have to be estimated. Solving the maximum likelihood equation requires, in general, a non-linear system of n equations in n unknowns, written as:

$$\left(h_1(\theta), h_2(\theta), ..., h_n(\theta)\right) = (0, 0, ..., 0), \qquad (17)$$

with $\theta = (\theta_1, \theta_2, ..., \theta_n)$ the n-dimensional vector of unknowns.

Using a numerical method, for instance a Quasi Gauss-Newton method [5], we can obtain the solution θ^*. Then we calculate the residue vector:

$$r_i = h_i(\theta)(i = 1, 2, ..., n),$$

and an estimation of the cdf F by

$$F_n = \left\{\left(r_i, \frac{1}{n}\right) \mid i = 1, 21, ..., n\right\}.$$

At step i (i ≥1) we generate the bootstrap sample $r_1^*, r_2^*, ..., r_n^*$ using resampling from $(r_1, r_2, ..., r_n)$. Then we solve, using the same method, the system

$$\left(h_1(\theta) + r_1^*, h_2(\theta) + r_2^*, ..., h_n(\theta) + r_n^*\right) = (0, 0, ..., 0),$$

and the solution is called a *bootstrap estimation* θ_i^* of the parameter .

After B steps, the sequence $\theta_1^*, \theta_2^*, ..., \theta_n^*$ is obtained and can be analyzed by different techniques.

If the variability in the sequence is not very large, the initial solution θ^* may be used in future analysis.

If the MVF model is accurate estimated, a procedure to solve the equation m(u)=z for any given z is required. Then a quadrature formula can be used to obtain the MTBF estimation.

6. Time series modelling and bootstrap

Let us consider the stationary finite order ARMA(p, q) model

$$A(L) y_t = B(L) \varepsilon_t, \qquad (18)$$

where A(L) and B(L) are invertible polynomials in the lag operator used to model the autoregressive, respective the moving average part, with $Y = (y_1, y_2, ..., y_T)'$ denoting the observed data and $\mathrm{var}(y_t) < \infty, \varepsilon_t$ are iid with $E\{\varepsilon_t\} = 0$ and $\mathrm{E}(\varepsilon^2) < \infty$. The bootstrap algorithm for the model (18) is based on the following steps:

1. Establish (p, q) the order of the ARMA process under study and choose a large positive τ in order to deal with the interval from -τ to T.

2. Obtain the estimates of the parameters (the coefficients of the polynomials A(L) and B(L), denoted by A(L), B(L).

3. Compute $\hat{\varepsilon}_t = \hat{B}^{-1}(L)\hat{A}(L)y_t,\ t = 1, 2, ..., T$. Recenter $\hat{\varepsilon}_t$ around zero.

4. Resampling (with replacement) from $\hat{\varepsilon}_t$.

5. Set $y^*_t = 0$ for $t < \tau$ and generate iid draws from ε^*_t for $t = -\tau, ..., T$.

6. Compute $y^*_t = \hat{A}^{-1}(L)\hat{B}(L)\varepsilon^*_t, t = -\tau, ..., T$

7. For the last T values of y^*_t obtain the bootstrap parameter estimates, namely $\hat{A}^*(L), \hat{B}^*(L)$.

8.Repeat the steps (3) to (7) in order to be able to make a good decision about some behaviour of the time series.

When applied for seasonal time series a simple modification to cover the seasonal aspects is necessary [1].

7. **Conclusions**

The above presentation shows the application of the bootstrap investigation methodology for analyzing various aspects like ill-conditionality, imprecision or uncertainty when solving some indirect measurement problems. All the time, it is assumed the availability of some estimation method of model, parameters, etc. The simulation model based on bootstrap method is used only to certify a good or bad behavior of some procedure when the requirements to use the procedure are difficult to check, or are inappropriate.

References

1. Albeanu, G., Madsen, H., Burtschy, B., Popențiu-Vlădicescu, Fl., Ghica, M., *Bootstrapping Time Series With Application to Risk Management*, R & RATA, Electronic Journal of International Group on Reliability, 1(3), 84—93, 2008.
2. Albeanu, G., *On Using Bootstrap Approach for Uncertainty Estimation*. In G. Guslicov (ed.), Proceedings of the 1st International Proficiency Testing Conference, Sinaia (October 11-13, 2007), 221-229, 2007.
3. Albeanu, G., Ghica, M. *Bootstrap Simulation Models for Probabilistic Approaches in Environmental Economics*, Proceedings of the Sixth

International Workshop on Mathematical Modelling of Environmental and Life Science Problems, Constanţa (September 5-9, 2007).
4. Albeanu, G., F. Popenţiu, F., Niculescu, R., *Recent Approaches in Renewal Function Computation*, In (T. Dohi, N. Limnios and S. Osaki, eds.) Proc. of the Second Euro-Japanese Workshop on Stochastic-Risk Modelling for Finance, Insurance, Production and Reliability, Chamonix-France, 518—523, 2002.
5. Albeanu, G., *On the Convergence of the Quasi Gauss Newton Methods for Solving Nonlinear Systems*, International Journal of Computer Mathematics, 66, 93—99, 1998.
6. Berkowitz, J., Kilian, L., *Recent Developments in Bootstrapping Time Series,* Econometric Reviews, 19, 1—48, 2000.
7. Booker, J.R., Small, J.C., *Methods for the Numerical Solution of the Equations of Viscoelasticity*, International Journal for Numerical and Analytical Methods in Geomechanics, 1(2), 139 – 150, 1977.
8. Bühlmann, P., *Bootstraps for Time Series*, Research Report, 87, ETH, Zürich, 1999.
9. Efron, B., Tibshirani, R.J., *An Introduction to the Bootstrap*, Chapman and Hall, New York , 1993.
10. Härdle, W., Horowitz, J. and Kreiss, J.-P., *Bootstrap Methods for Time Series,* International Statist. Review, 71, 435—459, 2003.
11. Pallini, A., *Efficient Bootstrap Estimation of Distribution Functions*, University of Bologna, 1999.
12. Perthame, B., *Introduction to Structured Equations in Biology*, CNA Summer School Lecture Notes, 2008.
13. Pham, H., *Software Reliability*, Springer, Singapore, 2000.
14. Politis, D.N., *The Impact of Bootstrap Methods on Time Series Analysis*, Statistical Science, 18, 219—230, 2003.
15. Scholz, F.W., *The Bootstrap Small Sample Properties*, Technical Report, http://www.stat.washington.edu/fritz/Reports/bootstrap-report.pdf, 2007.