

Exam Final

IFT6561

Gabriel C-Parent

DIRO
UdeM

Part 1

a

Experimental Results

The algorithm is implemented in San13CMC_CV.java.

Beta vector with CV

(0.00628, 0.000347, 0.00138, 0.00501, 0.00458, 0.00271, 0.00276, 0.00703)

Average without CV : 0.11448

Average with CV : 0.11291

Variance without CV : 0.025091

Variance with CV : 0.010342

Variance reduction factor : 2.4261

Confidence interval student 95 without CV : 0.11448 ± 0.0031046

Confidence interval student 95 with CV : 0.11291 ± 0.0019932

As shown in the data, the confidence interval with control variates (CV) is about 1.5 times tighter than without CV. The average of the estimator without CV is also contained within the 95% student confidence interval of the estimator with CV. Overall, the strategy using CV is pretty good, decreasing the variance by a factor of 2.4. This was expected, as stated in example 6.26.

b

The whole concept of stochastic derivatives is explained at p.101 of the book, I'm not rewriting it here, we'll just state the core of the matter.

The first necessary observation is that both θ_2 and θ_4 are parameters for the mean of exponential distributions. Using the same proof as given in example 1.49 of the book, we can explain why our estimator is unbiased.

Suppose we want to estimate $\frac{\partial \mathbb{E}[T]}{\partial \theta_i}$, we can write T as a function of θ_i .

Say f is the function. $f'(\theta_i, U) = V'_i(\theta_i)$ if the link i is in the longest path and 0 otherwise. Since, in this case, both θ_i are the means of exponential random variables, we have $Y_i = Y_i(\theta_i) = -\theta_i \ln(1 - U_i)$ and its derivative is $-\ln(1 - U_i)$. Using the corollary 6.6 to prove that we can interchange the

expectation and derivative operators to obtain that our estimator of $f'(\theta_i)$ is unbiased.

Basically, this is the rest of the example 1.49.

c

As stated in example 1.50 of the book, if the function $f_j(\theta_j, U) = \mathbb{1}[T > x]$, the estimator can only take two values, 0 and 1. Its derivative is either undefined, when the function jumps exactly at θ_j or is 0. It cannot be an unbiased estimator of the derivative because the original function is discontinuous at some point as a function of θ_j .

d

The process is explained in [1], beginning of p.17.

First thing, we need to condition on all activities except the ones depending on θ_2 and θ_4 . Then we proceed using equation (14) of [2] for both θ_2 and θ_4 . Further details are given in section 3 of [2].

I don't know how to formally prove it, but informally, we need to prove that the derivative of the cumulative functions of the activities (which are exponentials in this particular case) are continuous, unlike the indicator function.

e

Sadly, I didn't have enough time to implement the algorithm.

f

This is the same as in [2], p. 361. Basically, this is the problem of estimating

$$\frac{\delta P(Y > \theta)}{\delta \theta}$$

We first condition on a set of activity times s.t. the set and its complement have non-zero probability of having an activity on the critical path (that is by obtaining a minimal cut set, as in the class notes).

We then condition on the complement set of activity times and take the sample path derivative [2].

The choice of cut set is not trivial.

To give an example of why multiple values of threshold can be calculated at the same time, I refer to the example (i) of p.362 [2]. Clearly, the value of θ can be substituted for all the other desired thresholds. This allows us, for each state of the system, to query all the desired values of θ in parallel without the need of generating more pseudo-random variables. Sadly, I didn't have enough time to implement the algorithm.

g

Sadly, I didn't have enough time to implement the algorithm.

2

a

The algorithm was implemented in AsianVG_RQMC.java.

The point sets (PS) are in corresponding order: Sob + S, Sob + LMS + S, Kor + S, Kor + S + B

PS1 BGSS : 25

PS1 BGBS : 57

PS1 DGBS : 297

PS2 BGSS : 12

PS2 BGBS : 77

PS2 DGBS : 561

PS3 BGSS : 10

PS3 BGBS : 18

PS3 DGBS : 116

PS4 BGSS : 12

PS4 BGBS : 127

PS4 DGBS : 118

The results for Sob+S are lower than expected, especially for the bridge sampling, but the results for Sob + LMS + S match quite well with table 6.12.

For the Korobov lattice, Kor+S matches very well the expected results while Kor+S+B yields much greater variance reduction factor.

I think the cause of this is perhaps an error in the calculation of the variance reduction factor, or otherwise the variation is too important.

The important point is that we reach the same observation that DGBS usually yields superior reduction of variance.

b

$$\pi(x) = \frac{\lambda^\alpha x^{\alpha-1} \exp(-\lambda x)}{\Gamma(\alpha)}$$

$$\pi_\theta(x) = \frac{\exp(\theta x) \pi(x)}{\mathbb{M}(\theta)}$$

where \mathbb{M} is the mgf of the Gamma distribution. The exponentially twisted pdf is then

$$\pi_\theta(x) = \frac{\exp(\theta x)}{(1 - \frac{\theta}{\lambda})^{-\alpha}} \frac{\lambda^\alpha x^{\alpha-1} \exp(-\lambda x)}{\Gamma(\alpha)}$$

this yields the following new densities

$$\pi_\theta(x) = \frac{\exp(-(\lambda - \theta)) x^{\alpha-1}}{\Gamma(\alpha) (\lambda - \theta)^{-\alpha}}$$

We can generate pseudo-random variable by sampling from the new distributions $\text{Gamma}(\alpha, \lambda - \theta)$, since only the scale parameter was changed.

The likelihood ratio to multiply by will be the product of the individual likelihoods:

$$L(x) = \exp(-\theta x + \alpha \ln(\frac{\lambda}{\lambda - \theta}))$$

$$L(\omega) = L(x_1) * L(x_2)$$

We must simply apply these to the Gamma processes (substituting for the real parameters).

c

The algorithm was implemented in AsianVG1.java.

Since I couldn't find the relation between $\mathbb{E}[G^+(1) - G^-(1)]$ and θ , this one isn't working.

What I got is that I know the $\mathbb{E}[G^+(1) - G^-(1)]$ is equal to the variance gamma θ_{VG} parameter (by definition). But I can't seem to figure it out going back from the new $\lambda^* = \lambda \pm \theta$ because both terms of the gamma distributions should be related to the original value of θ_{VG} , I mean both the μ and ν terms are related to it so I don't see how to do it this way. Otherwise, I could probably try to use the relation suggested here <http://stats.stackexchange.com/questions/48378/difference-of-gamma-random-variables>, but I'm not sure how to proceed. Since the next many questions depend on it, well, you got me good.

d

Yeah, hard to compare with c) on that one.

e

The algorithm was implemented as AsianVGTronquee.java. As I wasn't able to derive the formula relating the average difference of gamma to the added θ term, it's pretty hard to compare. However, the code does yield a value similar to the one expected when simulating with BGSS for $K=140$ and 180 , so I guess it isn't so bad.

f

Sadly, I didn't have enough time to implement the algorithm.

Bibliography

- [1] Michael Fu. Gradient estimation. In *Handbook in Operations Research and Management Science*, 13. 2006.
- [2] Michael C. Fu. Sensitivity analysis in monte carlo simulation of stochastic activity networks. In *Perspectives in Operations Research*, pages 351–366. Springer, 2006.