# Controlling and Accelerating Convergence

*Jonathan Navarrete*

*May 29, 2017*

Topics to be covered:

1. Monitoring Convergence
2. Antithetic Variables

## Antithetic Variables

In previous experiments, when we've worked to generate pseudo-random samples from distributions, we've worked with *iid* (independent and identically distributed) peuso-random samples from an instrumental distribution. Generally, *iid* samples are always preferable, but not always cost efficient. As problems become more complicated, generating random samples from a target distribution will become more cumbersome and time/resource consuming. Therefore, in this section we will present methods in which we can double down on our generated samples to speed up convergence and utilize more of our available resources.

The method of antithetic variables is based on the idea that higher efficiency can be obtained through correlation. Given who samples $X = (x_1, ..., X_n)^T$ and $Y = (y_1, ..., y_n)^T$ from the distribution $f$ used in monte carlo integration.

The monte carlo integration estimator

$$\theta = \int_{-\infty}^{\infty} h(x) f(x) dx$$

If $X$ and $Y$ are negatively correlated, then the estimator $\hat{\theta}$ of $\theta$

$$\hat{\theta} = \frac{1}{2n} \sum_{i=1}^{n} [h(x_i) + h(y_i)]$$

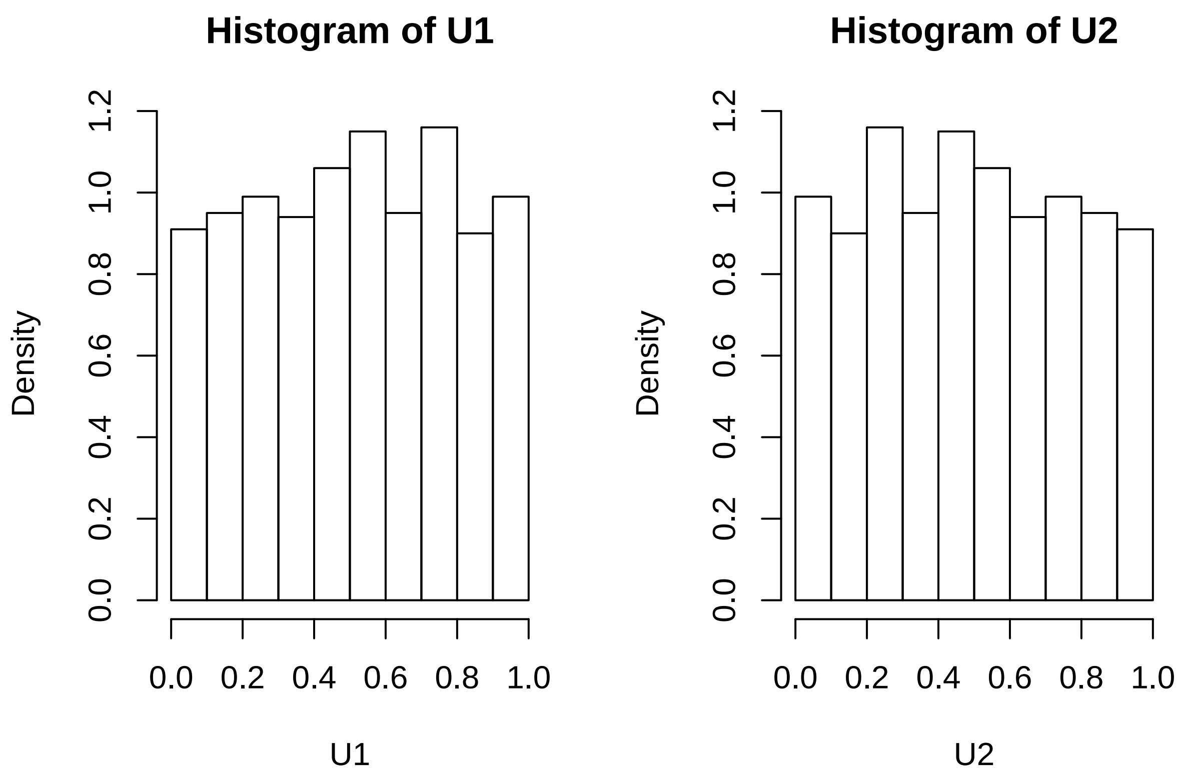
is more efficient than the estimator $\hat{\theta} = \frac{1}{2n} \sum_{i=1}^{2n} h(x_i)$. The random variables $X$ and $Y$ are then called *antithetic variables*.

Albeit useful, this method is not always possible. For arbitrary transformations $h(.)$, it is not always possible to generate negatively correlations $X$ and $Y$.

As covered in the introduction, we can generate negatively correlated samples from a uniform distribution.

```
U1 = runif(1000)
U2 = (1 - U1)

par(mfrow = c(1,2))
hist(U1, probability = TRUE)
hist(U2, probability = TRUE)
```




```r
par(mfrow = (c(1,1)))

print(cor(U1, U2))
```

```
## [1] -1
```

**Exercise 5.6**

Covariance:

$$
\begin{aligned}
Cov(e^U, e^{1-U}) &= E[e^U e^{1-U}] - E[e^U]E[e^{1-U}] \\
&= E[e^1] - (e-1)E[e^{1-U}] \\
&= E[e^1] - (e-1)E[e^1 e^{-U}] \\
&= e - (e-1)e^1[\frac{e-1}{e}] \\
&= e - (e-1)^2 = -0.23421
\end{aligned}
$$

$$
\begin{aligned}
Var(e^U + e^{1-U}) &= var(e^u) + var(e^{1-U}) + 2Cov(e^U, e^{1-U}) \\
&= E[e^{2U}] - E[e^U]^2 + E[e^{2-2U}] - E[e^{1-U}]^2 + 2Cov(e^U, e^{1-U}) \\
&= \frac{e^2-1}{2} - (e-1)^2 + E[e^2 e^{-2U}] - E[e^1 e^{-U}]^2 + 2Cov(e^U, e^{1-U}) \\
&= \frac{e^2-1}{2} - (e-1)^2 + \frac{e^2-1}{2} - (e-1)^2 + 2Cov(e^U, e^{1-U}) \\
&= -1 - 2(e-1)^2 + e^2 + 2Cov(e^U, e^{1-U}) \\
&= -1 - 2(e-1)^2 + e^2 + 2(-0.23421) \\
&= 0.0156512
\end{aligned}
$$

Therefore, the variance reduction is approximately 96%

### Exercise 5.7

Refer to Exercise 5.6. Use a Monte Carlo simulation to estimate $\theta$ by the antithetic variate approach and by the simple Monte Carlo method. Compute an empirical estimate of the percent reduction in variance using the antithetic variate. Compare the result with the theoretical value from Exercise 5.6.

For this example, $g(U) = e^U$. Simple Monte Carlo Method:

```
set.seed(6)
m = 10000
U = runif(m)
g = exp(U)

theta = mean(g) ## theta for simple MC
theta
```

```
## [1] 1.721562
```

```
var_theta1 = var(g)/m
```

```
set.seed(6)
m = 5000
U = runif(m)
T1 = exp(U)
T2 = exp(1-U)

cov(T1, T2)
```

```
## [1] -0.2317587
```

```
c = (1/2)

anti_thetic = c*mean(T1) + (1-c)*mean(T2) ## Antithetic Control Variate
anti_thetic
```

```
## [1] 1.717577
```

```
##variance of theta2
var_theta2 = var(T2)/m + c**2 * var(T1 - T2)/m + 2*c*cov(T2, T1 - T2)/m

(var_theta1 - var_theta2) / var_theta1
```

```
## [1] 0.9678411
```

The true value of $\theta$ is 1.718282. The antithetic control variate estimator came closest to the true value, and has a extremely low variance, a 96.78% reduction.

### Exercise 5.8

Let $U \sim Uniform(0, 1)$, $X = aU$, and $X' = a(1 - U)$, where a is a constant. Show that $\rho(X, X') =$ -1.

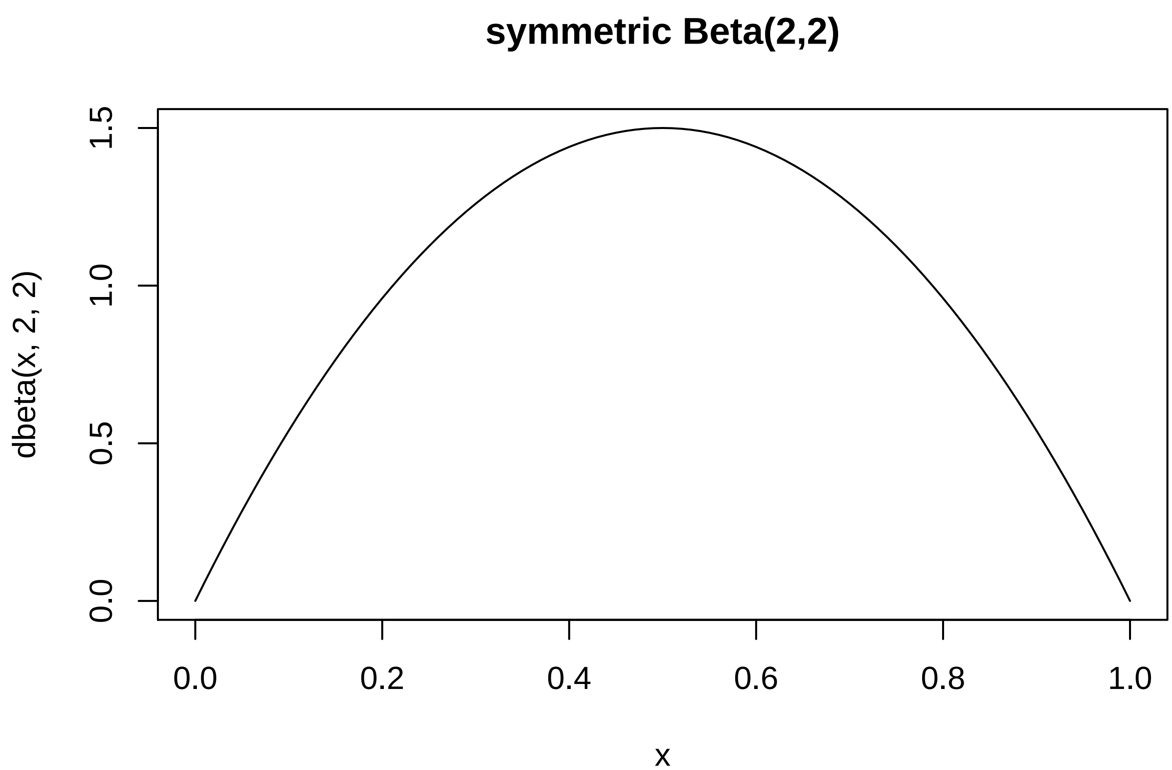
Note that, since $U \sim Unif(0, 1)$ then $(1 - U)$ is also $Unif(0, 1)$ distributed. $E[U] = 1/2$ and $Var(U) = 1/12$

$$
\begin{aligned}
Cov(X, X') &= a^2 Cov(U, 1-U) \to a^2 E[U(1-U)] - E[U]E[1-U] \\
&= a^2 E[U - U^2] - (1/2)(1/2) \to a^2(E[U] - E[U^2] - 1/4) \\
&= a^2(1/2 - E[U^2] - 1/4) \to a^2(1/4 - E[U^2]) \\
&= a^2(1/4 - 4/12) \\
&= a^2(-1/12)
\end{aligned}
$$

$$
\begin{aligned}
\rho &= \frac{Cov(X, X')}{\sqrt{Var(X)}\sqrt{Var(X')}} \\
&= \frac{a^2(-1/12)}{\sqrt{a^2(1/12)}\sqrt{a^2(1/12)}} \\
&\to \frac{a^2(-1/12)}{a^2(1/12)} = -1
\end{aligned}
$$

Is $\rho(X, X')$ = -1 if U is a symmetric beta random variable? Yes, we can show this computationally.

```
curve(expr = dbeta(x, 2, 2), from = 0, to = 1, main = "symmetric Beta(2,2)")
```




```
U = rbeta(1000, 2, 2)
cor(U, 1-U)
```

```
## [1] -1
```

**Execise 6.4**

```
set.seed(5)
m = 20
```

```r
LN_mu = function(m = 20, mu = 2, sigma = 4) {
  X = rlnorm(m, meanlog = mu, sdlog = sigma)
  CI = t.test(log(X), mu = mu, conf.level = 0.95)
  return(c(CI$conf.int[1], CI$conf.int[2]))
}

ConfIntervals = matrix(replicate(n = 1000, expr = LN_mu()), nrow = 1000, byrow = TRUE)

successes = sum((ConfIntervals[,1] < 2) & (2 < ConfIntervals[,2]))
## proportion of successes
mean((ConfIntervals[,1] < 2) & (2 < ConfIntervals[,2]))
```

```
## [1] 0.956
```

**Exercise 6.5**

```r
set.seed(5)
LN_mu2 = function(m = 20, mu=2) {
  X = rchisq(m, df = 2) ## data is now Chisquare(2)
  CI = t.test(X, mu = mu, conf.level = 0.95)
  return(c(CI$conf.int[1], CI$conf.int[2]))
}

ConfIntervals = matrix(replicate(n = 1000, expr = LN_mu2()), nrow = 1000, byrow = TRUE)

successes = sum((ConfIntervals[,1] < 2) & (2 < ConfIntervals[,2]))
## proportion of successes
mean((ConfIntervals[,1] < 2) & (2 < ConfIntervals[,2]))
```

```
## [1] 0.917
```

We see that the result for when the data originates from a non-Normal distribution results in a lower rate of success than when the data is more appropriately Normal, like in Exercise 6.4.