Examples

The examples in the text related to this material aren't to my liking. With that in mind, I've put together two examples which hopefully add some clarity to the principles discussed in this module.

Example 1 This is 4.15(a). Recall that a negative binomial random variable is the number of success/no-success trials that occur until the r^{th} success. Assuming that a success occurs with probability p, the probability mass function of a negative binomial random variable is $X \sim Negbin(r, p)$, where

$$P(X = x) = \binom{x-1}{r-1} p^r (1-p)^{x-r},$$

with $E(X) = \mu = r(1-p)/p$ and $\text{Var}(X) = \mu + \mu^2/r^2$. In `R`, we can generate values of a negative binomial with the `rnegbin`(n, μ, r) function. Below I am generating 1000 values of X where $X \sim Negbin(5, .5)$.

`rnegbin(1000, 5, 5).`

The mean I get is `mean(x)` $= 5.032$ and the standard deviation of this mean is `sd(x)/sqrt(1000)` $=$ .0999.

I can also generate values of X by taking advantage of the fact that if

$$X|y \sim \text{Poi}(y), \quad \text{and} \quad Y \sim \text{Gamma}(r, (1-p)/p),$$

then $X \sim Negbin(r, p)$. I can take advantage of this relationship and use Rao-Blackwellization to generate an estimate of the mean with a smaller standard deviation. To do this, I:

$E[\text{Pois}(\lambda)] = \lambda$

1. First generate $Y \sim Gamma(r, (1-p)/p)$ with `y = rgamma(1000, 5, 1)`.
2. Now take advantage of the fact that $E(X|Y = y) = y$. WIth this in mind, we can apply Rao-Blackwellization and estimate the mean of X as being `mean(y)`. I get this to be 4.97, and the standard deviation is `sd(y)/sqrt(1000)` $= .069$.

Example 2 Let's assume that we would like to use Monte Carlo Methods to estimate the expected value of an exponential distribution with parameter λ. Recall that the exponential distribution is $f(x) = \lambda \exp(-\lambda x)$, and that $F(x) = 1 - \exp(-\lambda x)$. With this in mind, one would generate n values of U, such that $U_1, U_2, \ldots, U_n \sim \text{Unif}(0, 1)$, and from this calculate n values of X and n values of Y, where $X_i = F^{-1}(U_i)$ and $Y_i = F^{-1}(1 - U_i)$. The estimated expected value would then be

$$\widehat{E(X)} = \frac{1}{2n} \sum_{j=1}^{n} (X_i + Y_i).$$

Below is an `R` program that does this for an arbitrary λ Run this code and make sure it makes sense to you.

```
antithetic_Exp = function(n, lambda) {
    U = runif(n,0,1);
    X = -log(1-U)/lambda;
    Y = -log(U)/lambda;
    estimated_mean = mean(c(X,Y));
    print('estimated mean')
    print(estimated_mean);
}
```

Example 3 Let's assume that we want to estimate $P(X > a) = \int_a^\infty f(x)dx$ where $f(x)$ is the Cauchy density. We know the density is symmetric about 0, making $P(X > 0) = .5$, so it's reasonable to estimate the probability above using control variates. We can simulate (in `R`) $X_1, X_2, \ldots, X_n$ from a Cauchy distribution, and calculate $\hat{I}_1 = \frac{1}{n}\sum_{j=1}^n \mathbf{1}(X_j \geq a)$ and $\hat{J}_1 = \frac{1}{n}\sum_{j=1}^n \mathbf{1}(X_j > 0)$. The estimator for the probability we want then becomes

$$\hat{I}_2 = \hat{I}_1 + \beta(\hat{J}_1 - .5).$$

Now we have to pick β. Remember that the "optimal" β was $\beta = -\text{Cov}(\hat{J}_1, \hat{I}_1)/\text{Var}(\hat{J}_1)$. It is easy to show that $\text{Var}\left(\hat{J}_1\right) = \frac{1}{n}0.5 \times 0.5$ (prove that to yourself if you don't know that), and the covariance is calculated as

$$\text{Cov}(\hat{I}_1, \hat{J}_1) = E\left(\hat{J}_1\hat{I}_1\right) - E\left(\hat{J}_1\right)E\left(\hat{I}_1\right) = \frac{1}{n}P(X > a)P(X > 0),$$

so if you kind-of sort-of know what $P(X > a)$ is and you choose β accordingly, you should be ok. Below is an `R` example where I set $a = 2.$, and I'm going to guess that the probability X is greater than 2 is .2. Run the code and make sure it makes sense to you.

```
controlVariablesExmpl = function(n) {
    estimate = c(rep(0,n));
    for (j in 1:n) {
      randomSample = rcauchy(n)
      I_hat1_vec = randomSample;
      I_hat1_vec[I_hat1_vec < 2] = 0;
      I_hat1_vec[I_hat1_vec > 2] = 1;
      I_hat1 = mean(I_hat1_vec);
      J_hat1_vec = randomSample;
      J_hat1_vec[J_hat1_vec < 0] = 0;
      J_hat1_vec[J_hat1_vec > 0] = 1;
      J_hat1 = mean(J_hat1_vec);
      beta = -(.2*.5/n)/(.5*.5/n);
      I_hat2 = I_hat1 + beta*(J_hat1 - .5);
      estimate[j] = I_hat2;
    }
    print(var(estimate));
    plot(estimate, type = "l");
}
```