# Statistics 2

## - Final Project -

Standard Error Computations for Uncertainty Quantification in Inverse Problems:
Asymptotic Theory vs. Bootstrapping

응용수학과

2014110375

최우빈

< Contents >



---------------------------------------------------------------------------------------------------------------

(1) Introduction

파라미터에 의존적인 dynamical mathematical model (ODE or integral equation) 중 하나로, inverse problem 방식을 이용하여 인구 수 모델을 bootstrapping 과 asymptotic theory 2가지 방법을 사용하였다. Inverse problem 이란, 데이터로부터 모델의 파라미터를 추정하는 것을 말한다,

인구 모델은 Verhulst-Pearl growth model 을 사용하였다.

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

위 식을 변수 분리형 미분방정식을 사용하여, $x(t)=f(t,\theta)=\dfrac{K}{1+\left(\frac{K}{x_0}-1\right)e^{-rt}},$

변형하였다. 사용 되는 파라미터는 $\theta=(K,r,x_0)$ 이다.

예측하고자 하는 최종 파라미터는 K = 17.5, r = 0.7, x0 = 0.1 이다.

이제 Contents 에서 소개한 (2) , (3) , (4) 방법을 차례로 소개하여 인구 수를 예측해 볼 것이다.

(2) Bootstrapping Algorithm for Constant Variance Data

Bootstrap 이란, 일반적으로 한번시작 되면 알아서 진행되는 일련의 과정들을 말한다.

통계학에선 모집단의 성질에 대해 표본을 통해 추정할 수 있는 것처럼, 표본의 성질에 대해서도 재표본을 통해 추정할 수 있다는 것이다. 즉 주어진 표본(샘플)에 대해서, 그 샘플에서 또 다시 샘플(재표본)을 여러 번(1,000~10,000번, 혹은 그 이상)추출하여 표본의 평균이나 분산 등이 어떤 분포를 가지는가를 알아낼 수 있는 것 이다.

OLS(Ordinary Least Square) 이란, 선형회귀분석에서 모르는 변수를 추정하는 방법이다. 이 방법은 단어의 뜻에서 알 수 있듯이 실제 관찰된 값과 추정에 의해 예상된 값의 차이의 제곱을 최소화시키는 방법이다. 이 방법은 결과값(종속변수)에서 나타나는 에러들을 최소화 시키는 방법이기 때문에 독립변수에는 에러가 발생하지 않거나 무시할 만한 수준의 에러가 존재한다고 가정한다. OLS는 자료들 간의 분산이 동일한 경우에 적용할 수 있는 방법이다.

$Y_j = f\left(t_j, \theta_0\right) + \mathcal{E}_j, \quad j = 1, \ldots, n,$ 의 관찰 과정에서, 주어진 실험 데이터, $(y_1, t_1), \ldots, (y_n, t_n)$ 를 가정하자.

여기서 $\mathcal{E}_j$ 은 F – 분포로부터, 평균은 0 $(E\left(\mathcal{E}_j\right) = 0)$ 과 constant variance $\sigma_0^2$, and $\theta_0$ 은 true value parameter 이다.

그 후 관찰된 데이터 들의 error를 측정할 것인데 OLS(ordinary Least Square) 방법을 사용하였다.

$$\theta_{\text{OLS}}(Y) = \theta_{\text{OLS}}^n(Y) = \arg\min_{\theta \in \Theta_{ad}} \sum_{j=1}^{n} \left[ Y_j - f\left(t_j, \theta\right) \right]^2,$$

$\theta_{\text{OLS}}$ 를 관찰된 데이터와 실제 모델 값들 차이를 minimize 하기 위해, $\sum_{j=1}^{n} \left[ Y_j - f\left(t_j, \theta\right) \right] \nabla f\left(t_j, \theta\right) = 0.$ 식을 사용하였다. 여기서, $\mathcal{E}_j = Y_j - f\left(t_j, \theta\right)$ 이것은 일종의 noise 를 말한다. $\widehat{\theta}_{\text{OLS}} = \widehat{\theta}_{\text{OLS}}^n = \arg\min_{\theta \in \Theta_{ad}} \sum_{j=1}^{n} \left[ Y_j - f\left(t_j, \theta\right) \right]^2$ 이 식을 통해 앞에서 언급한 theta(OLS)를 예측하고자 한다.

그 후, $\sigma_0^2 \approx \widehat{\sigma}_{OLS}^2 = \frac{1}{n-p} \sum_{j=1}^{n} \left(y_j - f\left(t_j, \widehat{\theta}\right)\right)^2.$ unbiased estimator 인 $\sigma_0^2$ 을 예측하기 위해, $\widehat{\sigma}_{\text{OLS}}^2$ 를 통해 estimate 하였다.

이제 MATLAB 으로 bootstrapping 을 constant variance data에 적용하기 위한 8 단계를 소개 하겠다.

1. OLS를 이용한 전체 실험 데이터 로부터, $\widehat{\theta}^0=\left(\widehat{K}^0,\widehat{r}^0,\widehat{x}_0^0\right)$ 를 추정한다.

2. standardized residuals를 $\bar{r}_j=\sqrt{\frac{n}{n-p}}\left(y_j-f\left(t_j,\widehat{\theta}^0\right)\right)$ 로 세운 후에, 파라미터 수 p = 3으로 놓고 Random variable이 F – 분포를 따른다.

3. size n의 bootstrap sample 을 만들기 위하여,

을 bootstrap sample $\{r_1^m,\ldots,r_n^m\}$. 의 형태로 대체한다the data (realizations) $\{\bar{r}_1,\ldots,\bar{r}_n\}$

4. bootstrap sample의 위치를 $y_j^m=f\left(t_j,\widehat{\theta}^0\right)+r_j^m$, 통해 구한다.

5. bootstrap으로 측정해서 저장한 length M 만큼의 vector theta를 OLS 방법을 통해서

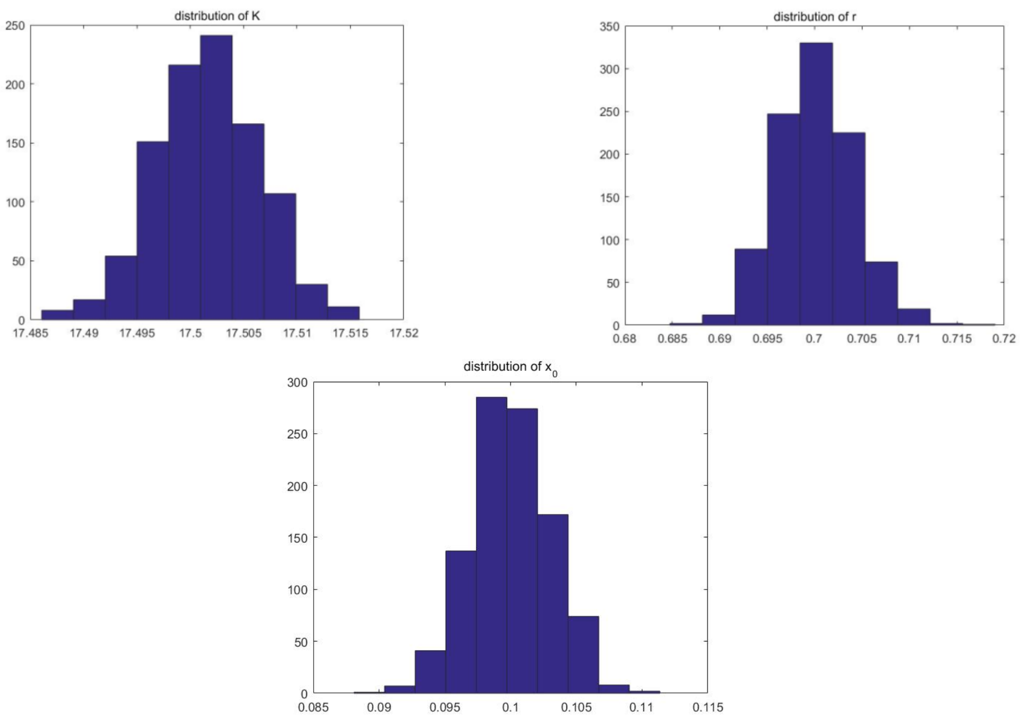
$\widehat{\theta}^{m+1}=\left(\widehat{K}^{m+1},\widehat{r}^{m+1},\widehat{x}_0^{m+1}\right)$ 를 새로 계속 측정하게 된다.

6. m = m + 1을 이용하여 for문을 만든 후 3.4.5의 단계들을 반복적으로 실행한다,

7. M 번의 연산과정을 넘어가게 되면(보통 1000번), length M 만큼의 vector theta를 print하면 된다.

8. 최종적으로, bootstrap sampling을 통해 mean, standard error, confidence interval 을 vector theta 를 통해서 구할 수 있다.

< Bootstrap Parameter Distributions Corresponding to 5% Noise with Constant Variance data >



위 3가지 표를 통해 총 1000번의 bootstrapping 을 하 였을 때, noise = 0.05로 정하고 true value parameter 인 K = 17.5, r = 0.7, x0 = 0.1 근방에 sample 들이 가장 많이 분포함을 알 수 있다.

(3) Asymptotic Theory for Constant Variance Data

Asymptotic theory 이란, large sample theory 라고 불리기도 하며, estimator와 통계적 검정을 위한, property를 측정한 genetic framework 이다. 이 framework는 일반적으로 n 이 무한대로 수렴함으로써 통계적 절차와 sample size n을 가정한다.

이제 주어진 statistical model을 통하여, asymptotic theory를 사용해서, standard error 와 estimates를 계산할 수 있다.

다음 4가지 단계를 MATLAB을 통해 하는 방법을 설명하고자 한다.

1. OLS를 사용하여, $\widehat{\theta}=\left(\widehat{K},\widehat{r},\widehat{x_0}\right)$ 를 추정한다.

2. sensitivity equation 인, $\frac{d}{dt}\frac{\partial x}{\partial \theta}=\frac{\partial \mathcal{F}}{\partial x}\frac{\partial x}{\partial \theta}+\frac{\partial \mathcal{F}}{\partial \theta}.$ 식을 활용하여, 미분방정식의 하나인,

$\frac{dx(t)}{dt}=\mathcal{F}\left(x(t,\theta),\theta\right)=rx(t)\left(1-\frac{x(t)}{K}\right),$ 식을 사용하여, logistic model을 구할 수 있다.

그리고 sensitivity matrix $\chi=\frac{\partial f}{\partial \theta},$ 와 variance 를 연산 할 수 있다.

따라서 ODE가 제공하는 solution은 $\chi_{j,k}=\frac{\partial x\left(t_j\right)}{\partial \theta_k}=\frac{\partial f\left(t_j,\theta\right)}{\partial \theta_k},\quad \text{for } j=1,\ldots,n,\ k=1,\ldots,p.$ 이다.

그리고, $\chi=\chi^n$ 이고, n X p matrix 이다.

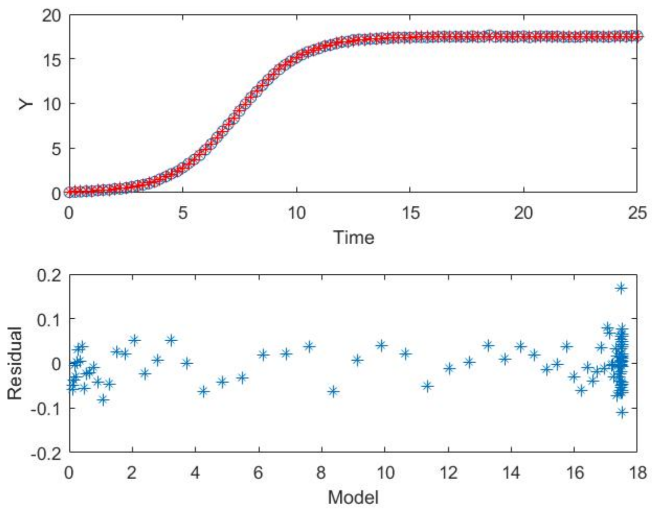
constant variance 의 unbiased estimate 는 $\sigma_0^2 \approx \widehat{\sigma}_{OLS}^2=\frac{1}{n-p}\sum_{j=1}^{n}\left(y_j-f\left(t_j,\widehat{\theta}\right)\right)^2$ 이다.

3. Estimate the covariance matrix.

대략적인 true covariance matrix 는 $\Sigma_0^n=\sigma_0^2\left[\chi^T(\theta_0)\chi(\theta_0)\right]^{-1}$, 형태 인데, true parameter theta0 와 covariance는 알려지지 않았다. 따라서 우리는 covariance matrix를 $\Sigma_0^n \approx \widehat{\Sigma}^n\left(\widehat{\theta}\right)=\widehat{\sigma}_{OLS}^2\left[\chi^T\left(\widehat{\theta}\right)\chi\left(\widehat{\theta}\right)\right]^{-1}$ 에 의해, $\widehat{\theta}$ 와 $\widehat{\sigma}_{OLS}^2$ 사용하여 구할 것 이다.

4. $\widehat{\Sigma}^n\left(\widehat{\theta}\right)$ 사용하여 standard error을 연산할 것이다. $SE_k\left(\widehat{\theta}\right)=\sqrt{\widehat{\Sigma}_{kk}^n\left(\widehat{\theta}\right)}.$

< Logistic curve with K = 17.5, r =0.7 and x_0 = 0.1 >




위 그래프는 시간(t) 에 따른 K값의 수렴 과정이고,

아래 그래프는 K값을 가지는 model 값들의 residual

을 나타낸다.

noise = 0.05로 정하였다.

(4) Bootstrapping Algorithm for Non - constant Variance Data

(2), (3)은 constant variance data를 사용하여 bootstrapping 와 asymptotic theory 이었지만,

이번에 GLS(General Least Square) 기반의 bootstrapping을 구현해 볼 것 이다.

먼저 GLS 란, 통계학에서 OLS와 마찬가지로 선형회귀분석에서 모르는 변수를 추정하는 분석기법이다. GLS는 자료들의 분산이 동일하지 않을 때(이분산성이 존재하는 경우) 또는 자료들 간에 상관관계가 존재할 때 GLS를 적용한다. 이러한 상황에서는 OLS 보다 GLS가 더 효율적이고 정확한 추정을 제공한다.

**처음에 실험 데이터가 주어졌을 때, 관찰 과정을,** $Y_j = f\left(t_j, \theta_0\right)\left(1+\mathcal{E}_j\right)$

**이 식으로 정의한다.**

그 후 GLS는 OLS와 다르게 연산 할 때, 가중치 W를 곱한다.

$\sum_{j=1}^{n} w_j \left[ Y_j - f\left(t_j, \theta_{\mathrm{GLS}}\right)\right] \nabla f\left(t_j, \theta_{\mathrm{GLS}}\right) = 0,$ 여기서, $w_j = f^{-2}(\mathrm{tj}\ , \theta\mathrm{GLS}).$ 이다.

이제 MATLAB 연산을 위한, GLS 기반 bootstrapping 방법의 8 단계를 소개 하겠다.

1. GLS를 사용하여, $\widehat{\theta}^0 = \left(\widehat{K}^0, \widehat{r}^0, \widehat{x}_0^0\right)$를 추정한다.

2. the non-constant variance standardized residuals 를 정의한다. $\bar{s}_j = \dfrac{y_j - f\left(t_j, \widehat{\theta}^0\right)}{f\left(t_j, \widehat{\theta}^0\right)}.$

3. $\left\{\bar{s}_1, \ldots, \bar{s}_n\right\}$ 의 데이터를 bootstrap sample의 형태인 $\left\{\bar{s}_1^m, \ldots, \bar{s}_n^m\right\}.$ 을 대체하고, size n 만큼의

Random sampling bootstrap sample을 만들어낸다.

4. bootstrap sample 들의 위치는 $y_j^m = f\left(t_j, \widehat{\theta}^0\right) + f\left(t_j, \widehat{\theta}^0\right) s_j^m,$ 이 식을 사용하여, 만들어낸다.

5. GLS를 사용하여 bootstrap sample 들로부터, 새로운 estimate $\widehat{\theta}^{m+1} = \left(\widehat{K}^{m+1}, \widehat{r}^{m+1}, \widehat{x}_0^{m+1}\right)$

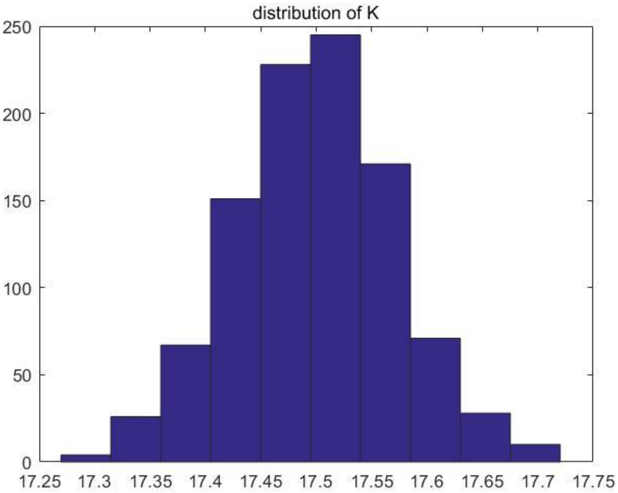
을 얻을 수 있다. 이 때, bootstrap estimate를 저장한 length M 만큼의 vector theta를 알 수 있다.

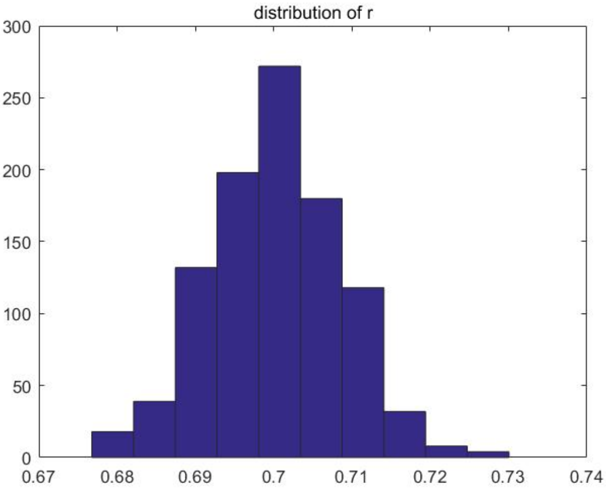
6. m = m+1로 for문을 이용하여 3,4,5번 단계를 반복 실행한다.

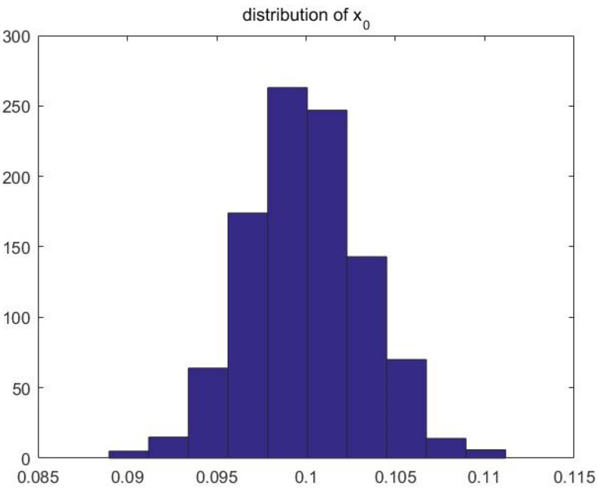
7. M번의 연산과정(보통 1000번)이 끝나고 난 후, length M의 vector theta 의 결과값을 알 수 있다.

8. OLS bootstrap과 마찬가지로 vector theta를 통해 mean , standard error, confidence interval을 구할 수 있다.

< Bootstrap Parameter Distributions for 5% Noise with Non - Constant Variance >




- 위 그래프를 통해 OLS 와 비교 했을 때, GLS 의 장점은 에러 추정치의 가중치를 계속 가중 적용시켜서 더 잘 맞는 모델을 만들 수 있다는 것이다.

(5) Conclusion

Asymptotic theory과 Bootstrapping은 매개 변수 추정의 불확실성을 정량화 한다. Asymptotic theory는 Bootstrapping보다 계산적으로 항상 빠르다. OLS를 사용하는 일정한 분산 데이터의 경우 Bootstrapping을 사용할 때 명확한 이점이 없다. 그러나 Asymptotic theory는 복잡한 시스템의 경우 민감도를 계산하기에 너무 복잡 할 수 있다. 그래서 표준편차를 측정하기엔 Bootstrapping가 더 적절하다.

계산시간이 적절히 고려된다면, Asymptotic theory가 더 유리하다.

**일반적으로 자료들 간의 분산이 동일하지 않은 상태가 보다 일반적인 상황이다. 그러므로 GLS 가 좀더 실생활 관련 문제들을 해결하는 데, 연관성이 있다.** 따라서 동분산성이 존재하는 특수한 상황에서는 OLS 를 적용하므로 OLS 가 GLS 의 특수한 형태로 볼 수 있다. 따라서 OLS 를 적용하는 경우 해당 자료에는 동분산성이 존재한다는 가정이 내재되어 있다. OLS 는 자료들 간에 상관관계가 존재하지 않는 경우에, GLS 는 상관관계가 존재하는 경우에 적용한다. OLS 에서는 독립변수의 계수를 weights 없이 Least Square methods 으로 분석하고, GLS 에서는 correlated 된 자료들 간의 상관계수에 따라 weights 를 적용하여 분석한다.

<Reference>

- Standard Error Computations for Uncertainty Quantification in Inverse Problems: Asymptotic Theory vs. Bootstrapping

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

Published in final edited form as: Math Comput Model. 2010 November 1; 52(9-10): 1610–1625. doi:10.1016/j.mcm.2010.06.026.

Bootstrap_GLS.m

```
% GLS for non - constant
% Bootstrapping    model output, data
% 이방법을 사용해서 CI 를 구할수 있다. distribution을 알아냈으니 mean 과 standard deviation 을
% 알아냈으니 CI 를 구할 수 있다.

clc
clear all

t=0:1:149;                    % x 축

theta=[17 0.6 0.1];           % theta0=[K r x0] K=17.5; r=0.7; x0=0.1;
trueparametervalue=[17.5 0.7 0.1];

n=length(t);                  % 150개
m=1000;                       % 1000번
options=optimset('TolX',1.0e-4,'MaxFunEvals',10000);   % iteration number

lb= [0 0 0];              % lower bound
ub= [inf inf inf];        % upper bound

% 첫번째 열 setting

y(1,:)=normlogistic(trueparametervalue,t);              % y 는 model
Y(1,:)=y(1,:).*(1+(1/20)*randn(1,n));             % Y 는 생성된 data  논문 뒤에 noise = 0.05 로 주어짐
Weights(1,:) = 1./(y(1,:).^2);    % 초기 Weights는 y(1,:)의 -2승

yerrsum = @(x) sum(Weights(1,:).*(Y(1,:)-normlogistic(x,t)).^2); % 초기 Weights를 곱함
theta1(1,:) = fminsearch(yerrsum,theta);
% plot(t,y(1,:))
% 2~1000 열 setting

for i=1:m
    y(i+1,:)=normlogistic(theta1(i,:),t);      % y 는 theta에 의존
    Y(i+1,:)= y(1,:).*(1+(1/20)*randn(1,n));        % Y= y * (1 + r);   r은 residual
    Weights(i+1,:) = 1./(y(i+1,:).^2);   % Weights 를 1000번 반복문 실행하여 조정.
    yerrsum = @(x) sum(Weights(i+1,:).*(Y(i+1,:)-normlogistic(x,t)).^2); % 반복문 실행할 때 마다 생긴 Weights를 곱함
    theta1(i+1,:) = fminsearch(yerrsum,theta1(i,:));
end

figure(1)    %K
hist(theta1(:,1))
title('distribution of K')
figure(2)    %r
hist(theta1(:,2))
title('distribution of r')
figure(3)    %x0
hist(theta1(:,3))
title('distribution of x_0')
```