Adriana AGAPIE, PhD
UNESCO Chair of Business Administration
Luiza BĂDIN, PhD
Department of Mathematics
The Bucharest Academy of Economic Studies
***Gh. Mihoc - C. Iacob* Institute of Mathematical Statistics and Applied Mathematics**

CONFIDENCE INTERVALS FOR THE GUESSTIMATION ALGORITHM: A BOOTSTRAP APPROACH

Abstract. *Economists attempting to estimate linear models are frequently restricted due to data scarcity in terms of short time series of data and also of parameter non constancy. In this case, a realistic alternative is often to guess rather than to estimate parameters of such models. An algorithm of repetitive guessing (drawing) parameters from iteratively changing distributions, with the objective of minimizing the squares of ex-post prediction errors, weighted by penalty weights and subject to a learning process, has been recently introduced and sufficient conditions for convergence were theoretically described. In this paper, Repetitive Stochastic Guesstimation (RSG) and Simulated Annealing (SA) are compared for the problem of a linear regression coefficients' estimation, when only small and undersized samples are available. A robust alternative - based on bootstrap confidence intervals - to the RSG is built: Repetitive Stochastic Bootstrapped Guesstimation (RSGBOOT). A Monte Carlo experiment is designed to compare performances of RSG, RSGBOOT and SA. In the second part, confidence intervals for the RSG point estimators are built in a Bayesian framework. Again, a Monte Carlo analysis is conducted in the case of a linear regression equation.*

Keywords: *Repetitive Stochastic Guesstimation, Simulated Annealing, Bootstrapped likelihood, Bayesian inference.*

JEL Classification: C13, C15, C63

1. INTRODUCTION

Optimization problems surface regularly in statistical applications. Multi-modal (eventually non-differentiable) 'cost functions' often appear in empirical econometrics. Several stochastic algorithms replace failure of gradient techniques in determining global extreme. Still, a hidden dependence within the particular

problem in hand between any optimization algorithm and the cost function considered was underlined in the article by Wolpert and Macready (1997). In that paper it is argued that the particulars of the cost function involved are crucial, and blind faith in an algorithm to search effectively across a broad class of problems is not justified. Thus, any results with a certain algorithm are applicable with certain success at the problem class presented. Potential generalizations have to be tested each time. This paper is analyzing the regression problem for small and undersized samples. A classical algorithm is compared with a newly introduced one: Simulated Annealing (SA) versus Repetitive Stochastic Guesstimation (RSG) (see Charmeza, 2002 and Agapie, 1999 for a comparison of SA and RSG). An improved version of RSG is built and compared to the previous two algorithms. In the second part of this paper the inclusion of prior information is put in a Bayesian framework and the attempt of building confidence intervals for the RSG estimators is sketched. Further research directions conclude the paper.

2. REGRESSION PROBLEM FOR SMALL & UNDERSIZED SAMPLES

Even if in almost econometric textbooks the problem of multi-colinearity is emphasized, a similar problem-concerning small sample size is quite often skipped. An extreme situation appears when the number of observations is smaller then the number of parameters to be estimated. In this case the sample estimations for coefficients are not unique. But this extreme case is easy enough to recognize. Problem of small sample size is subtler and yet very serious. It occurs when rank condition is barely satisfied and it is prevalent in empirical economics.

The consequences of small sample size problems are serious. Precision of estimation is reduced. There are two aspects of this reduction: estimates may have large errors and will be very sensitive to sample data - in the sense that addition of few more observations can sometimes produce drastic shifts in the estimates. Problem of undersized samples arises when the rank condition is not satisfied due to the insufficient number of observations. Still, additional information about the values of the parameters to be estimated can allow for restricting the number of solutions to the unique one. Thus, considering the above aspects, estimation problem of the simplest, classical regression equation in the presence of undersized samples is dragged to the field of global optimization. In the next sections we will turn our attention to a relatively classical algorithm for global optimization (RSG) and compare it to the SA algorithm.

3. BRIEF REVIEW OF REPETITIVE STOCHASTIC GUESSTIMATION AND SIMULATED ANNEALING

In 1996 Charemza proposed a stochastic sampling method for dealing with non-linear optimization tasks as well as with small sample size problems**.** A detailed description of this new technique – called Repetitive Stochastique Guesstimation (RSG) is given in Charemza (2002). Sufficient conditions for convergence are

derived in Agapie (2009). This algorithm is very interesting since it contains different ideas from both global optimization field and econometric practice, and is placed on the boundary between classical and Bayesian econometrics.
Three points are to be stressed:

- At the initial stage RSG makes use of the prior beliefs concerning the parameters to be estimated (according to the economist's expertise and intuition)
- RSG is successively restricting the search space from one iteration to another, providing an asymptotic convergence of the algorithm in some extreme point
- RSG uses two objective functions, instead of a single one.

A straight implication of the first point is that there is dependence between the initial points and intervals considered and the RSG estimates. This dependence will be explored in the following sections. On the other hand, these prior beliefs about the initial values of the parameters and intervals can be quantified and analyzed in a Bayesian framework.

The idea of running a searching algorithm from some expected valued for the parameters, according to experts intuition also appeared in Marcet's (1991) method of parameterized expectations. Concerning the second point, this restriction on search area retrieves the common sense expectation of "*increasing the guesstimator's confidence by narrowing the interval from which the parameters are to be guessed, as time goes on"* (Charemza, 1996). On the other hand, the brute technique is similar to a search technique called *'Fibonacci search'*, first developed by Kiefer (1953). Fundamental differences are at the practical functions used for decreasing the successive intervals. The *Fibonacci search* is essentially deterministic and thus the intervals of uncertainty are governed by some difference equations. In the case of RSG algorithm, the intervals of uncertainty are probabilistic since they are dependent to a probabilistic-weighted objective function.

Regarding the two criterions mentioned at the third point above, namely the non-weighted and weighted objective functions, the (penalty) weights in the last one are normally distributed according to the difference between the currently guess and the previous best guess. This makes RSG a dynamical optimization method, by making the objective function time dependent. Against the class of least-squares learning technique, RSG has the great advantage of being very easy to manipulate and able to deal with the undersized sample problems. SA can be traced to *Metropolis algorithm,* which attempts to simulate the behavior of an ensemble of atoms in equilibrium at a given temperature. Econometric applications of this algorithm are surveyed in Goffe, Ferrier and Rogers (1994). There are two common points with RSG: both algorithms make random extractions (by uniform probability distribution, commonly) in order to achieve new candidate solutions and they both depend on a strictly decreasing parameter (called temperature-in SA, and learning rate-in RSG). The major difference with SA is the fact that the

acceptance criterion is deterministic, in the sense that there is no chance for a 'worst' value to be accepted like the current solution.

4. IMPROVING ON THE RSG ALGORITHM: A BOOTSTRAP APPROACH

A method for constructing likelihood based confidence regions for a vector parameter using the bootstrap and non-parametric density estimation is applied. The technique and its theoretical properties are described in Hall (1987).

In our case, RSG replications for a given iteration are regarded as bootstrap selections from the current coefficient's distribution; a bootstrap-t confidence interval is computed and centered at the best parameter value computed at the previous iteration. At the next iteration, exploring the existent confidence interval computed at the previous iteration does both searching for better coefficient values and constructing confidence intervals. With this approach we make clear the trade between the number of iterations and replications; in this case is preferable a small number of iterations and a large number of replications. Technical details are given below.

Suppose we have an initial value (starting value) for a coefficient, say $C_i^{initial}$ and lower and upper values for it: C_i^L , C_i^U . At iteration *k*, we extract some uniform '*number of replications*' potential coefficients inside the interval (C_i^L , C_i^U): $C_i^1 ,...C_i^{nrepl}$.

With these values we compute *the sample variance*

$$\sigma^2 = \frac{1}{nrepl} \sum_{k=1}^{nrepl} (C_i^k - C_i^{CAPI})^2$$, where C_i^{CAPI} stands for the *i* component of the vector-coefficient *C* computed at the previous iteration (*CAPI*), and the estimated standardized skewness:

$$\gamma = \frac{1}{nrepl} \sum_{k=1}^{nrepl} \left(\frac{C_i^k - C_i^{CAPI}}{\sigma}\right)^3$$

Also, let us denote by C_i^{BACI} (the best at current iteration-*BACI* - *i* component of the vector coefficient *C*) that value among $C_i^1 ,...C_i^{nrepl}$, for which both non-weighted and weighted criterion functions attain minimum.

The corresponding lower and upper values for the next iteration are:

$$C_i^L = C_i^{BACI} - \frac{\sqrt{\sigma}}{\sqrt{nrepl}}\left[1.96 + \frac{1}{6 * nrepl}\gamma(1.96^2 - 3) + \frac{1}{nrepl}\right]$$

$$C_i^U = C_i^{BACI} - \frac{\sqrt{\sigma}}{\sqrt{nrepl}}[-1.96 + \frac{1}{6 * nrepl}\gamma(1.96^2 - 3) + \frac{1}{nrepl}]$$

This modified version of RSG algorithm will be denoted RSGBOOT.

5. COMPARING THE METHODS: A MONTE CARLO APPROACH AND NUMERICAL RESULTS

Data generating process

Consider the following linear regression equation:

$\boldsymbol{Y} = \boldsymbol{X}\,\boldsymbol{C} + \mathcal{E}$ where:

$\boldsymbol{X}$ is a fixed in repetitive samples *(n, p)* matrix of explanatory variables generated from *U(0,1)*;
$\boldsymbol{C}$ is the *(p, 1)* vector of 'true' coefficients, generated once (at the beginning) from *U(0,1)*;
$\mathcal{E}$ is an *(n, 1)* error-vector, sampled for each Monte Carlo replication from *N(0,1)*.

For each Monte Carlo replication, we compute *Y* and estimate coefficients using RSG, RSGBOOT and SA algorithms. The number of coefficients is varied from *1* to *41* and the number of observations considered is $n = 21$. So, the degrees of freedom vary from *20* to *–20*. Since RSG, RSGBOOT and SA estimates are in fact dependent on the initial parameters and intervals a grid exploration for this dependence was performed. 10, 30, 50, 70 and 90% successively altered coefficients' starting values. For each parameter's initial value successive starting intervals considered that we could be wrong by *111, 145, 200, 333* and *1000%*. Summing up, for each initial value and interval considered, one hundred Monte Carlo replications were performed for successively estimate *1* to *41* coefficients. Based on the results obtained after completing all Monte Carlo replications, average bias and average root mean square errors were computed. The number of RSG and RSGBOOT drift changes was *20* and, within one iteration, *50* replications were performed. In order to have a comparison between RSG and RSGBOOT one may built a *0-1* bias-score matrix with *25* lines - number of pairs *(initial parameter value, initial interval)* - and *41* columns (number of degrees of freedom). One on a certain position of the bias-score matrix means that the RSGBOOT 's average bias is smaller than RSG' s average bias. A similar *0-1* score matrix is considered for the root mean square error (rmse). A graphical representation for the two score-matrixes (*bias* and *rmse*) is given in Figures 1-2.

Figure 1. Score-bias matrix

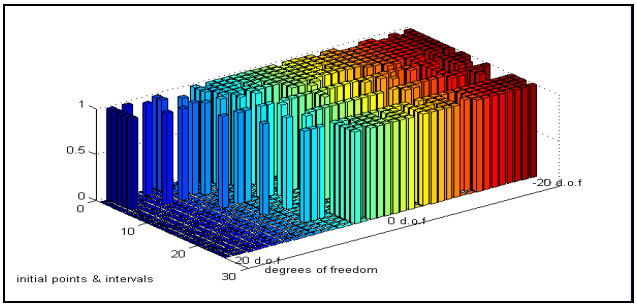
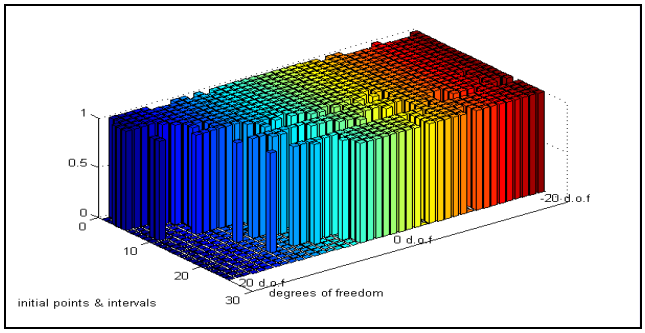


Figure 2. Score-rmse matrix



On the Y-axis points from 1 to 25 give staring parameter values and intervals. If we denote with C^{true} the true value of a certain coefficient C, then the starting point for SA, RSG and RSGBOOT is:

$$C^{initial} = C^{true} + C^{true} \times errat \times (rrr > 0.5) - C^{true} \times errat \times (rrr < 0.5)$$

where $(rrr > 0.5)$ is one if a random uniform number rrr is higher then 0.5 and zero otherwise. The parameter *'errat'* has the interpretation of *'percentage alteration of the coefficient's true value'*. The initial interval for the $C^{initial}$ coefficient to be estimated is given by $(C^{initial} - ll;\ C^{initial} + ll);\ ll = abs(C^{initial})/llu$.

For example, if number *1* on the Y-axis is associated to the pair (*errat = 0.1, llu = 0.1*) that means that the initial value of the coefficient was altered with 10% and we considered that we could be wrong with 1000%. Now, *1* corresponds to the pair (*errat = 0.1, llu = 0.1*), *2* to (*errat = 0.1, llu = 0.3*), *5* to (*errat = 0.1, ll = 0.9*), *6* to (*errat = 0.3,llu = 0.1*) and so on, up to *25*, which corresponds to (*errat = 0.9, llu = 0.9*).

Looking at the graphical representation for the *score_bias* matrix it can be noticed that RSGBOOT's biases are in general smaller then RSG's when one deals

with high alteration for the initial values, large intervals and small to negative degrees of freedom. The second graph, the *score_rmse* matrix, underlines the fact that in general RSGBOOT's root mean square errors are smaller. In conclusion, RSGBOOT is a sort of 'robust' alternative, to be used in the case of small and undersized samples and when high degree of uncertainty governs the choice for initial values and intervals.

When comparison against SA algorithm is performed, in all situations average bias is smaller than both RSG's and RSGBOOT's. Yet, differences are not flagrant. When positive to higher degrees of freedom are considered, the closer to SA is RSG. For negative degrees of freedom, RSGBOOT is the closest to SA. If root mean square error is the comparison criterion, RSGBOOT is comparable or even better than SA. This fact sustains the robustness of the bootstrapped version for RSG algorithm.

Results concerning average bias and average rmse's are depicted in Figures 3-4. Initial coefficients values were altered with 90% and starting intervals considered that we could be wrong by 1000%.

Figure 3. **Average Bias**

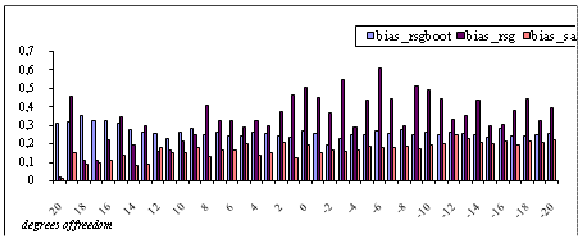
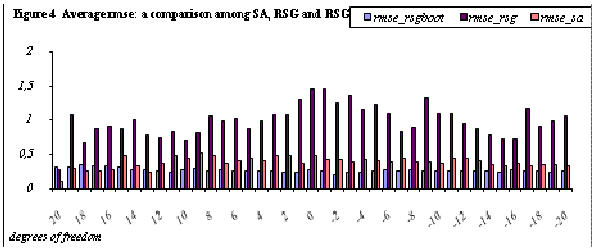


Figure 4. **Root Mean Square Errors**



6. BAYESIAN ANALYSIS FOR THE RSG ALGORITHM

The scope of this section is to analyze, on a regression equation example, to what extent considering initial values to be estimated by the RSG algorithm can be compared to a Bayesian approach. If priors put on coefficients for RSG estimations can be assimilated with Bayesian priors, then it is possible to compute posterior

confidence intervals. A Monte Carlo experiment was performed to measure the adequacy of these intervals. It was counted the proportion of times that a posterior high probability region contains the true value of the parameter for repeated realizations of the posterior density. Conclusion is that priors on coefficients to be estimated with the use of RSG algorithm have a bayesian meaning just when we have reasonably degrees of freedom, higher or equal to eight. This result comes in the line with the observation that RSG estimates are consistent when degrees of freedom are increasing.

6.1. The Bayesian framework

Additional information on the parameters of a model can be quantified in a form of a distribution function named '*a-priori distribution function*'.

If a likelihood function *L (data/parameters)* is available then '*a-posteriori distribution*' of the parameters can be computed: f *(parameter/data)* $\propto$ f *(parameter) L (data/parameter)*

[according to usual operations with pdf's :

f *(data, parameter) =* L *(data/parameter)* f *(parameter)*

$= f$ *(parameter/data)* f *(data)*

$\Rightarrow$

f *(parameter/data)* $= \frac{f(parameter)\,L(data/parameter)}{f(data)}$]

This '*a-posteriori distribution'* of the parameters allow us to make inferences among them, which incorporates the effect of the additional information.

6.2 Bootstrap Likelihood

Serious data analysis requires serious considerations of the effect of model assumptions. In classical statistical theory it is usually only possible to construct likelihood-based regions when we have considerable information about the distribution of our statistic. When problem of estimating unknown distributions appears, bootstrap techniques might be considered (see *e. g.* Bădin, 2007).

The Bayesian bootstrap is the Bayesian analogue of the bootstrap. Instead of simulating the sampling distribution of a statistic estimating a parameter, the Bayesian bootstrap simulates the posterior distribution of the parameter; operationally and inferentially the methods are quite similar. Because both methods of drawing inferences are based on somewhat peculiar model assumptions, neither method should be applied without some consideration of the reasonable of these model assumptions. Neither the bootstrap nor the Bayesian bootstrap presents a general panacea for avoiding sensitivity to model assumptions.

In this section is performed an empirical application for the theoretical results given by Dennis Boss and John Monahan in the paper "Bootstrap methods using prior information". This is accomplished by replacing the likelihood in Bayes's formula by a bootstrap estimate of the sampling density of a robust estimator.

If $Y_1,\dots Y_n$ are independent normal deviates with mean C then the likelihood for C can be expressed in terms of the (sufficient) statistic $\overline{Y}$.
If the number of observations, n, is not large enough, or if $Y_1,\dots Y_n$ are not necessarily normal deviates but have a distribution hard to evaluate and if $\hat{C}$ is an estimator for the mean C suspected to be inefficient, then a '*robust*' likelihood can be computed with the help of bootstrap technique. There are two principal steps in the development of this idea. First, generate B random samples from the empirical distribution of Y's and for each sample estimate C. Get the B simulated estimates $\left(\hat{C}^{*}_{1} \quad \hat{C}^{*}_{2} \quad \dots \quad \hat{C}^{*}_{B}\right)$.

Next, compute: $\hat{L} = \frac{1}{\sqrt{2\pi}} B^{-5/6} \sum_{r=1}^{B} e^{-0.5\left[\frac{C-(2\hat{C}-\hat{C}^{*}_{r})}{B^{-5/6}}\right]^2}$. Then combine prior densities on coefficients expressing additional information and get the correspondent posterior densities. In the case of the RSG algorithm: C is the '*true'* value of a coefficient, $\hat{C}$ stands for the RSG estimate of the C coefficient and $\left(C^{*}_{r}\right)_{r=1\dots B}$ will be the bootstrapped RSG estimates of the C coefficient.

6.3 Monte Carlo experiment: design and numerical results

Consider the linear regression model:

$Y = X\,C + \varepsilon$ where X = *rndu (n, p)* is fixed in repetitive samples

C is a *(p,1)* vector of random variables

ε = *rndn (n, 1)* is the vector of random perturbations to be added

Assumptions:

- for $i = 1,\dots, p$ consider $C_i \in N(1/p, sigma)$
- for $i, j = 1.., p$ C_i and C_j are independent random variables

Principal steps in performing this experiment follow.

Step1 set

- the number of extractions (*no_ext*) from a- priori distribution function of the coefficients' vector
- the number of bootstrap replications B required for the likelihood function computation
- the number of Monte Carlo replications (MCR)

generate X = *rndu (n, p)* once and keep it fixed in repetitive samples

For each Monte Carlo replication, indexed by i (i=1,..,MCR) generate the vector of random perturbations $\varepsilon^{(i)}$ and execute *Step 2* :

Step 2 set j=1;

do until j > no_ext;

- consider $C^{(j)} = [C_1^{(j)}, ..., C_p^{(j)}]$ a realization of the random vector of coefficients C
- compute $Y^{(j)} = X C^{(j)} + \varepsilon^{(i)}$
- using the previous $Y^{(j)}$ and X compute the vector of RSG-estimates for the coefficients: $C^{RSG,(j)}$
- by sampling with replacement B times from the residual vector $\varepsilon^{(i)}$ compute correspondent B-bootstrapped RSG vectors of coefficients $C^{*,RSG}$
- compute for each coefficient the bootstrapped likelihood function $\hat{L}$
- multiply $\hat{L}$ with the correspondent a-priori distribution- denote with g the result

j=j+1;
end do;

Results

The following correspondence may be considered:

$$\begin{pmatrix} C_1 ..... & C_p \\ g(C_1)..... & g(C_p) \end{pmatrix}$$

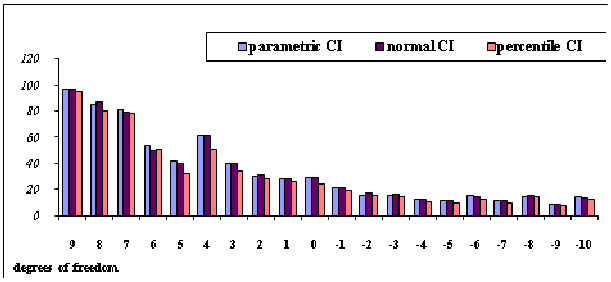
Next:

- adjust *g* to be a proper distribution (to integrate to one)
- compute (bootstrap) confidence intervals around the mean of the g distribution
- count how many times RSG estimates for the coefficients fall in these intervals

Parameters for the presented model are as follows:

- *number of observations* n=10
- *number of coefficients* is varied from 1 to 20 (d.f. varies from 9 to –10)
- *number of extractions* from the a-priori distribution no_ext =50
- *number of bootstrap replications* (for computing the likelihood function) B=50
- *number of bootstrap replications for computing confidence intervals* around the mean of a-posteriori distribution :50
- *number of Monte Carlo replications* :10

Figure 5. A posteriori confidence intervals



The results depicted in Figure 5 lead to the conclusion, that a posteriori confidence intervals can be computed in the case of degrees of freedom higher or equal to *8*. Further Monte Carlo experiments have to be performed. The advantage of the method used is that it incorporates prior information, which performs well without direct knowledge of the RSG estimator distribution since the bootstrap estimate of the sampling density considered is a robust estimator. On the other hand, this sort of bayesian approach is feasible only when the number of observations is at least ten since otherwise repetitive selections with replacement from the empirical errors' distribution may be inappropriate. From this point of view, for example, it should be done a distinction between two degrees of freedom coming from ten observations and eight parameters to be estimated or four observations and two parameters to be estimated.

7. ONE LOOK BACK, ONE GLANCE AHEAD

A complete comparison among SA, RSG and RSGBOOT has to be done preliminary on every model to be estimated, linear or not, since these stochastic optimization algorithms are very sensitive to model specification.

RSGBOOT can be considered the RSG's robust version, when high degrees of uncertainty are governing the initial conditions.

Bayesian framework seems to be suitable for high enough degrees of freedom. When the number of available observations is higher than ten, no matter how many coefficients are to be estimated, the adequacy of a posteriori confidence intervals has to be studied in connection with the robust properties of the bootstrapped likelihood considered (e.g., considering various a priori variances attached, both time independent and dependent). Also, one has to distinct between, e.g., two degrees of freedom coming from four observations and two coefficients, and respectively coming from ten observations and eight coefficients.

The problem of determining confidence intervals can be tested to both RSGBOOT and SA's point estimators in a similar manner. Eventual success could

have direct application, for instance in testing stationarity for very short time series - and many other similar problems.

Acknowledgements
A special debt is owed to Wojciech Charemza for providing a consistent stream of encouragement and suggestions.

REFERENCES

[1] **Agapie, A. (2009)**, ***Convergence of the Guesstimation Algorithm,*** *Communications in Statistics-Theory and Methods*, **38**, 711-718;
[2] **Agapie, A. (1999)**, ***Stochastic Optimisation in Econometric Models - A Comparison of GA, SA and RSG***, MEET IV, *Research Memorandum ACE Project*, 99/1;
[3] **Bădin, L. (2007)**, ***A Note on Some Exact Jackknife and Bootstrap Estimators,*** *Economic Computations and Economic Cybernetics Studies and Research*, 41, 1-2, 179-188;
[4] **Boss, D., Monahan, J. (1986)**, ***Bootstrap Methods Using Prior Information***, *Biometrika*, vol.73, 1, 77-83;
[5] **Charemza, W. (2002)**, ***Guesstimation***, *Journal of Forecasting 21,* 417-433;
[6] **Goffe, W., Ferrier, G., Rogers, J. (1994)**, ***Global Optimization of Statistical Functions with Simulated Annealing,*** *Journal of Econometrics*, 60, 65-99;
[7] **Hall, P. (1987)**, ***On the Bootstrap and Likelihood-based Confidence Regions***, *Biometrika,* vol.74, 3, 481-93;
[8] **Kiefer, J. (1953)**, ***Sequential Minmax Search for a Maximum****, Proceedings of the American Mathematical Society*, 4, 502-506;
[9] **Marcet, A. (1991)**, ***Solving Non-Linear Models by Parameterizing Expectations,*** *Mimeo,* Northwestern University;
[10] **Marcet, A., Sargent, T. (1989)**, ***Convergence of Least Squares Learning with Hidden State Variables and Private information***, *Journal of Political Economy*, vol. 97, 6;
[12] **Wolpert, D., Macready, W. (1997)**, ***No Free Lunch Theorems for Optimization***, IEEE Trans. on Evolutionary Computation, vol. 1, 1, 67-82.