

Output Analysis and Run-Length Control

1 Output Analysis

Recall the simulation framework we use when we want to estimate $\theta = E[h(\mathbf{X})]$ where $\mathbf{X} \in \mathbf{R}^n$:

- first simulate $\mathbf{X}_1, \dots, \mathbf{X}_n$ IID and then set

$$\hat{\theta}_n = \frac{h(\mathbf{X}_1) + \dots + h(\mathbf{X}_n)}{n}$$

- strong Law of Large Numbers says

$$\hat{\theta}_n \rightarrow \theta \text{ as } n \rightarrow \infty$$

But at this point we don't know how large n should be so that we can have confidence in $\hat{\theta}_n$ as an estimator of θ . Put another way, for a fixed value of n , what can we say about the quality of $\hat{\theta}_n$? We will now answer this question and to simplify our notation we will take $Y_i := h(\mathbf{X}_i)$.

1.1 Confidence Intervals

One way to answer this question is to use a *confidence interval*. Suppose then we want to estimate θ and we have a random vector $\mathbf{Y} = (Y_1, \dots, Y_n)$ whose distribution depends on θ . Then we seek $L(\mathbf{Y})$ and $U(\mathbf{Y})$ such that

$$\mathbf{P}(L(\mathbf{Y}) \leq \theta \leq U(\mathbf{Y})) = 1 - \alpha$$

where $0 \leq \alpha \leq 1$ is a pre-specified number. We then say that $[L(\mathbf{Y}), U(\mathbf{Y})]$ is a $100(1 - \alpha)\%$ confidence interval for θ .

Note that $[L(\mathbf{Y}), U(\mathbf{Y})]$ is a random interval. However, once we replace \mathbf{Y} with a sample vector, \mathbf{y} , then $[L(\mathbf{y}), U(\mathbf{y})]$ becomes a real interval. We now discuss the Chebyshev Inequality and the Central Limit Theorem, both of which can be used to construct confidence intervals.

1.1.1 The Chebyshev Inequality

Recall the variance of $\hat{\theta}_n$ is given by $\text{Var}(\hat{\theta}_n) = \frac{\sigma^2}{n}$ where $\sigma^2 := \text{Var}(Y)$. We would guess that a small value of $\text{Var}(\hat{\theta}_n)$ implies a more accurate estimate of θ and this is indeed confirmed by *Chebyshev's Inequality* which states that

$$\mathbf{P}\left(|\hat{\theta}_n - \theta| \geq k\right) \leq \frac{\text{Var}(\hat{\theta}_n)}{k^2}. \quad (1)$$

We can see from (1) that a smaller value of $\text{Var}(\hat{\theta}_n)$ therefore improves our confidence in $\hat{\theta}_n$. We could easily use Chebyshev's Inequality to construct (how?) confidence intervals for θ but it is generally very conservative.

Exercise 1 Why does Chebyshev's Inequality generally lead to conservative confidence intervals?

Instead, we will use the Central Limit Theorem to obtain better estimates of $\mathbf{P}\left(|\hat{\theta}_n - \theta| \geq k\right)$ and as a result, narrower confidence intervals for θ .

1.1.2 The Central Limit Theorem

The Central Limit Theorem is among the most important theorems in probability theory and we now state it.

Theorem 1 (Central Limit Theorem)

Suppose Y_1, \dots, Y_n are IID and $E[Y_i^2] < \infty$. Then

$$\frac{\hat{\theta}_n - \theta}{\sigma/\sqrt{n}} \Rightarrow N(0, 1) \text{ as } n \rightarrow \infty$$

where $\hat{\theta}_n = \sum_{i=1}^n Y_i/n$, $\theta := E[Y_i]$ and $\sigma^2 := \text{Var}(Y_i)$.

Note that we assume nothing about the distribution of the Y_i 's other than that $E[Y_i^2] < \infty$. If n is sufficiently large in our simulations, then we can use the CLT to construct confidence intervals¹ for $\theta := E[Y]$. We now describe how to do this.

1.2 A $100(1 - \alpha)\%$ Confidence Interval for θ

Let $z_{1-\alpha/2}$ be the the $(1 - \alpha/2)$ percentile point of the $N(0, 1)$ distribution so that

$$\mathbf{P}(-z_{1-\alpha/2} \leq Z \leq z_{1-\alpha/2}) = 1 - \alpha$$

where $Z \sim N(0, 1)$. Suppose now that we have simulated IID samples, Y_i , for $i = 1, \dots, n$, and that we want to construct a $100(1 - \alpha)\%$ CI for $\theta = E[Y]$. That is, we want $L(\mathbf{Y})$ and $U(\mathbf{Y})$ such that

$$\mathbf{P}(L(\mathbf{Y}) \leq \theta \leq U(\mathbf{Y})) = 1 - \alpha.$$

The CLT implies $\sqrt{n}(\hat{\theta}_n - \theta)/\sigma$ is approximately $N(0, 1)$ for large n so we have

$$\begin{aligned} \mathbf{P}\left(-z_{1-\alpha/2} \leq \frac{\sqrt{n}(\hat{\theta}_n - \theta)}{\sigma} \leq z_{1-\alpha/2}\right) &\approx 1 - \alpha \\ \Rightarrow \mathbf{P}\left(-z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \leq \hat{\theta}_n - \theta \leq z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}\right) &\approx 1 - \alpha \\ \Rightarrow \mathbf{P}\left(\hat{\theta}_n - z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \leq \theta \leq \hat{\theta}_n + z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}\right) &\approx 1 - \alpha. \end{aligned}$$

Our approximate $100(1 - \alpha)\%$ CI for θ is therefore given by

$$[L(\mathbf{Y}), U(\mathbf{Y})] = \left[\hat{\theta}_n - z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}, \hat{\theta}_n + z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \right] \quad (2)$$

Recall that $\hat{\theta}_n = (Y_1 + \dots + Y_n)/n$, so L and U are indeed functions of \mathbf{Y} . There is still a problem, however, as we do not usually know σ^2 . We get around this issue by estimating σ^2 with

$$\hat{\sigma}_n^2 = \frac{\sum_{i=1}^n (Y_i - \hat{\theta}_n)^2}{n - 1}.$$

It is easy to show that $\hat{\sigma}_n^2$ is an unbiased estimator of σ^2 and that $\hat{\sigma}_n^2 \rightarrow \sigma^2$ almost surely as $n \rightarrow \infty$. So now we replace σ with $\hat{\sigma}_n$ in (2) to get

$$[L(\mathbf{Y}), U(\mathbf{Y})] = \left[\hat{\theta}_n - z_{1-\alpha/2} \frac{\hat{\sigma}_n}{\sqrt{n}}, \hat{\theta}_n + z_{1-\alpha/2} \frac{\hat{\sigma}_n}{\sqrt{n}} \right] \quad (3)$$

as our *approximate* $100(1 - \alpha)\%$ CI for θ when n is large.

¹Recall again that $Y_i = h(X_i)$ in our earlier notation.

Remark 1 Note that when we obtain sample values of $\mathbf{y} = (y_1, \dots, y_n)$, then $[L(\mathbf{y}), U(\mathbf{y})]$ becomes a real interval. Then we can no longer say (why not?) that

$$\mathbf{P}(\theta \in [L(\mathbf{y}), U(\mathbf{y})]) = 1 - \alpha.$$

Instead, we say that we are $100(1 - \alpha)\%$ confident that $[L(\mathbf{y}), U(\mathbf{y})]$ contains θ . Furthermore, the smaller the value of $U(\mathbf{y}) - L(\mathbf{y})$, the more confidence we will have in our estimate of θ .

Example 1 (Pricing a European Call Option)

Suppose we want to estimate the price, C , of a call option on a stock whose price process, S_t , is a $GBM(\mu, \sigma)$. The relevant data is $r = .05$, $T = 0.5$ years, $S_0 = \$100$, $\sigma = 0.2$ and strike $K = \$110$. Then we know that

$$C = E^Q[e^{-rT} \max(S_T - K, 0)]$$

where we can assume that $S_t \sim GBM(r, \sigma)$ the risk-neutral probability measure, Q . That is, we assume $S_T = S_0 \exp((r - \sigma^2/2)T + \sigma Z)$ where $Z \sim N(0, T)$.

Though we can of course compute C exactly, we can also estimate C using simulation. Suppose we use $n = 10,000$ samples. Then we can estimate a 95% confidence interval for C as follows.

Matlab Code for Estimating the BS Option Price and Confidence Intervals

```
> S0=100; K=110; T=0.5; r=0.05; sigma=.2;
> n=10000;      % n is the number of samples we use
> Z=sqrt(T)*randn(n,1);
> ST = S0*exp((r-(sigma^2)/2)*T + sigma*Z);
> Payoff = exp(-r*T)*max(0,ST-K);
> % Payoff is the vector of X values in our notation
> C_est = mean(Payoff) % Our estimated price

C_est = 15.2658

>> temp = 1.96*std(Payoff)/sqrt(n);
>> CI_95 = [C_est - temp, C_est + temp]

CI_95 = 15.0129 15.5188
```

1.2.1 Properties of the Confidence Interval

The *width* of the confidence interval is given by

$$U - L = \frac{2\hat{\sigma}_n z_{1-\alpha/2}}{\sqrt{n}}$$

and so the *half-width* then is $(U - L)/2$. The width clearly depends on α , $\hat{\sigma}_n$ and n . However, $\hat{\sigma}_n \rightarrow \sigma$ almost surely as $n \rightarrow \infty$, and σ is a constant. Therefore, for a fixed α , we need to increase n if we are to decrease the width of the confidence interval. Indeed, since $U - L \propto \frac{1}{\sqrt{n}}$, we can see for example that we would need to increase n by a factor of four in order to decrease the width of the confidence interval by only a factor of two.

2 Run-Length Control

Up to this point we have selected n in advance and then computed the approximate CI. The width of the CI is then a measure of the error in our estimate. Now we will do the reverse by first choosing some error criterion that we want our estimate to satisfy, and then choosing n so that this criterion is satisfied.

There are two types of error that we will consider:

1. Absolute error, which is given by $E_a := |\theta_n - \theta|$ and
2. Relative error, which is given by $E_r := \left| \frac{\theta_n - \theta}{\theta} \right|$.

Now we know that $\hat{\theta}_n \rightarrow \theta$ as $n \rightarrow \infty$ so that E_a and E_r both $\rightarrow 0$ as $n \rightarrow \infty$. (If $\theta = 0$ then E_r is not defined.) However, in practice $n \neq \infty$ and so the errors will be non-zero. We specify the following error criterion:

Error Criterion: Given $0 \leq \alpha \leq 1$ and $\epsilon \geq 0$, we want $\mathbf{P}(E \leq \epsilon) = 1 - \alpha$. E is the error type we have specified, i.e., relative or absolute.

The goal then is to choose n so that the error criterion is approximately satisfied and this is easily done.

Suppose, for example, that we want to control absolute error, E_a . Then, as we saw earlier,

$$\mathbf{P} \left(\theta_n - z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \leq \theta \leq \theta_n + z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \right) \approx 1 - \alpha.$$

This then implies $\mathbf{P} \left(|\hat{\theta}_n - \theta| \leq z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \right) \approx 1 - \alpha$, so in terms of E_a we have

$$\mathbf{P} \left(E_a \leq z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \right) \approx 1 - \alpha.$$

If we then want $\mathbf{P}(E_a \leq \epsilon) \approx 1 - \alpha$, it clearly suffices to choose n such that

$$n = \frac{\sigma^2 z_{1-\alpha/2}^2}{\epsilon^2}.$$

If we are working with relative error, then exactly the same type of computation implies that $\mathbf{P}(E_r \leq \epsilon) \approx 1 - \alpha$ if

$$n = \frac{\sigma^2 z_{1-\alpha/2}^2}{\theta^2 \epsilon^2}.$$

There are still some problems, however:

1. When we are controlling E_r , we need to know σ and θ in advance.
2. When we are controlling E_a , we need to know σ in advance.

Of course we do not usually know σ or θ in advance. In fact, θ is what we are trying to estimate! There are two methods we can use to overcome this problem: the *two-stage method* and the *sequential method*, both of which we will now describe.

2.1 The Two Stage Procedure

Suppose we want to satisfy the condition $\mathbf{P}(E_a \leq \epsilon) = 1 - \alpha$ so that we are trying to control the absolute error. Then we saw earlier that we would like to set

$$n = \frac{\sigma^2 z_{1-\alpha/2}^2}{\epsilon^2}.$$

Unfortunately, we don't know σ^2 but we can solve this problem by first doing a *pilot* simulation to estimate it. The idea is to do a small number, p , of initial runs to estimate σ^2 . We then use our estimate, $\hat{\sigma}^2$, to compute an estimate, \hat{n} , of n . Finally, we repeat the simulation, but now we use \hat{n} runs. We have the following algorithm.

Two-Stage Monte Carlo Simulation for Estimating $E[h(\mathbf{X})]$

```

/* Do pilot simulation first */
for i = 1 to p
    generate  $\mathbf{X}^i$ 
end for
set  $\hat{\theta} = \sum h(\mathbf{X}^i)/p$ 
set  $\hat{\sigma}^2 = \sum (h(\mathbf{X}^i) - \hat{\theta})^2/(p-1)$ 
set  $n = \frac{\hat{\sigma}^2 z_{1-\alpha/2}^2}{\epsilon^2}$ 

/* Now do main simulation */
for i = 1 to n
    generate  $\mathbf{X}^i$ 
end for
set  $\hat{\theta}_n = \sum h(\mathbf{X}^i)/n$ 
set  $\hat{\sigma}_n^2 = \sum (h(\mathbf{X}^i) - \hat{\theta}_n)^2/(n-1)$ 
set 100(1 -  $\alpha$ ) % CI =  $\left[ \hat{\theta}_n - z_{1-\alpha/2} \frac{\hat{\sigma}_n}{\sqrt{n}}, \hat{\theta}_n + z_{1-\alpha/2} \frac{\hat{\sigma}_n}{\sqrt{n}} \right]$ 

```

For this method to work, it is important that $\hat{\theta}$ and $\hat{\sigma}^2$ be sufficiently good estimates of θ and σ^2 . Therefore, it is important to make p sufficiently large. In practice, we usually take $p \geq 50$. We can use an analogous two-stage procedure if we want to control the *relative* error and have $\mathbf{P}(E_r \leq \epsilon) = 1 - \alpha$.

Example 2 (Pricing a European Call Option Revisited)

Recall Example 1 where we estimated the price of a call option and computed an approximate 95% CI for the price. Suppose we now want to control the absolute error so that

$$\mathbf{P}(E_a \leq .05) = 1 - \alpha.$$

Note that this is equivalent to saying that we want our 100(1 - α)% CI to have width less than or equal to $2 \times 5 = 10$ cents. Let us use the two stage simulation method to choose n , using $p = 100$ and $\alpha = .05$.

Matlab Code for Estimating the BS Option Price

```

> S0=100; K=110; T=0.5; r=0.5; sigma=.2;
> p=100;
> % p is the number of samples we use for the pilot simulation

> Z=sqrt(T)*randn(p,1);
> ST = S0*exp((r-(sigma^2)/2)*T + sigma*Z);
> Payoff = exp(-r*T)*max(0,ST-K);
> % Payoff is the vector of h(X) values in our notation

> n = (std(Payoff)^2 * (1.96^2)) / (.05^2)

    n = 191719
> n = ceil(n);
> % End of pilot survey. Now start main simulation

> Z=sqrt(T)*randn(n,1);
> ST = S0*exp((r-(sigma^2)/2)*T + sigma*Z);
> Payoff = exp(-r*T)*max(0,ST-K);
> C_est = mean(Payoff) % Our estimated price

    C_est = 15.2372

> epsilon = 1.96*std(Payoff)/sqrt(n);
> CI_95 = [C_est - epsilon, C_est + epsilon]

    CI_95 = 15.1794 15.2950

```

Observe that the width of the approximate 95% CI is indeed approximately 10 cents!

Example 3 (Portfolio Evaluation Revisited)

Recall our earlier portfolio evaluation example. We would like to estimate $\theta := \mathbf{P}(L) = E[I_L]$ where L is the event that $W_T/W_0 \leq .9$. Suppose we would like to compute an approximate 95% CI for θ with a half-width equal to .01. Then, using a pilot study with $p = 100$, we can do this as follows.

A Matlab Function for Estimating $\mathbf{P}(W_T/W_0 \leq .9)$

```
function[CI] = portfolio_evaluation_pilot(p,rho)

% This function estimates the probability that wealth of the portfolio
% falls by more than 10%. We construct an approx. 95% CI for this
% probability that has half-width approx. equal to .001

mua=.15; mub=.12; siga=.2; sigb=.17; T=1;
S0a=100; S0b=120; na=100; nb=100; za=1.96; epsilon = .001;
W0 = na*S0a + nb*S0b;
Sigma = [siga^2 siga*sigb*rho; siga*sigb*rho sigb^2];

% First do the pilot simulation
B=randn(2,p); C=chol(Sigma); V=C'*B;

STa = S0a *exp((mua - (siga^2)/2)*T + sqrt(T)*V(1,:));
STb = S0b *exp((mub - (sigb^2)/2)*T + sqrt(T)*V(2,:));

WT = na*STa + nb*STb; L = [WT/W0 < .95];
n = ceil((std(L)^2 * (za^2)) / (epsilon^2))

% Now do the main simulation

B=randn(2,n);

STa = S0a *exp((mua - (siga^2)/2)*T + sqrt(T)*V(1,:));
STb = S0b *exp((mub - (sigb^2)/2)*T + sqrt(T)*V(2,:));

WT = na*STa + nb*STb; theta = mean(WT/W0 < .95);
L = [WT/W0 < .9]; epsilon = za*std(L)/sqrt(n);
CI = [theta - epsilon, theta + epsilon];
```

2.2 The Sequential Procedure

Suppose again that we wish to satisfy the condition $\mathbf{P}(E_a \leq \epsilon) = 1 - \alpha$. Then we saw earlier that we would like to set

$$n = \frac{\sigma^2 z_{1-\alpha/2}^2}{\epsilon^2}.$$

In contrast to the pilot procedure, we do not precompute n during the sequential procedure. Instead, we continue to generate samples until

$$\frac{\widehat{\sigma}_n z_{1-\alpha/2}}{\sqrt{n}} \leq \epsilon$$

where $\widehat{\sigma}_n$ is again the estimate of σ based upon the first n samples. It is important that n be sufficiently large so that $\widehat{\theta}_n$ and $\widehat{\sigma}_n^2$ are sufficiently good estimates of θ and σ^2 , respectively. As a result, we typically insist that $n \geq 50$ before we stop. Approximate confidence intervals are then computed as usual.

Question: Have we allowed any biases to creep in here?

We have the following algorithm:

Sequential Monte Carlo Simulation for Estimating $E[h(\mathbf{X})]$

```

set check = 0, n = 1
while (check = 0)
  generate  $\mathbf{X}^n$ 
  set  $\widehat{\theta}_n = \sum h(\mathbf{X}^i)/n$ 
  set  $\widehat{\sigma}_n^2 = \sum (h(\mathbf{X}^i) - \widehat{\theta}_n)^2/(n-1)$ 
  if ( $n \geq p$ ) and  $\left( \frac{\widehat{\sigma}_n z_{1-\alpha/2}}{\sqrt{n}} \leq \epsilon \right)$ 
    check = 1
  else
    n = n + 1
  end if
end while
set 100(1 -  $\alpha$ ) % CI =  $\left[ \widehat{\theta}_n - z_{1-\alpha/2} \frac{\widehat{\sigma}_n}{\sqrt{n}}, \widehat{\theta}_n + z_{1-\alpha/2} \frac{\widehat{\sigma}_n}{\sqrt{n}} \right]$ 

```

In practice we do **not** need to store every value, $h(\mathbf{X}^i)$ for $i = 1, \dots, n$, in order to update $\widehat{\theta}_n$ and $\widehat{\sigma}_n$. Indeed, we can update $\widehat{\theta}_n$ and $\widehat{\sigma}_n$ efficiently by observing that

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} + \frac{h(\mathbf{X}^n) - \widehat{\theta}_{n-1}}{n} \quad \text{and}$$

$$\widehat{\sigma}_n^2 = \left(\frac{n-2}{n-1} \right) \widehat{\sigma}_{n-1}^2 + n \left(\widehat{\theta}_n - \widehat{\theta}_{n-1} \right)^2.$$

Remark 2 If we want to control the relative error and have $\mathbf{P}(E_r \leq \epsilon) = 1 - \alpha$, then we would simulate samples until

$$\frac{\widehat{\sigma}_n(z_{1-\alpha/2})}{\widehat{\theta}_n \sqrt{n}} \leq \epsilon.$$

Example 4 (Pricing a European Call Option Once More)

We now price the same call option as before, but this time using the sequential approach.

A Matlab Function for Estimating $P(W_T/W_0 \leq .9)$

```
function[CI] = Eurocall_sequential(p,epsilon)

% This function uses the sequential algorithm for estimating a 95% CI for
% the price of a European call option. At least p runs are made.

S0=100; K=110; T=0.5; r=0.5; sigma=.2; za=1.96;
old_theta = 0; sig_square = 0; i=1; check=0;
con1 = S0*exp((r-(sigma^2)/2)*T);
con2=exp(-r*T);

while (check == 0)

    S = con1*exp(sigma*sqrt(T)*randn);
    Payoff = con2*max(0,S-K);
    new_theta = old_theta + (Payoff - old_theta)/i;
    if (i > 1)
        sig_square = sig_square*(i-2)/(i-1) + i*(new_theta - old_theta)^2;
    end;
    if ((i >= p) & (za*sqrt(sig_square/i) < epsilon))
        check = 1;
    else
        i = i+1;
        old_theta=new_theta;
    end;
end;

CI = [new_theta - za*sqrt(sig_square/i), new_theta + za*sqrt(sig_square/i)];
```

Remark 3 *When using Matlab, it is best to avoid loops as much as possible as they tend to decrease the computational speed, sometimes significantly. Instead, it is better to vectorise where possible, and indeed this is what we have been doing in most of the Matlab examples so far.*