Standard Error Computations for Uncertainty Quantification in Inverse Problems

: Focused on Bootstrap Method

Sangman Jung
Department of Applied Mathematics, Kyung Hee University

Abstract

1. We computationally investigate two approaches for uncertainty quantification in inverse problems for nonlinear parameter dependent dynamical systems.

2. We compare and contrast parameter estimates, standard errors, confidence intervals, and computational times for Bootstrapping method.

Introduction

One of the more ubiquitous computational problems in all of science and engineering is the inverse problem for estimation of parameters from longitudinal observations of system responses. This is usually formulated in terms of a parameter dependent dynamical mathematical model (ordinary, partial, delay differential or integral equation) for which observations of solutions (or certain components of the solutions) are to be used to estimate some unknown parameters.

Dynamical model we interested :

$$\frac{dx(t)}{dt} = rx(t)\left(1-\frac{x(t)}{K}\right),\ x(0)=x_0$$

General solution and parameter :

$$x(t) = f(t,\theta) = \frac{K}{1+\left(\frac{K}{x_0}-1\right)e^{-rt}},\ \theta=(K,r,x_0)$$

We set the true value :

$$\theta = (17.5, 0.7, 0.1) \text{ and } 0 \le t \le 25$$

We want to find the estimate $\hat{\theta}$, Standard error, and Confidence intervals in that solution using bootstrapping method and asymptotic theory.
In order to estimating θ, we use the methods : OLS/GLS.
This methods are used for regression problems in general.

OLS / GLS ?

Since we want to minimize ϵ_j, we can apply OLS method for ϵ_j, and hence we get the following equation for y_j.

1. OLS (Ordinary Least Square) :
Assume that $y_j = f(t_j, \theta_0) + \epsilon_j$, then OLS estimate is

$$\hat{\theta}_{OLS} = \hat{\theta}^n_{OLS} = \arg\min_{\theta\in\Theta_{ad}} \sum_{j=1}^{n} \left[y_j - f(t_j,\theta)\right]^2$$

2. GLS (General Least Square) :
Assume that $y_j = f(t_j, \theta_0)(1+\epsilon_j)$, then GLS estimate is

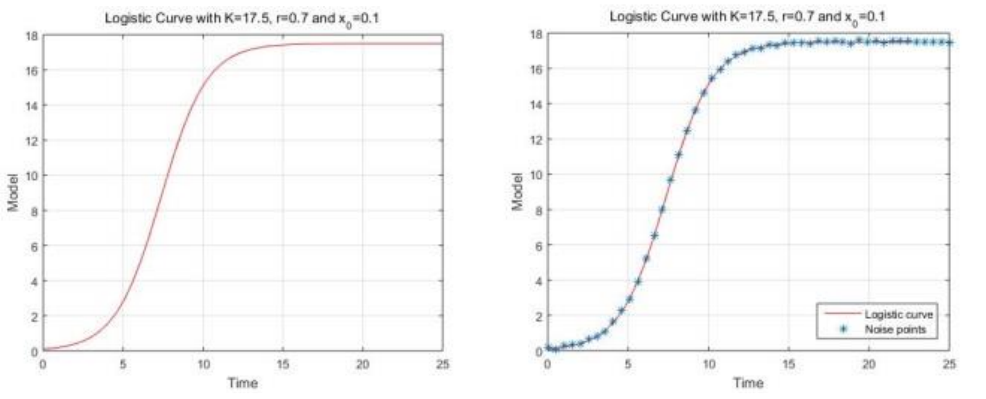
$$\hat{\theta}_{GLS} = \arg\min_{\theta\in\Theta_{ad}} \sum_{j=1}^{n} \widehat{\omega_j}\left[y_j - f(t_j,\theta)\right]^2$$

where weights $\widehat{\omega_j} = f^{-2}(t_j, \hat{\theta}_{GLS})$, initial GLS estimates calculated by $\hat{\theta}_{OLS}$.

Data sets and Algorithm

1. Simulated noisy data sets

For the underlying model with the parameter values $\theta = (17.5, 0.7, 0.1)$, we assume that $y_j = f(t_j, \theta_0) + \epsilon_j$, ϵ_j : noise, y_j : observations.



On the left side of the figures is Logistic curve of the true value $\theta = (17.5, 0.7, 0.1)$, and On the right side of the figures is noisy data sets with noise level : 5% by Logistic curve of the true value of parameter.

2. Bootstrapping algorithm for CV data using OLS

(1) First estimate $\widehat{\theta^0} = (\widehat{K^0}, \widehat{r^0}, \widehat{x_0^0})$ from the entire sample $\{y_j\}_{j=1}^{n}$ using OLS.

(2) Using this estimate define the standardized residuals

$$\overline{r_j} = \sqrt{\frac{n}{n-p}}\left(y_j - f(t_j, \widehat{\theta^0})\right) \text{ for } j = 1, \cdots, n.$$

(3) Create a bootstrap sample of size n using random sampling with replacement from the data $\{\overline{r_1}, \cdots, \overline{r_n}\}$ to form a bootstrap sample $\{r_1^m, \cdots, r_n^m\}$.

(4) Create bootstrap sample points

$$y_j^m = f(t_j, \hat{\theta}^0) + r_j^m \text{ where } j = 1, \cdots, n.$$

(5) Obtain a new estimate $\hat{\theta}^{m+1} = (\widehat{K}^{m+1}, \hat{r}^{m+1}, \widehat{x_0}^{m+1})$ from the bootstrap sample $\{y_j^m\}$ using OLS. Add $\hat{\theta}^{m+1}$ into the vector Θ, where Θ is a vector of length M which stores the bootstrap estimates.

(6) Set $m = m+1$ and repeat steps 3-5.

(7) Carry out the above iterative process M times where M is large (e.g., $M = 1000$), resulting in a vector Θ of length M.

(8) We then calculate the mean, standard error, and confidence intervals from the vector Θ using the formulae

$$\hat{\theta}_{boot} = \frac{1}{M}\sum_{m=1}^{M} \hat{\theta}^m,$$

$$Cov(\hat{\theta}_{boot}) = \frac{1}{M-1}\sum_{m=1}^{M} (\hat{\theta}^m - \hat{\theta}_{boot})^T(\hat{\theta}^m - \hat{\theta}_{boot}),$$

$$SE_k(\hat{\theta}_{boot}) = \sqrt{Cov(\hat{\theta}_{boot})_{kk}}$$

3. Bootstrapping algorithm for NCV data using GLS

It is very similar to the algorithm that CV data using OLS, but it is a little different to calculate a weights and using GLS estimates.

Results of numerical simulations

Table 1
Bootstrap OLS estimates for CV data, noise level = 0.01

θ	$\hat{\theta}$	SE($\hat{\theta}$)	95% CI
$\widehat{K}_{boot}$	17.516934	0.018183	(17.481295, 17.552573)
$\hat{r}_{boot}$	0.700754	0.003763	(0.693378, 0.708130)
$\widehat{x_0}_{boot}$	0.100052	0.002568	(0.095018, 0.105086)

Table 2
Bootstrap OLS estimates for CV data, noise level = 0.05

θ	$\hat{\theta}$	SE($\hat{\theta}$)	95% CI
$\widehat{K}_{boot}$	17.519803	0.021869	(17.476938, 17.562668)
$\hat{r}_{boot}$	0.700469	0.007163	(0.686428, 0.714510)
$\widehat{x_0}_{boot}$	0.100300	0.006172	(0.088203, 0.112398)

Table 3
Bootstrap OLS estimates for CV data, noise level = 0.1

θ	$\hat{\theta}$	SE($\hat{\theta}$)	95% CI
$\widehat{K}_{boot}$	17.504109	0.030023	(17.445263, 17.562954)
$\hat{r}_{boot}$	0.701875	0.012291	(0.677842, 0.725966)
$\widehat{x_0}_{boot}$	0.099066	0.010489	(0.078506, 0.119626)

Table 4
Computation times (s) for bootstrapping in OLS

Noise level (%)	Bootstrapping
1	7.746404
5	7.785673
10	9.815415

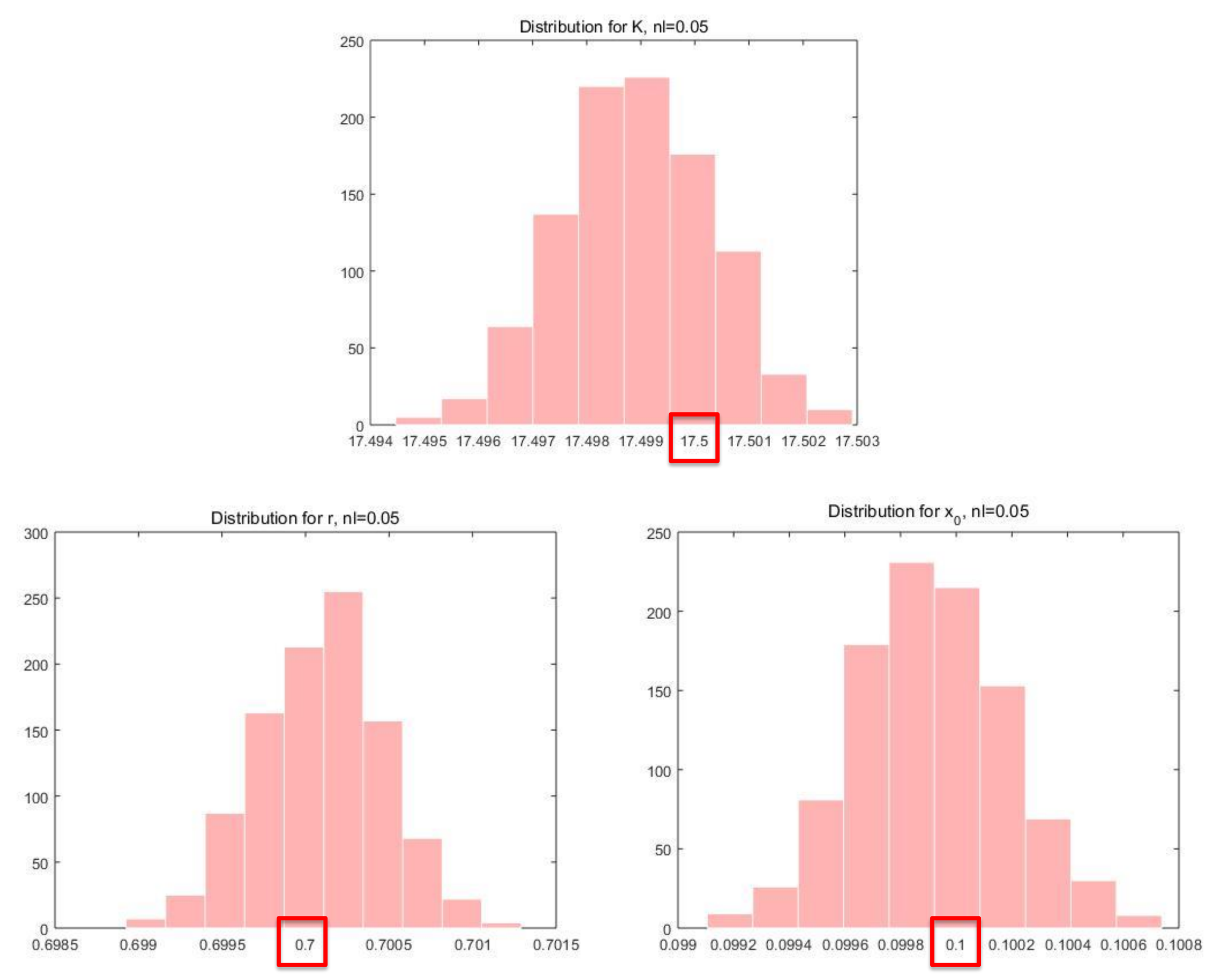


Table 1
Bootstrap GLS estimates for NCV data, noise level = 0.01

θ	$\hat{\theta}$	SE($\hat{\theta}$)	95% CI
$\widehat{K}_{boot}$	17.527813	0.233004	(17.071125, 17.984501)
$\hat{r}_{boot}$	0.706952	0.115032	(0.481488, 0.932415)
$\widehat{x_0}_{boot}$	0.103641	0.098708	(−0.089826, 0.297109)

Table 2
Bootstrap GLS estimates for NCV data, noise level = 0.05

θ	$\hat{\theta}$	SE($\hat{\theta}$)	95% CI
$\widehat{K}_{boot}$	17.527055	0.241803	(17.053121, 18.000990)
$\hat{r}_{boot}$	0.705166	0.118306	(0.473286, 0.937047)
$\widehat{x_0}_{boot}$	0.105288	0.102703	(−0.096009, 0.306586)

Table 3
Bootstrap GLS estimates for NCV data, noise level = 0.1

θ	$\hat{\theta}$	SE($\hat{\theta}$)	95% CI
$\widehat{K}_{boot}$	17.531845	0.265714	(17.011045, 18.052646)
$\hat{r}_{boot}$	0.700993	0.130958	(0.444315, 0.957671)
$\widehat{x_0}_{boot}$	0.110671	0.114475	(−0.113701, 0.335044)

Table 4
Computation times (s) for bootstrapping in GLS

Noise level (%)	Bootstrapping
1	11.910531
5	12.107679
10	11.939790

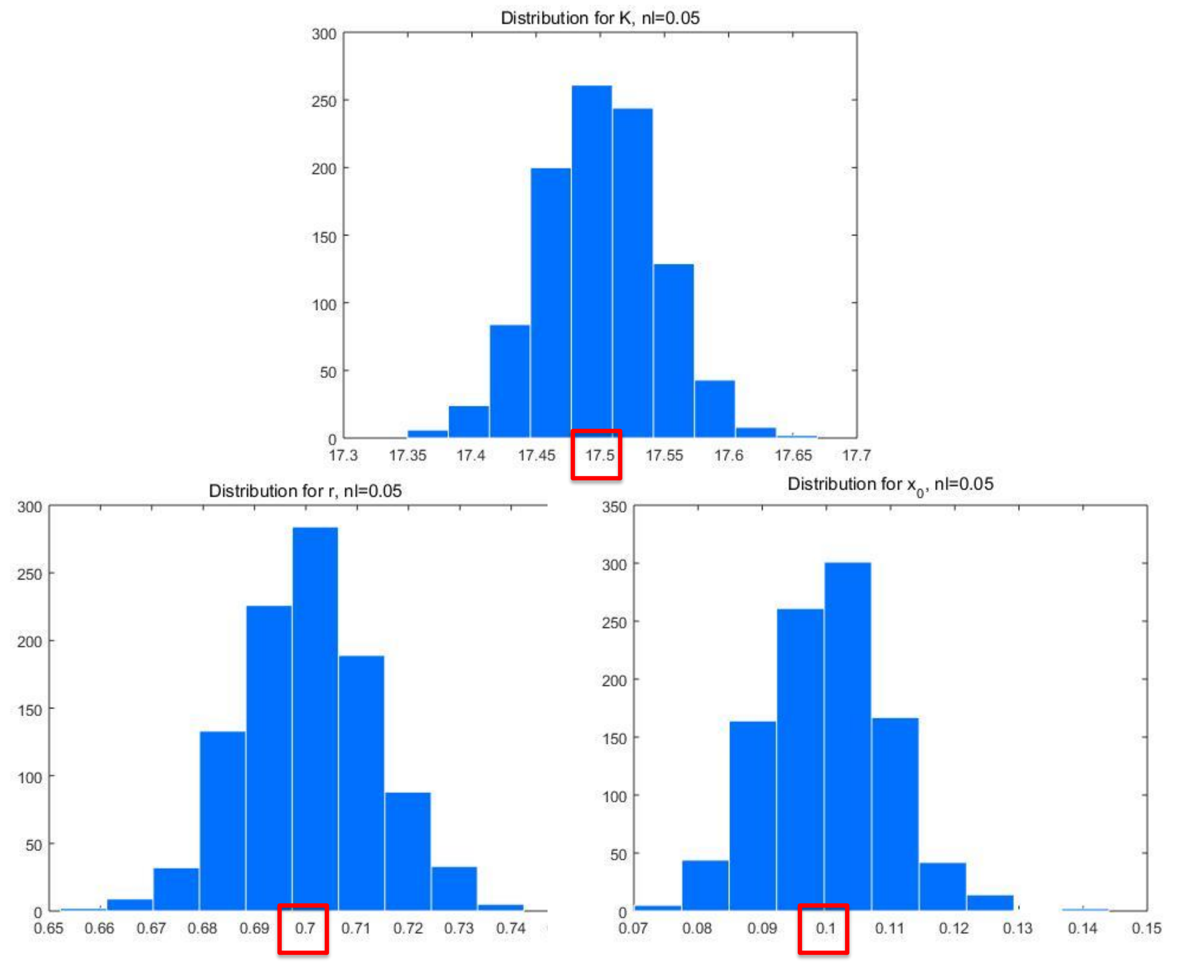
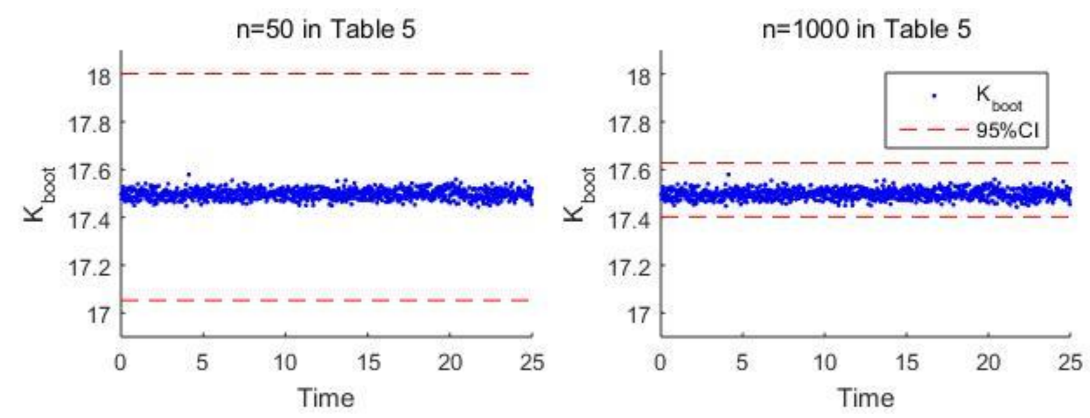


Table 5
Bootstrap GLS estimates for NCV, ①$n = 50$ vs ②$n = 1000$

θ	$\hat{\theta}$	SE($\hat{\theta}$)	95% CI
① $\widehat{K}_{boot}$	17.527055	0.241803	(17.053121, 18.000990)
① $\hat{r}_{boot}$	0.705166	0.118306	(0.473286, 0.937047)
① $\widehat{x_0}_{boot}$	0.105288	0.102703	(−0.096009, 0.306586)
② $\widehat{K}_{boot}$	17.514899	0.057610	(17.401983, 17.627816)
② $\hat{r}_{boot}$	0.701388	0.027306	(0.647868, 0.754909)
② $\widehat{x_0}_{boot}$	0.100072	0.023268	(0.054465, 0.145679)

Figure
Confidence Intervals, ①$n = 50$ vs ②$n = 1000$



Discussion

In the case of using Bootstrapping for CV and NCV, they were all well-estimated and especially the GLS method was shown more suitable than using the OLS method because the GLS method gives the estimates which are NCV.
Table 5 illustrates that the larger the size of n becomes, the smaller the confidence interval is and because of this reason, we assured that we have obtained more precise estimates.
This eventually means that finding more approximate estimates depend on the size of n. Additionally, we were able to learn that the size of the sample, the efficiency of the code and functions that are stored in the calculating software are several ways of reducing the speed of calculating Bootstrapping.

References

Original Paper :
Standard error computations for uncertainty quantification in inverse problems: Asymptotic theory vs. bootstrapping

H.T. Banks ∗ , Kathleen Holm, Danielle Robbins
Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC 27695-8212, United States
Center for Quantitative Sciences in Biomedicine, North Carolina State University, Raleigh, NC 27695-8212, United States

Independent Learning & Research total process :
https://github.com/92normal/Independent_Learning_and_Research_1
If you have more question, please contact jsm2014@khu.ac.kr.