ABSTRACT

VARIANCE-REDUCTION TECHNIQUES FOR ESTIMATING QUANTILES AND VALUE-AT-RISK

by
Fang Chu

Quantiles, as a performance measure, arise in many practical contexts. In finance, quantiles are called values-at-risk (VARs), and they are widely used in the financial industry to measure portfolio risk. When the cumulative distribution function is unknown, the quantile can not be computed exactly and must be estimated. In addition to computing a point estimate for the quantile, it is important to also provide a confidence interval for the quantile as a way of indicating the error in the estimate. A problem with crude Monte Carlo is that the resulting confidence interval may be large, which is often the case when estimating extreme quantiles. This motivates applying variance-reduction techniques (VRTs) to try to obtain more efficient quantile estimators. Much of the previous work on estimating quantiles using VRTs did not provide methods for constructing asymptotically valid confidence intervals.

This research developed asymptotically valid confidence intervals for quantiles that are estimated using simulation with VRTs. The VRTs considered were importance sampling (IS), stratified sampling (SS), antithetic variates (AV), and control variates (CV). The method of proving the asymptotic validity was to first show that the quantile estimators obtained with VRTs satisfies a Bahadur-Ghosh representation. Then this was employed to prove central limit theorems (CLTs) and to obtain consistent estimators of the variances in the CLTs, which were used to construct confidence intervals. After the theoretical framework was established, explicit algorithms were presented to construct confidence intervals for quantiles when applying IS+SS, AV and CV. An empirical study of the finite-sample behavior of the confidence intervals was also performed on two stochastic models: a standard normal/bivariate normal distribution and a stochastic activity network (SAN).

VARIANCE REDUCTION TECHNIQUES
FOR ESTIMATING QUANTILES AND VALUE-AT-RISK

by
Fang Chu

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Information Systems

Department of Information Systems

May 2010

APPROVAL PAGE

VARIANCE REDUCTION TECHNIQUES
FOR ESTIMATING QUANTILES AND VALUE-AT-RISK

Fang Chu

4-28-10

Dr. Marvin Nakayama, Dissertation Advisor — Date
Associate Professor of Computer Science and Information Systems, NJIT

4/28/10

Dr. Renraw Chen, Committee Member — Date
Professor of Finance, Fordham University

4/28/10

Dr. George Widmeyer, Committee Member — Date
Associate Professor of Information Systems, NJIT

Dr. Julian Scher, Committee Member — Date
Associate Professor of Information Systems, NJIT

4/28/10

Dr. James Calvin, Committee Member — Date
Associate Professor of Computer Science, NJIT

4/28/10

Dr. Michael Recce, Committee Member — Date
Associate Professor of Information Systems, NJIT

BIOGRAPHICAL SKETCH

Author: Fang Chu

Degree: Doctor of Philosophy

Date: May 2010

Undergraduate and Graduate Education:

- Doctor of Philosophy in Information Systems,
  New Jersey Institute of Technology, Newark, New Jersey, 2010

- Bachelor of Science in Computer Science (Software Engineering),
  Tianjin University, China, 2005

Major: Information Systems

Publications:

Chu, F., and Nakayama, M. K., *Constructing Confidence Intervals for Quantiles When Using Variance-Reduction Techniques*, extended abstract accepted for 8th International Workshop on Rare Event Simulation June 2010, Cambridge, UK

Chu, F., and Nakayama, M. K., *Confidence Intervals for Quantiles Estimated Using Importance Sampling*, 2010, invited and to be submitted to *Proceedings of the 2010 Winter Simulation Conference.*

Chu, F., and Nakayama, M. K., *Confidence Intervals for Quantiles When Applying Variance-Reduction Techniques*, 2010, to be sumbitted to a journal.

To My Beloved Wife, Parents and Grandparents

ACKNOWLEDGMENT

I owe my deepest gratitude to Dr. Marvin Nakayama, my dissertation advisor. I have been incredibly fortunate to have an advisor who fully understood my career goal and wholeheartedly helped me realize it. His mentorship was paramount in encouraging me to become a researcher, and most importantly, an independent thinker. For everything you have done for me, Dr. Nakayama, thank you.

I would also love to thank all of my committee members for their guidance and support. Dr. George Widmeyer, in his decision-support systems class, had inspired me to explore the simulation applications in business, which eventually developed into the framework of my research topic. Dr. Renraw Chen has not only introduced me to quantitative finance with his enlightening lectures, but also motivated me to conduct research and pursue a career in this challenging field with enthusiasm. Without the valuable feedback and enormous assistance from Dr. Julian Scher, Dr. Michael Recce and Dr. James Calvin, this work would never had been completed.

I would like to take the opportunity to thank Dr. Michael Bieber and Dr. Brook Wu for getting my graduate career started on the right foot and providing me the foundation for becoming a researcher. Thanks also go out to all those faculty members who brought my understanding of information systems to a higher level through their passion in research and teaching. I would also like to extend my gratitude to all the colleagues in the two IS labs.

A very special thank you to my friends Yao Hu and Qiming Wang for the support and assistance they have lent me over years. Thanks a lot for everything. Thanks also go out to Li Yuan, Ye Yang, Tao Wu, Wen Deng, Wei Yu, Xiaowei Jiang, Rong Cui, Xiaoya Liu, Yixuan Li, Robert Baksa, Kung-E Cheng, Ali Avni, Maria Plummer, Feiyan Chen, Peixin Zhang, Yuan Yuan and many "meal plan" buddies for providing so much needed humor and entertainment in what could have been otherwise a somewhat stressful Ph.D.

life.

Finally, and most importantly, I would like to thank my wife Lingyan Shi. Her support, encouragement, quiet patience and unwavering love were undeniably the bedrock upon which the past five years of my life have been built. Her unyielding devotion and love have saved me from the most depressing moments in my life. It was under her watchful eye that I gained so much drive and an ability to tackle challenges head on. I thank my parents and grandparents for their faith in me and allowing me to be as ambitious as I wanted. Also, I thank Lingyan's parents for allowing me to take their daughter over six thousands miles away from the "Paradise City" in China and providing me with unending encouragement and support.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF TABLES
(Continued)

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

Many systems and processes exhibit complex behaviors and operate in uncertain environments. Examples include computer networks, database systems, telecommunication networks, transportation systems, and financial markets. Random events, such as failures of components, unknown future demands and prices, and natural disasters, are inherent to these systems. A system designer must determine the performance of a proposed system before implementing it, and an operator needs to estimate the effect of management decisions on future performance. To better understand how such a system or process behaves, one often will build a mathematical model. When the studied process or system includes randomness, the model is often a stochastic process.

Discrete-event stochastic simulation and Monte Carlo simulation (e.g., Law (2006)) are powerful tools in science and engineering for studying the behavior of stochastic systems that are too complicated to allow for direct mathematical analysis. While analytical methods often require simplifying and unrealistic assumptions when applied to complex systems or processes, the simulation approach allows the user to incorporate as much detail and uncertainty as needed to model accurately the system or process under study. Simulation of the model entails generating random samples from the probability distributions in the model to imitate the stochastic behavior of the system or process over time. The simulation produces output data, which is collected and analyzed.

Because a simulation includes randomness, its output is also random, thus necessitating the use of statistical methods to analyze the output. Most previous work on statistical analysis of simulation output focuses on estimating the mean performance of the system under study. When considering the mean performance of the system over a finite time horizon, the standard approach is to run independent and identically distributed (i.i.d.) replications of the system over the finite time horizon, and then use classical statistical methods (e.g., Hogg et al. (2004)) to analyze the outputs from the replications.

More specifically, let X be a random variable denoting the (random) performance of a stochastic system over a finite time horizon, and suppose that we are interested in computing the mean performance $\alpha = E[X]$, where E denotes expectation. For example, in project planning, X may represent the (random) time to complete a project, which is modeled as a stochastic activity network, and we are interested in computing the mean time to complete the project. Running n i.i.d. replications of the system then yields i.i.d. samples $X_1, X_2, \ldots, X_n$. The sample average $\bar{X}_n = (1/n)\sum_{i=1}^{n} X_i$ is then an estimate of the true mean α.

To provide an estimate of the error in $\bar{X}_n$, we can also give a confidence interval for α, which is typically derived from a central limit theorem (CLT); e.g., see Section 1.9 of Serfling (1980). Specifically, a CLT for the sample average $\bar{X}_n$ roughly states that $\sqrt{n}(\bar{X}_n - \alpha)/\sigma \overset{D}{\approx} N(0,1)$ for large sample sizes n, where σ^2 is the variance of X, $N(0,1)$ denotes a normal random variable with mean 0 and variance 1, and $\overset{D}{\approx}$ means approximately equal in distribution. Thus, since $P(N(0,1) \leq 1.96) = 0.975$ and by the symmetry of the normal density function, the CLT implies

$$\begin{aligned} 0.95 &= P(-1.96 \leq N(0,1) \leq 1.96) \\ &\approx P\left(-1.96 \leq \frac{\sqrt{n}(\bar{X}_n - \alpha)}{\sigma} \leq 1.96\right) \\ &= P\left(\bar{X}_n - \frac{1.96\sigma}{\sqrt{n}} \leq \alpha \leq \bar{X}_n + \frac{1.96\sigma}{\sqrt{n}}\right), \end{aligned} \tag{1.1}$$

which gives $(\bar{X}_n \pm 1.96\sigma/\sqrt{n})$ as an approximate 95% confidence interval for α. However, the variance constant σ^2 in the CLT is typically unknown, so we replace it with an estimator, the sample variance $S_n^2 = (1/(n-1))\sum_{i=1}^{n}(X_i - \bar{X}_n)^2$. This then leads to an approximate 95% confidence interval for α as $(\bar{X}_n \pm 1.96 S_n/\sqrt{n})$. In fact, since S_n is a consistent estimator of σ (i.e., S_n converges in probability to σ as $n \to \infty$; e.g., see p. 69 of Serfling (1980)), we can show that the above approximate 95% confidence interval is asymptotically valid

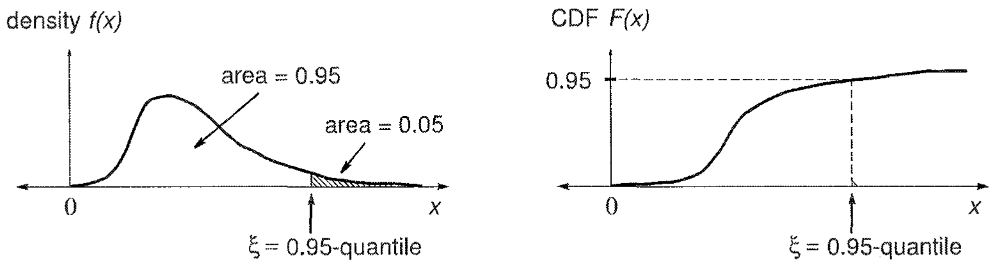

Figure 1.1 0.95-quantile of a distribution.

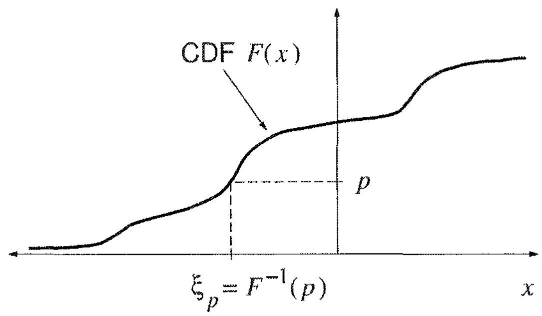

Figure 1.2 Inverting the CDF.

in the sense that $P\{\alpha \in (\bar{X}_n \pm 1.96 S_n/\sqrt{n})\} \to 0.95$ as $n \to \infty$.

In some situations, one may be interested in performance measures that are not simply means. One such measure is a *quantile*. For $0 < p < 1$, the pth quantile of a continuous random variable X is defined as the smallest constant ξ_p such that $P\{X \leq \xi_p\} = p$. Figure 1.1 gives an example of the 0.95-quantile of a distribution. In terms of the cumulative distribution function (CDF) F of X, we can express the pth quantile as $\xi_p = F^{-1}(p)$, where $F^{-1}(p)$ is the smallest value of x such that $F(x) \geq p$; see Figure 1.2. For example, the 0.5-quantile is the median. Quantiles arise in many practical contexts and are sometimes of more interest than means. For example, some internet service providers charge a user based on the 0.95 quantile of the user's traffic load in a billing cycle (Goldenberg et al. 2004). In project planning, a planner may want to determine a time t such that the project has a 95% chance of completing by t, so $t = \xi_{0.95}$ is the 0.95-quantile. In finance, where a quantile is known as a value-at-risk, an analyst may be interested in the 0.99-quantile $\xi_{0.99}$ of the loss of a portfolio over a certain time period (e.g., two weeks), so there is a 1% chance that the loss over this period will be greater than $\xi_{0.99}$. Value-at-Risk (VaR) is widely used in the financial industry as a measure of portfolio risk; e.g., see Duffie and Pan (1997). In a broader sense, VaR is also a preferred approach to compute market risk, one of the three components of the first pillar of Basel II, the second of the Basel Accords (recommendations on banking laws and regulations) issued by the Basel Committee on Banking Supervision. The Basel II Framework describes a more comprehensive measure and minimum standard for capital adequacy. Generally speaking, quantile estimation is of great interest to the financial industry, which motivates our exploration in how to estimate a quantile and measure the "accuracy" of quantile estimators by constructing confidence intervals.

Often in practice, the CDF F is unknown or cannot be computed explicitly, but we still may be able to collect samples from F. We thus seek a sampling-based estimator of ξ_p. One complication arises from the fact that a quantile is not a mean (nor a function of a mean) of a random variable, so we cannot estimate a quantile using a sample average. Instead, the following approach (e.g., see Section 2.3 of Serfling (1980)) can be applied. First collect i.i.d. samples $X_1, X_2, \ldots, X_n$ from distribution F, and use these to construct an estimator of F. One such estimator of F is the *empirical CDF* F_n, where $F_n(x)$ is the fraction of the n samples less than or equal to x. Then the fact that $\xi_p = F^{-1}(p)$ suggests constructing a quantile estimator as $\hat{\xi}_{p,n} = F_n^{-1}(p)$. We can alternatively compute the quantile estimator $\hat{\xi}_{p,n}$ in this case as follows. Sort the samples $X_1, X_2, \ldots, X_n$ into ascending order as $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$, where $X_{(i)}$ is the ith smallest of the samples. Then $\hat{\xi}_{p,n} = X_{(\lceil np \rceil)}$, where $\lceil \cdot \rceil$ is the round-up function. Figure 1.3 illustrates an example of this. When applying simulation to generate the i.i.d. samples of X used to construct F_n, we call the method *crude Monte Carlo*.

In addition to computing a point estimate for a quantile, it is important to also pro-

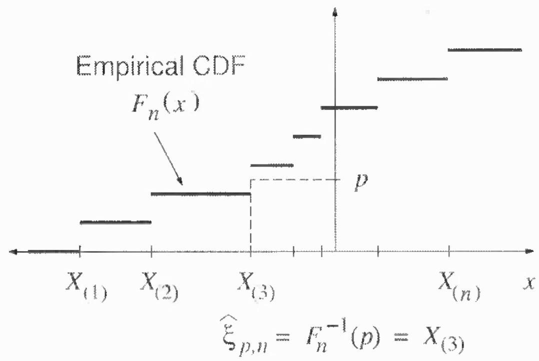

Figure 1.3 Inverting the empirical CDF.

vide a confidence interval for the quantile as a way of indicating the error in the estimate. A common approach to developing a confidence interval is to first show that the quantile estimator satisfies a CLT, and then replace the variance constant in the CLT with a consistent estimator of it to construct a confidence interval. For crude Monte Carlo, one can appeal to the CLT for quantile estimators formed from i.i.d. samples; e.g., see Section 2.3.3 of Serfling (1980). More specifically, let $\hat{\xi}_{p,n} = F_n^{-1}(p)$ be the estimator of the p-quantile ξ_p. A CLT for $\hat{\xi}_{p,n}$ roughly states that $\sqrt{n}(\hat{\xi}_{p,n} - \xi_p)/(p(1-p)/f(\xi_p)) \stackrel{D}{\approx} N(0,1)$ for large sample sizes n, where f is the density function of F. Using a similar derivation to (1.1), we then obtain $(\hat{\xi}_{p,n} \pm 1.96p(1-p)/(f(\xi_p)\sqrt{n}))$ as an approximate 95% confidence interval for ξ_p. Typically, $f(\xi_p)$ is unknown and must be estimated. Bloch and Gastwirth (1968), Bofinger (1975) and Babu (1986) provide consistent estimators $\hat{f}_n$ of $f(\xi_p)$ that are applicable for crude Monte Carlo, which we can then use to construct an approximate 95% confidence interval for ξ_p as $(\hat{\xi}_{p,n} \pm 1.96p(1-p)/(\hat{f}_n\sqrt{n}))$.

A problem with crude Monte Carlo is that the resulting confidence interval may be large, which is often the case when estimating extreme quantiles (i.e., when p is close to 0 or 1). This motivates applying variance-reduction techniques (VRTs) to try to obtain more efficient quantile estimators; see Chapter 4 of Glasserman (2004) for an overview of VRTs for estimating a mean. VRTs often increase efficiency by collecting additional data not ordinarily used or by generating samples in a different way. Even with today's high-powered computers, VRTs are still useful (and in certain settings essential) because complex simulation models, especially high-dimensional ones with many sources of randomness and complicated transition mechanisms, may take an extremely long time just to complete a single replication. Examples of VRTs include importance sampling, stratified sampling, antithetic variates, and control variates, and we briefly review these methods below.

In importance sampling (Glynn and Iglehart (1989)), which is typically used in rare-event simulations, the probabilistic dynamics of the system are changed to make the rare event of interest (e.g., system failures or extreme portfolio losses) occur more frequently. Unbiased estimates are recovered by multiplying the samples by a correction factor known as the likelihood ratio. When applied properly, importance sampling can lead to variance reductions of orders of magnitude; e.g., see Heidelberger (1995). However, if not appropriately used, importance sampling can actually increase variance (or even lead to infinite variance).

Stratified sampling (e.g., Sections 4.3 and 9.2.3 of Glasserman (2004)) constrains the proportion of samples of the output X into different strata of the sample space. The samples in each stratum are averaged, and the sample averages across strata are combined to form an overall estimator. This can lead to a variance reduction when the sampling proportions are chosen appropriately.

Antithetic variates (e.g., Section 11.3 of Law (2006)) generates sample outputs in pairs, where the outputs within a pair are negatively correlated. The basic intuition is that if one output is generated by "favorable" random fluctuations, then this is paired with another output with "unfavorable" fluctuations. Averaging the two negatively correlated outputs then smooths out the "good" and "bad" outputs, which leads to a reduction in variance.

Control variates (e.g., Section 11.4 of Law (2006)) reduces variance by taking advantage of the correlation between an auxiliary variable C, whose mean ν is known, and the output X, whose mean α we want to compute. We call C a *control variate*, and CV uses the known error $C - \nu$ in C to correct for the unknown error of X as an estimate of the unknown α. For example, in a stochastic activity network, we know the distributions of the times for each of the edges, so we would also likely know their means. Then we can determine the path through the network with the largest mean, and then set C to be the random length of that path.

There has been some previous work on applying VRTs to estimate a quantile. Hsu and Nelson (1990) and Hesterberg and Nelson (1998) develop quantile estimators using control variates. Avramidis and Wilson (1998) consider quantile estimation with a general class of correlation-induction techniques, which includes antithetic variates and Latin hypercube sampling (LHS). Jin et al. (2003) establish exponential convergence rates for quantile estimators, including those using LHS, and also develop a type of combined stratified-LHS quantile estimator. Glynn (1996) uses importance sampling for quantile estimation, and Glasserman et al. (2000b) combine importance sampling with stratified sampling to estimate value-at-risk. Variance reduction for quantile estimation typically entails applying VRTs to estimate the CDF F and then inverting the resulting CDF estimator $\tilde{F}_n$.

Most previous work on estimating quantiles using VRTs does not provide methods for constructing asymptotically valid confidence intervals. (An exception is Hsu and Nelson (1990), who generalize an interval estimation technique based on the binomial distribution for their control-variate estimator.) One approach to constructing a confidence interval is to first establish a CLT for the quantile estimator. Specifically, suppose that $\tilde{\xi}_{p,n} = \tilde{F}_n^{-1}(p)$ is the estimator of the p-quantile ξ_p formed by inverting $\tilde{F}_n$, the VRT estimator of the CDF F. Then a CLT for $\tilde{\xi}_{p,n}$ roughly states that $\sqrt{n}(\tilde{\xi}_{p,n} - \xi_p)/\kappa_p \overset{D}{\approx} N(0,1)$ for large n, for some constant κ_p. We can then unfold this to obtain an approximate 95% confidence interval for ξ_p to be $(\tilde{\xi}_{p,n} \pm 1.96\kappa_p/\sqrt{n})$. For this interval to be useful, we need to provide a consistent estimate of κ_p. It turns out that $\kappa_p = \psi_p/f(\xi_p)$, where ψ_p^2 is the variance constant in the CLT for $\tilde{F}_n(\xi_p)$ (i.e., ψ_p is defined in the CLT $\sqrt{n}(\tilde{F}_n(\xi_p) - F(\xi_p))/\psi_p \overset{D}{\approx} N(0,1)$)and $f(\xi_p)$ is the density function of the (unknown)

CDF F evaluated at the (unknown) quantile. Glynn (1996) notes that to construct confidence intervals, "the major challenge is finding a good way of estimating" $f(\xi_p)$, "either explicitly or implicitly," but he does not provide a method for doing this. Indeed, Glasserman et al. (2000b) state (p. 1357) that estimation of $f(\xi_p)$ "is difficult and beyond the scope of this paper." The consistency proofs of previously developed methods (e.g., Bloch and Gastwirth (1968), Bofinger (1975), and Babu (1986)) for estimating $f(\xi_p)$ (or $1/f(\xi_p)$) when using crude Monte Carlo do not generalize when using VRTs. (Viewed as a function of $0 < p < 1$, $1/f(\xi_p)$ is sometimes called the *sparsity function* (Tukey (1965)) or the *quantile-density function* (Parzen (1979)), and these two references discuss its usefulness, apart from quantile estimation, in analyzing data and distributions.)

In our dissertation we provide a way to consistently estimate $1/f(\xi_p)$ and ψ_p when using VRTs, and taking the product of these estimators yields a consistent estimator of κ_p. This enables us to construct an asymptotically valid confidence interval for the quantile when applying VRTs, which is the main contribution of our work. We establish our results within a general framework for VRTs specified by a set of assumptions on the resulting CDF estimator. We first prove the quantile estimator resulting from inverting the CDF estimator satisfies a weaker form of a so-called Bahadur (1966) representation established by Ghosh (1971), and we call this a Bahadur-Ghosh representation. Also of independent interest, this result shows that the quantile estimator can be approximated by a linear function of the CDF estimator evaluated at ξ_p, with a remainder term vanishing in probability as the sample size grows. We then apply the Bahadur-Ghosh representation to prove a CLT and to derive a consistent estimator for the asymptotic variance in the CLT. We show that different VRTs, including a combination of importance sampling and stratified sampling, antithetic variates, and control variates, fit in our framework, and we provide algorithms for constructing confidence intervals for quantiles estimated using these VRTs.

As an alternative approach, one could divide all the data into batches (also known as subsamples), and then produce a confidence interval by constructing a quantile estimate from each batch and computing the sample variance of the (i.i.d.) quantile estimates; e.g., see p. 491 of Glasserman (2004). However, a drawback of batching is that accurate quantile estimation often requires large sample sizes; e.g., see Avramidis and Wilson (1998). Thus, it is preferable to have methods that use all of the sampled data to construct a single quantile estimator, as we do.

The rest of the dissertation has the following organization. Chapter 2 reviews variance-reduction techniques for estimating a mean. In this chapter, a few financial examples are chosen to demonstrate a wide range of applications of Monte Carlo methods to estimate the "mean performance" in finance. Chapter 3 discusses quantile estimation and provides the background on the Bahadur-Ghosh representation for crude Monte Carlo. In Chapter 4 we establish a general framework for proving a Bahadur-Ghosh representation and for developing asymptotically valid confidence intervals for quantiles when applying a generic VRT. We then employ this framework in Chapters 5–7 to examine specific VRTs (combined importance sampling and stratified sampling, antithetic variates, and control variates). Chapter 8 presents experimental results. All proofs within a chapter are collected at the end of that chapter.

CHAPTER 2

REVIEW OF TECHNIQUES FOR ESTIMATING MEANS

Before presenting techniques for estimating quantiles in the later chapters, we first review methods for estimating means, which is the most common use of simulation. We discuss different Monte Carlo techniques for estimating the mean $\alpha = E[X]$ of a random variable X having CDF F. We start by reviewing estimating α using crude Monte Carlo, and then we discuss how to apply VRTs to estimate α. We also review some convergence concepts from probability.

Although the discussion here focuses on estimating means, the techniques presented also apply to estimating the CDF F of X. Recall that $F(x) = P\{X \leq x\}$ for each x, and we can write $F(x) = E[I(X \leq x)]$, where $I(A)$ is the indicator function of the event A, with $I(A) = 1$ if A occurs and $I(A) = 0$ otherwise. Thus, $F(x)$ is the mean of $I(X \leq x)$. As noted in Chapter 1, we estimate a quantile by first estimating the CDF and then inverting the CDF estimator. We will specialize the techniques developed here to estimate a CDF in Chapters 3 and 5–7. To simplify notation, we will use $\tilde{\alpha}_n$ to denote the estimator of α for all of the different VRTs rather than develop different notation for each method.

2.1 Crude Monte Carlo

To implement crude Monte Carlo to estimate $\alpha = E[X]$ where X is a random variable with CDF F, we draw n i.i.d. observations $X_1, X_2, \ldots, X_n$ from F, and then compute the sample average

$$\hat{\alpha}_n = \bar{X}_n \equiv \frac{1}{n}\sum_{i=1}^{n} X_i \tag{2.1}$$

as a point estimator of α. The estimator $\hat{\alpha}_n$ is *unbiased* since for all $n > 0$,

$$E[\hat{\alpha}_n] = E\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n}\sum_{i=1}^{n} E[X_i] = \alpha$$

because the X_i are i.i.d. with $E[X_i] = \alpha$. As the sample size n gets larger, $\hat{\alpha}_n$ gets closer to α, and this concept can be made precise through the (weak or strong) law of large numbers. Specifically, the *weak law of large numbers (WLLN)* states that if α is finite, then $P(|\hat{\alpha}_n - \alpha| > \varepsilon) \to 0$ as $n \to \infty$ for every $\varepsilon > 0$. We often write this as $\hat{\alpha}_n \xrightarrow{P} \alpha$ as $n \to \infty$, which is also known as *convergence in probability*, and we say that $\hat{\alpha}_n$ is a (weakly) *consistent* estimator of α. The *strong law of large numbers (SLLN)* states that if α is finite, then $P(\lim_{n\to\infty} \hat{\alpha}_n = \alpha) = 1$, which is also written as $\hat{\alpha}_n \to \alpha$ *almost surely (a.s.)*. For more details on the weak and strong laws, see Section 22 of Billingsley (1995).

Although the WLLN and SLLN guarantee that $\hat{\alpha}_n$ will be close to α when the sample size n is large, we would like to get a sense of how close they are for a large but fixed n. The central limit theorem (CLT) provides a way of doing this. Specifically, let $\sigma^2 = \text{Var}[X] = E[(X-\alpha)^2]$ denote the variance of X. The CLT states that if $0 < \sigma^2 < \infty$, then

$$P\left\{\frac{\sqrt{n}}{\sigma}(\hat{\alpha}_n - \alpha) \le x\right\} \to P\{N(0,1) \le x\}$$

as $n \to \infty$ for all x, where $N(a,b^2)$ denotes a normal random variable with mean a and variance b^2. We often denote this as $\sqrt{n}(\hat{\alpha}_n - \alpha)/\sigma \xrightarrow{L} N(0,1)$ as $n \to \infty$, where $\xrightarrow{L}$ denotes convergence in distribution; see Section 25 of Billingsley (1995) for details.

From the CLT, we can use the derivation in (1.1) to obtain an approximate 95% confidence interval for α as

$$(\hat{\alpha}_n \pm 1.96\sigma/\sqrt{n}). \tag{2.2}$$

(Of course, we could also construct an approximate confidence interval for any other confidence level $100(1-\delta)\%$ by replacing 1.96 with the critical value $z_\delta = \Phi^{-1}(1-\delta/2)$, where Φ is the CDF of a $N(0,1)$ random variable.) However, since σ is typically unknown, we need to estimate it for the above confidence interval to be useful. An estimator of σ^2 is the sample variance

$$\hat{\sigma}_n^2 = S_n^2 \equiv \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \hat{\alpha}_n)^2.$$

As noted in Chapter 1, $\hat{\sigma}_n^2 \xrightarrow{P} \sigma^2$ as $n \to \infty$ (e.g., see p. 69 of Serfling (1980)), so $\hat{\sigma}_n^2$ is a consistent estimator of σ^2. The continuous-mapping theorem (p. 334 of Billingsley (1995)) then ensures that $\hat{\sigma}_n \xrightarrow{P} \sigma$ as $n \to \infty$, so $\hat{\sigma}_n$ is a consistent estimator of σ. Hence, since $\sigma/\hat{\sigma}_n \xrightarrow{P} 1$ as $n \to \infty$ by the continuous-mapping theorem, Slutsky's theorem (p. 19 of Serfling (1980)) implies

$$\frac{\sqrt{n}}{\hat{\sigma}_n}(\hat{\alpha}_n - \alpha) = \frac{\sigma}{\hat{\sigma}_n}\frac{\sqrt{n}}{\sigma}(\hat{\alpha}_n - \nu) \xrightarrow{L} 1 \cdot N(0,1) = N(0,1)$$

as $n \to \infty$. As a consequence, for large sample sizes n,

$$\begin{aligned}
0.95 &= P(-1.96 \le N(0,1) \le 1.96) \\
&\approx P\left(-1.96 \le \frac{\sqrt{n}(\hat{\alpha}_n - \alpha)}{\hat{\sigma}_n} \le 1.96\right) \\
&= P\left(\bar{X}_n - \frac{1.96\hat{\sigma}_n}{\sqrt{n}} \le \alpha \le \bar{X}_n + \frac{1.96\hat{\sigma}_n}{\sqrt{n}}\right).
\end{aligned}$$

Thus, an approximate 95% confidence interval for α is

$$\left(\hat{\alpha}_n \pm 1.96\hat{\sigma}_n/\sqrt{n}\right). \tag{2.3}$$

2.2 Variance Reduction

From (2.2) and (2.3), we can use $1.96\sigma/\sqrt{n}$ or $1.96\hat{\sigma}_n/\sqrt{n}$ as a measure of the error in the estimator $\hat{\alpha}_n$ of α. Thus, for a given confidence level of 95%, there are two ways in which we can reduce the error: increase the number of samples n, or try to decrease σ. Although $\sigma = \sqrt{\mathrm{Var}[X]}$ is fixed and cannot be changed, suppose that there is another random variable Z with the same mean $\alpha = E[Z] = E[X]$ as X and with $\mathrm{Var}[Z] < \mathrm{Var}[X]$. Thus, rather than collecting samples of X, we could instead sample Z to estimate α and obtain a variance reduction. Specifically, we draw i.i.d. samples $Z_1, Z_2, \ldots, Z_n$ of Z, and form the sample average $\tilde{\alpha}_n = (1/n)\sum_{i=1}^n Z_i$. Let $\tau^2 = \mathrm{Var}[Z]$, and assume that $0 < \tau^2 < \infty$, so the CLT implies

$$\sqrt{n}(\tilde{\alpha}_n - \alpha)/\tau \xrightarrow{L} N(0,1) \tag{2.4}$$

as $n \to \infty$. Hence, using an analogous derivation to (1.1) yields another approximate 95% confidence interval for α as

$$(\tilde{\alpha}_n \pm 1.96\tau/\sqrt{n}). \tag{2.5}$$

We can replace τ^2 with the sample variance $\tilde{\tau}_n^2 = (1/(n-1))\sum_{i=1}^{n}(Z_i - \tilde{\alpha}_n)^2$. Then we get an approximate 95% confidence interval for α as

$$(\tilde{\alpha}_n \pm 1.96\tilde{\tau}_n/\sqrt{n}). \tag{2.6}$$

If $\mathrm{Var}[Z] < \mathrm{Var}[X]$, then we see that the estimator $\tilde{\alpha}_n$ has smaller error than $\hat{\alpha}_n$ for the same size n. Hence, the goal of many (but not all) variance-reduction techniques is to identify another random variable Z with the same mean as X but with smaller variance.

Another way to describe the benefit of applying VRTs is in terms of the number of samples needed to achieve a certain precision. Suppose we want a 95% confidence interval with half width of ε; i.e., $\hat{\alpha}_n \pm \varepsilon$ or $\tilde{\alpha}_n \pm \varepsilon$. Thus, for crude Monte Carlo, we require the sample size n to satisfy $\frac{1.96\sigma}{\sqrt{n}} = \varepsilon$, or $n = (1.96\sigma/\varepsilon)^2$. Similarly, when applying a VRT, we need a sample size $n = (1.96\tau/\varepsilon)^2$. Hence if $\tau < \sigma$, then the VRT estimator requires fewer samples than crude Monte Carlo to achieve the same precision.

Not all VRTs we consider can be applied where α is estimated via i.i.d. samples of a random variable Z. More generally, we may start with an estimator $\tilde{\alpha}_n$ of α, where $E[\tilde{\alpha}_n] = \alpha$ but $\tilde{\alpha}_n$ is not necessarily just a sample average. If the estimator $\tilde{\alpha}_n$ satisfies the CLT (2.4) for some constant $0 < \tau < \infty$ and $\tilde{\tau}_n$ is a consistent estimator of τ, then we obtain the approximate 95% confidence intervals in (2.5) and (2.6). In the next sections, we examine specific VRTs.

Since generating an estimator using a VRT often requires more computational effort than for crude Monte Carlo, a VRT should only be applied if the reduction in variance outweighs the additional computational effort. For our problems, the additional work needed to apply a VRT is often negligible, so this is not an issue. For more details on the more general setting, see Glynn and Whitt (1992).

2.2.1 Antithetic Variates

The basic idea of implementing antithetic variates (AV) (e.g., Section 11.3 of Law (2006)) to estimate $\alpha = E[X]$ is to generate samples from CDF F in pairs (X, X'), where X and X' are negatively correlated, and then average the samples within the pair as $Z = (X + X')/2$. Since X and X' have the same distribution, we have $E[Z] = \alpha$ and $\text{Var}[Z] = (1/4)(\text{Var}[X] + \text{Var}[X'] + 2\text{Cov}[X, X']) = (1/2)(\text{Var}[X] + \text{Cov}[X, X'])$, where $\text{Cov}[A, B]$ denotes the covariance of random variables A and B defined by $\text{Cov}[A, B] = E[(A - E[A])(B - E[B])] = E[AB] - E[A]E[B]$. If $\text{Cov}[X, X'] < 0$, then $Z = (X + X')/2$ has smaller variance than the average of two independent copies of X.

There are various ways in which we can generate negatively correlated X and X' with the same marginal distribution F. For example, suppose that the output X can be expressed as $X = h(U_1, \ldots, U_d)$ for some function h, where $U_1, \ldots, U_d$ are i.i.d. uniform random variables on the unit interval. Then $X' = h(1 - U_1, \ldots, 1 - U_d)$ has the same distribution as X since $1 - U_i$ is also uniform on $[0, 1]$. If h is monotonic in each of its arguments, then X and X' are negatively correlated; e.g., see Section 8.1 of Ross (1997).

Suppose some implementation of antithetic sampling produces n pairs of observations $(X_1, X_1'), (X_2, X_2'), \ldots, (X_n, X_n')$ that have the following properties:

1. the pairs are i.i.d.;

2. in each pair i, X_i and X_i' each have marginal CDF F, and X_i and X_i' are negatively correlated.

The AV estimator $\tilde{\alpha}_n$ is the average of all $2n$ random variables:

$$\begin{aligned}\tilde{\alpha}_n &= \frac{1}{2n}\sum_{i=1}^{n}(X_i + X_i') \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i + X_i'}{2}\right) \\ &= \frac{1}{2}(\hat{\alpha}_n + \hat{\alpha}_n'),\end{aligned}$$

where $\hat{\alpha}_n$ is defined in (2.1) and $\hat{\alpha}_n' = (1/n)\sum_{i=1}^{n} X_i'$. Thus, we see that $\tilde{\alpha}_n$ is the sample average of i.i.d. samples of $Z = (X + X')/2$. The estimator $\tilde{\alpha}_n$ satisfies $E[\tilde{\alpha}_n] = \alpha$ since $E[X_i] = E[X_i'] = \alpha$ because X_i and X_i' each have marginal CDF F. Also, we have the CLT in (2.4) holds when $\tau^2 = \mathrm{Var}[Z]$ satisfies $0 < \tau^2 < \infty$. Letting $Z_i = (X_i + X_i')/2$, we then obtain an estimator of τ^2 as

$$\tilde{\tau}_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Z_i - \tilde{\alpha}_n)^2,$$

which we can use in (2.6) to obtain an approximate 95% confidence interval for α.

2.2.2 Control Variates

The key idea of control variates (CV) is to leverage knowledge of known quantities to reduce the error in an estimate of an unknown quantity; e.g., see Section 11.4 of Law (2006). Let (X, C) be a pair of correlated random variables, where we are interested in estimating the mean α of X, and suppose that we know the mean ν of C. We call C a *control variate*. For example, in a queueing system, we might take C as the average of the first 5 customers' service times, and we would typically know the mean ν of the service-time distribution. Define $Z = X - (C - \nu)$, and note that $E[Z] = E[X] - (E[C] - \nu) = \alpha$, so we can collect samples of Z to estimate α.

The intuition behind control variates is as follows. Suppose that X and C are positively correlated. Then $X > \alpha$ and $C > \nu$ tend to occur together. Hence, if $C > \nu$, then $Z = X - (C - \nu) < X$, so Z corrects for the fact that X is likely larger than its mean α.

Similarly, $X < \alpha$ and $C < \nu$ tend to occur together when X and C are positively correlated. Hence, if $C < \nu$, then $Z = X - (C - \nu) > X$, so Z corrects for the fact that X is likely smaller than its mean α. If the correlation between X and C is strong enough, then we can obtain a variance reduction from sampling Z rather than X.

More generally, we can define $Z = X - \beta(C - \nu)$ for any constant β, so $E[Z] = \alpha$. To implement CV, we generate n i.i.d. samples (X_i, C_i), $i = 1, 2, \ldots, n$, of (X, C), and let $Z_i = X_i - \beta(C_i - \nu)$. We then form an estimator of α as

$$\tilde{\alpha}_{n,\beta} = \frac{1}{n}\sum_{i=1}^{n} Z_i = \frac{1}{n}\sum_{i=1}^{n} [X_i - \beta(C_i - \nu)]. \tag{2.7}$$

The variance of Z is

$$\tau^2 = \mathrm{Var}[X - \beta(C - \nu)] = \mathrm{Var}[X] + \beta^2 \mathrm{Var}[C] - 2\beta \mathrm{Cov}[X, C], \tag{2.8}$$

and $\mathrm{Var}[Z] < \mathrm{Var}[X]$ when $2\beta\mathrm{Cov}(X,C) > \beta^2\mathrm{Var}(C)$. In other words, CV yields a variance reduction when X and C have sufficiently strong correlation.

The variance of Z (and $\tilde{\alpha}_{n,\beta}$) depends on the choice of the constant β. Since τ^2 in (2.8) is a quadratic function of β, it takes the minimum value $(1-\rho^2)\mathrm{Var}(X)$ at $\beta_* = \mathrm{Cov}(X,C)/\mathrm{Var}(C)$, where $\rho = \mathrm{Cov}(X,C)/\sqrt{\mathrm{Var}(X)\mathrm{Var}(C)}$ is the correlation of X and C. However, $\mathrm{Cov}(X,C)$ and $\mathrm{Var}[C]$ are typically unknown and must be estimated, and we estimate β_* via

$$\hat{\beta}_n = \frac{(1/n)\sum_{i=1}^{n}(X_i - \bar{X}_n)(C_i - \bar{C}_n)}{(1/n)\sum_{i=1}^{n}(C_i - \bar{C}_n)^2}, \tag{2.9}$$

where $\bar{C}_n = (1/n)\sum_{i=1}^{n} C_i$. We then substitute $\hat{\beta}_n$ for β in (2.7) to obtain the CV estimator of α as

$$\tilde{\alpha}_n = \frac{1}{n}\sum_{i=1}^{n}\left[X_i - \hat{\beta}_n(C_i - \nu)\right]. \tag{2.10}$$

One complication of the estimator $\tilde{\alpha}_n$ is that it is no longer the average of i.i.d. observations since each summand in (2.10) includes $\hat{\beta}_n$, which induces dependence among the summands. However, it can be shown (e.g., pp. 195–196 of Glasserman (2004)) that $\tilde{\alpha}_n$ still satisfies the CLT in (2.4) with τ^2 defined in (2.8), and

$$\tilde{\tau}_n^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2 - \frac{2\hat{\beta}_n}{n}\sum_{i=1}^{n}(X_i - \bar{X}_n)(C_i - \nu) + \frac{\hat{\beta}_n^2}{n}\sum_{i=1}^{n}(C_i - \nu)^2$$

is a consistent estimator of τ^2. Thus, (2.6) provides an approximate 95% confidence interval for α.

2.2.3 Stratified Sampling

Stratified sampling (SS) (e.g., see Sections 4.3 and 9.2.3 of Glasserman (2004)) constrains the proportion of samples of the output X into different strata of the sample space. Samples from each stratum are taken to estimate the mean in that stratum, and the resulting estimates from each stratum are combined to obtain an overall estimator.

Let (X,Y) be a pair of dependent random variables, where we are interested in estimating the mean α of X. We will use Y as a *stratification variable*, which is sometimes an auxiliary quantity that is generated in the process of generating output X. Partition the support of Y into $k < \infty$ strata $S_1, S_2, \ldots, S_k$; i.e., $S_i \cap S_j = \emptyset$ for $i \neq j$ and $P(Y \in \cup_{i=1}^{k} S_i) = 1$. For example, the S_i may be disjoint intervals, and Sections 4.3 and 9.2.3 of Glasserman (2004) describes other possible choices for the strata. Define $\lambda_i = P(Y \in S_i)$, which we assume is known. Then $\alpha = E[X]$ can be written as

$$\alpha = \sum_{i=1}^{k} \lambda_i E[X | Y \in S_i]. \tag{2.11}$$

This motivates estimating α by separately estimating each $E[X \mid Y \in S_i]$ in (2.11) using a fraction of our total sampling budget n and combining the resulting estimators with the weights λ_i. To do this, we first specify positive constants γ_i, $i = 1, \ldots, k$, such that $\sum_{i=1}^{k} = 1$. Then let $n_i = \gamma_i n$ be the number of samples used to estimate $E[X \mid Y \in S_i]$. (We discuss later possible choices for γ_i.) To implement stratified sampling, for the ith stratum, we generate n_i samples Y_{ij}, $j = 1, \ldots, n_i$, from S_i. Then for each $j = 1, \ldots, n_i$, we generate X_{ij} given $Y = Y_{ij}$. This then results in n_i i.i.d. pairs (X_{ij}, Y_{ij}), $i = 1, \ldots, n_i$, from stratum S_i. Then, the SS estimator of α is

$$\tilde{\alpha}_n = \sum_{i=1}^{k} \frac{\lambda_i}{n_i} \sum_{j=1}^{n_i} X_{ij}. \tag{2.12}$$

Because X_{ij} has the distribution of X given $Y \in S_i$, the estimator is unbiased since

$$\begin{aligned} E[\tilde{\alpha}_n] &= \sum_{i=1}^{k} \lambda_i \frac{1}{n_i} \sum_{j=1}^{n_i} E\left[X_{ij}\right] \\ &= \sum_{i=1}^{k} \lambda_i \frac{1}{n_i} \sum_{j=1}^{n_i} E\left[X \mid Y \in S_i\right] \\ &= \sum_{i=1}^{k} \lambda_i E\left[X \mid Y \in S_i\right] = \alpha \end{aligned}$$

by (2.11). Let σ_i^2 be the variance of X_{ij} in the ith stratum, and we assume each $\sigma_i^2 < \infty$. Then the variance of $\tilde{\alpha}_n$ is

$$\mathrm{Var}(\tilde{\alpha}_n) = \sum_{i=1}^{k} \lambda_i^2 \mathrm{Var}\left[\frac{1}{n_i} \sum_{j=1}^{n_i} X_{ij}\right] = \sum_{i=1}^{k} \lambda_i^2 \frac{\sigma_i^2}{n_i}. \tag{2.13}$$

It can shown (e.g., pp. 215-216 of Glasserman (2004)) that the SS estimator $\tilde{\alpha}_n$ satisfies the CLT in (2.4), with $\tau^2 = \sum_{i=1}^{k} \lambda_i^2 \sigma_i^2 / \gamma_i$. We can then consistently estimate τ^2 using

$$\tilde{\tau}_n^2 = \sum_{i=1}^{k} \frac{\lambda_i^2}{\gamma_i} \left[\frac{1}{n_i - 1} \sum_{j=1}^{n_i} \left(X_{ij} - \frac{1}{n_i} \sum_{j'=1}^{n_i} X_{ij'}\right)^2\right],$$

which we can then put in (2.6) to obtain an approximate 95% confidence interval for α.

Whether or not the SS estimator $\tilde{\alpha}_n$ achieves a variance reduction depends on the stratification weights γ_i. With *proportional allocation*, we set $\gamma_i = \lambda_i$, so that the sampling variability across strata is eliminated without affecting sampling variability within strata, and this is guaranteed to result in the SS estimator $\tilde{\alpha}_n$ in (2.12) having no greater variance than the crude Monte Carlo estimator $\hat{\alpha}_n$ in (2.1); e.g., see p. 217 of Glasserman (2004) Other allocation rules can also be adopted that are at least as effective as proportional allocation. The choice of γ_i that minimizes the variance in (2.13) is given by $\gamma_i = \lambda_i \sigma_i / (\sum_j \lambda_j \sigma_j)$, i.e., proportional to the product of the stratum probability and the stratum standard deviation.

2.2.4 Importance Sampling

Importance sampling (IS) (Glynn and Iglehart 1989) is a VRT that is often used in rare-event simulations, such as estimating the mean time to failure of a highly reliable system or estimating the buffer-overflow probability in a queueing system with a large but finite buffer (Heidelberger 1995). The basic idea of IS is to change the probability distributions governing the stochastic system under study to cause the rare event of interest (e.g., system failures or buffer overflows) to occur more frequently. Unbiased estimates are recovered by multiplying the samples by a correction factor known as the likelihood ratio.

Suppose X has CDF F with probability density function f, and let E be the expectation operator under F. We want to estimate $\alpha = E[X] = \int x f(x)\,dx$. Let F_* be another CDF with density function f_* having the property that for every x, $f(x) > 0$ implies $f_*(x) > 0$. Let E_* denote the expectation operator under F_*, and define the *likelihood ratio* $L(x) = f(x)/f_*(x)$. A *change of measure* can be carried out by writing the expectation value α of X as

$$\begin{aligned}\alpha = E[X] &= \int x f(x)\,dx = \int x \frac{f(x)}{f_*(x)} f_*(x)\,dx \\ &= \int x L(x) f_*(x)\,dx = E_*[X L] \qquad (2.14)\end{aligned}$$

with $L = L(X)$.

This then suggests estimating α by averaging outputs XL generated using F_* rather than sampling output X from the original CDF F. Specifically, let $X_1, \ldots, X_n$ be i.i.d. samples generated using F_*, and let $L_i = f(X_i)/f_*(X_i)$ be the likelihood ratio for the ith sample. Then the IS estimator of α is

$$\tilde{\alpha}_n = \frac{1}{n}\sum_{i=1}^{n} X_i L_i,$$

which is unbiased by (2.14), i.e., $E_*[\tilde{\alpha}_n] = \alpha$. Thus, the variance of XL under F_* is

$$\tau^2 = \mathrm{Var}_*(XL) = E_*\left[X^2L^2\right] - \alpha^2, \tag{2.15}$$

where Var_* is the variance under IS distribution F_*. Assuming that $0 < \tau^2 < \infty$, the IS estimator $\tilde{\alpha}_n$ satisfies the CLT in (2.4). We can estimate the variance τ^2 via

$$\tilde{\tau}_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i^2L_i^2 - \tilde{\alpha}_n)^2,$$

which we can then use in (2.6) to obtain an approximate 95% confidence interval for α.

The key to applying importance sampling is choosing F_* appropriately so that τ^2 in (2.15) is smaller than $\mathrm{Var}[X]$. Since $E_*[\tilde{\alpha}_n] = \alpha$, IS achieves a variance reduction when $E_*[X^2L^2] < E[X^2]$.

Suppose $f(x) = 0$ for all $x < 0$, so X is a nonnegative random variable and $\alpha > 0$. Define f_* such that

$$f_*(x) = \frac{xf(x)}{\alpha} \tag{2.16}$$

for all $x \geq 0$, and $f_*(x) = 0$ for $x < 0$. Then $f_*(x) \geq 0$ for all $x \geq 0$ since $\alpha > 0$ and $f(x) \geq 0$ because f is a density function. Also,

$$\int f_*(x) = \int \frac{xf(x)}{\alpha}dx = \frac{\alpha}{\alpha} = 1.$$

Thus, f_* is a density function. If we use this f_* to implement IS, each sample output is

$$XL = X\frac{f(X)}{f_*(X)} = X\frac{f(X)}{Xf(X)/\alpha} = \alpha,$$

so every sample output is exactly α. Therefore, the IS estimator of α has zero variance, and we require only one sample to estimate α without any error! Unfortunately, the density f_* in (2.16) is not implementable because it depends on α, which we do not know since we are trying to estimate it. However, f_* in (2.16) provides us with some insight into how to choose a good change of measure. Specifically, note that the optimal IS density $f_*(x)$ in (2.16) is large when $xf(x)$ is large, so we should try to choose a change of measure that gives more weight to values of x for which $xf(x)$ is large.

One way of obtaining an IS distribution F_* is to apply *exponential tilting or twisting* (Glynn and Iglehart 1989). To develop this approach, we first define the moment generating function (MGF) $m(\theta)$ of the random variable X under the original distribution F with density f as

$$m(\theta) = E[e^{\theta X}] = \int e^{\theta x} f(x)\, dx.$$

Then for any θ for which $m(\theta) < \infty$, we define the exponentially tilted density

$$f_*(x) = e^{\theta x} f(x)/m(\theta). \tag{2.17}$$

The key is choosing θ to obtain a variance reduction. This choice for the change of measure is especially effective for estimating buffer-overflow probabilities in a queue with a large but finite buffer; e.g., see Heidelberger (1995).

2.3 Applications

We now provide examples illustrating how the estimation techniques just described can be applied. Our examples, which are all from finance, examine how to obtain a fair price of a financial product known as a derivative. A derivative is a contract that depends on the price of one or more underlying assets, such as stocks. One example of a derivative is an *option*, which grants the holder the right rather than the obligation to buy (*call* option) or sell (*put* option) an underlying asset at a fixed price K by a fixed time T in the future.

Options have different types. For example, one of the simplest forms is called a European call option when the underlying asset is a stock paying no dividends and this option can only be exercised at (and not before) time T. Suppose the stock price is S_T at time $T = m\Delta T$ (the expiry time is m days with ΔT as one day) and the predetermined price in the contract is K (also known as *strike price*). At time T, if the price of the underlying asset is S_T and $S_T > K$, then the holder of the contract will exercise the option to purchase one share of the underlying asset at price K and then immediately sell it at the current market price S_T to gain a profit $S_T - K$; if $S_T < K$, the option will not be exercised and there is zero payoff. Then the payoff to the European call option holder at time T is

$$(S_T - K)^+ = \max\{0, S_T - K\}.$$

We need to make some assumptions about some market participants, such as large investment banks, to evaluate the value of options.

1. There are no transaction costs.

2. All trading profits are subject to the same tax rate.

3. Borrowing and lending are possible at the risk-free interest rate r.

In a *risk-neutral* world, investors require no compensation for risk. Under the *risk-neutral measure* the expected return on all securities is the risk-free interest rate, which means the present value of the option is its expected payoff in a risk-neutral world discounted at the risk-free rate. This is an important principle in option pricing known as *risk-neutral valuation*, which states the price we obtain is correct not only in a risk-neutral world but also in the real world as well; see Hull (2003) for a more general discription of risk-neutral valuation. Thus, the expected present value of the payoff to this European call option is $\alpha = E\left[e^{-rT}(S_T - K)^+\right]$, where E is the expectation under the risk-neutral measure, and e^{-rT} is a discount factor with r a continuously compounded risk-free rate. Thus, α is the fair price of the option.

We assume the price of the underlying stock follows a geometric Brownian motion, namely a continuous-time stochastic process in which the logarithm of the stock price follows a Brownian motion; see Section 10.3.2 of Ross (2007) for a basic description of geometric Brownian motion and an introduction to continuous-time stochastic processes and stochastic differential equations (SDE). The well-known Black-Scholes model proposed by Black and Scholes (1973) describes the evolution of the stock price through the SDE

$$\frac{dS_t}{S_t} = rdt + \sigma dW(t), \tag{2.18}$$

where σ is the volatility of the stock price and W is a standard Brownian motion. The Brownian motion W has the property that the change ΔW during a period of length ΔT satisfies $\Delta W = \varepsilon\sqrt{\Delta T}$, where ε is a random variable having standard normal distribution $N(0,1)$. This indicates ΔW itself has a normal distribution with $E[\Delta W] = 0$ and $\mathrm{Var}(\Delta W) = \Delta T$. Suppose S_{t_i} is the stock price at $t_i = i\Delta T$ for $i = 0,1,2,\ldots$. For this European call option in a multi-period time interval $T = m\Delta T$, solving (2.18) leads to

$$S_{t_i} = S_{t_{i-1}} e^{\mu+\psi Z_i} = S_0 \prod_{j=1}^{i} e^{\mu+\psi Z_j}, \tag{2.19}$$

where $Z_1, Z_2, \ldots, Z_m$ are i.i.d. standard normals $N(0,1)$, $\mu = (r - \sigma^2/2)\Delta T$ is the risk-free rate of return under the risk-neutral measure, S_0 is the present price of the stock, and $\psi^2 = \sigma^2\Delta T$. From (2.19), a closed-form solution for α, the expected discounted payoff of the European call option, can be computed (Black and Scholes 1973), which indicates Monte Carlo methods are not needed for this case.

Valuing more complex options, however, may require various Monte Carlo methods to be applied. Next, we will use different types of Monte Carlo techniques to evaluate complex options.

2.3.1 Using Crude Monte Carlo to Price an Asian Option

An Asian option is an average value option different from the usual European option and American option. The payoff of an Asian option depends on the average price level of the underlying asset over the life of the contract. For an Asian call option, the payoff at time T is $(\bar{S}-K)^+$ with

$$\bar{S} = \frac{1}{m}\sum_{i=1}^{m} S_{t_i}. \tag{2.20}$$

This means the option is exercised at T if and only if $\bar{S} > K$, and the payoff is the amount by which $\bar{S}$ exceeds K. To evaluate the expected discounted payoff

$$\alpha = E\left[e^{-rT}(\bar{S}-K)^+\right], \tag{2.21}$$

we first need to generate samples of

$$\bar{S} = \frac{1}{m}\sum_{i=1}^{m} S_{t_i} \tag{2.22}$$

by simulating the paths $S_{t_1}, S_{t_2}, \ldots, S_{t_m}$ based on (2.19). Then for the ith simulated path, the discounted payoff of the Asian call option is $X_i = e^{-rT}(\bar{S}_i - K)^+$, where $\bar{S}_i$ is the average price level defined in (2.20) on the ith path. After simulating n such paths, calculate the sample average of the X_i as $\hat{\alpha}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$, which is a consistent estimator of the fair price of this call option. Computing the sample variance of the X_i then leads to a confidence interval for the fair price α. Section 1.1 of Glasserman (2004) provides more details on pricing an Asian call option with crude Monte Carlo.

2.3.2 Using CV to Price an Asian Option

Kemna and Vorst (1990) suggest an effective approach to price Asian options with CV. To evaluate α defined in (2.21), we can define a CV C to be the discounted payoff of a geometric average option:

$$C = e^{-rT}(\bar{S}_G - K)^+,$$

where $\bar{S}_G = \left(\prod_{i=1}^m S_{t_i}\right)^{1/m}$. We can analytically compute $v = E[C]$ in closed-form by using the fact that

$$\bar{S}_G = S_0 \exp\left(\frac{\mu}{m}\sum_{i=1}^m t_i + \frac{\sigma}{m}\sum_{i=1}^m W(t_i)\right);$$

see an explanation of pricing a geometric average option in Section 3.2.2 in Glasserman (2004). Then, we can establish a CV estimator of α as

$$\tilde{\alpha}_n = \frac{1}{n}\sum_{i=1}^n \left[X_i - \hat{\beta}_n(C_i - v)\right],$$

where $\hat{\beta}_n$ is computed according to (2.9) and C_i is the discounted payoff of the geometric average option on the ith replication. Glasserman (2004) makes comparisons to indicate applying a CV C can result in an approximate fifty-fold speed-up in simulation due to a dramatic variance reduction.

2.3.3 Using AV to Price a Knock-out Option

A knock-out option is an option with a specified barrier and its payoff ceases to exist if the underyling asset price crosses the barrier before the expiry T. Suppose a prespecified lower barrier price is $A < \infty$, and the option is "knocked out" if the price of the underlying asset goes below the barrier A at any time t_i, $i = 1, 2, \ldots, m$. (We could also specify an upper barrier, but we do not to simplify the discussion.) Thus, the payoff at T is defined as

$$I(S_{t_i} > A, i \leq m)(S_T - K)^+.$$

Using (2.19), the discounted payoff of a knock-out option is

$$\begin{aligned}
&e^{-rT}I(S_{t_i} > A, i \leq m)(S_T - K)^+ \\
&= e^{-rT}I(S_0\prod_{j=1}^{i} e^{\mu+\psi Z_j} > A, i \leq m)(S_0\prod_{j=1}^{m} e^{\mu+\psi Z_j} - K)^+ \\
&\equiv h(Z_1, Z_2, \ldots, Z_m). \qquad (2.23)
\end{aligned}$$

Now we want to compute the fair price $\alpha = E\left[h(Z_1, Z_2, \ldots, Z_m)\right]$ of this knock-out option. Since Z_j has a standard normal distribution $N(0,1)$, we know $-Z_j$ also has a standard normal distribution $N(0,1)$ with negative correlation to Z_j due to its symmetry with respect to the origin. Then, it is natural to build an AV pair $(h(Z_1, Z_2, \ldots, Z_m), h(-Z_1, -Z_2, \ldots, -Z_m))$ with

$$h(Z_1, Z_2, \ldots, Z_m) \stackrel{D}{=} h(-Z_1, -Z_2, \ldots, -Z_m),$$

where $\stackrel{D}{=}$ denotes equal in distribution. By (2.23), we see that h is mononotonic in each Z_j because

1. ψ is positive, so for each $k > j$, S_{t_k} defined in (2.19) increases as Z_j increases;

2. indicator functions are monotonically increasing.

Therefore, the AV estimator of the expected discounted payoff of this knock-out option is $\bar{Y}_n = \frac{1}{n}\sum_{k=1}^{n} Y_k$, where

$$Y_k = \frac{1}{2}\left[h(Z_{k,1}, Z_{k,2}, \ldots, Z_{k,m}) + h(-Z_{k,1}, -Z_{k,2}, \ldots, -Z_{k,m})\right],$$

and $h(Z_{k,1}, Z_{k,2}, \ldots, Z_{k,m})$ is defined by (2.23) with $Z_{k,j}$ being the jth independent sample of a standard normal in the kth replication.

2.3.4 Using IS to Price a Knock-in Option

A knock-in option is similar to a knock-out option as an option with barriers, but a knock-in option requires that the underlying asset price hits a barrier before the expiration T; see Boyle et al. (1997) for a general desciption of knock-in options. Let H be a lower barrier, and let K be the strike price. The knock-in option pays $S_T - K$ at time T if $S_T > K$ and $S_{t_i} < H$ for some $i \leq m$. In some contracts, the lower barrier is specified to be much smaller than the initial price and the specified strike price is much larger than H. In this case, it is a rare event for the underlying asset price to drop below the lower barrier and then move up to exceed the strike price to generate a positive payoff at time T. Therefore, most paths of a crude Monte Carlo simulation will result in a zero payoff and crude Monte Carlo is not efficient. However, IS can potentially make knock-in less rare.

Set $H = S_0 e^{-b} < \infty$ as the lower barrier, where a constant $b > 0$ is given. Suppose the strike price is $K = S_0 e^c$ where constant $c > 0$. We can write (2.19) as

$$S_{t_i} = S_0 e^{U_i},$$

where $U_i = \sum_{j=1}^{i} X_j$ with X_j i.i.d. normal having mean μ and variance ψ^2. Let $\bar{\tau} = \inf(i : S_{t_i} < H)$, which is the first time to drop below H, or equivalently $U_i < -b$ for the first time. Let $\tau = \min(\bar{\tau}, m)$. Then the probability of a positive payoff is $P(\tau < m, U_m > c)$, which is close to zero if b and c are both large.

Let $\kappa(\theta)$ be the cumulant generating function of X_j, where $\kappa(\theta) = \ln m(\theta)$ and $m(\theta) = E\left[e^{\theta X_j}\right]$ is the moment generating function of X_j. For the exponentially tilted distribution defined in (2.17) with tilting parameter θ, it turns out its mean is $\kappa'(\theta)$, the derivative of $\kappa(\theta)$; e.g., see p.261 of Glasserman (2004). When the tilting parameter $\theta < 0$ (resp., $\theta > 0$), then the mean of the exponentially tilted distribution is smaller (resp., larger) than the original mean μ of X_j. We will design an IS to apply a change of measure in the following manner: use some $\theta_1 < 0$ to exponentially twist the distribution of X_j

until the barrier H is hit and then twist the sequence $\{X_{\tau+1}, X_{\tau+2}, \ldots\}$ by some $\theta_2 > 0$ to subsequently push the price towards K. This means U_i is given a drift of

$$\mu_1 = \kappa'(\theta_1) \tag{2.24}$$

until $i = \tau$ and then the drift is

$$\mu_2 = \kappa'(\theta_2), \tag{2.25}$$

where $\mu_1 < 0$ and $\mu_2 > 0$ to increase the chances that the option payoff is positive. By a change of measure, we have

$$P(\tau < m, U_m > c) = E_* \left[LI(\tau < m, U_m > c) \right],$$

where it can be shown that the likelihood ratio

$$\begin{aligned} L &= \prod_{j=1}^{\tau} \frac{f(X_j)}{\exp(\theta_1 X_j - \kappa(\theta_1)) f(X_j)} \prod_{j=\tau+1}^{m} \frac{f(X_j)}{\exp(\theta_2 X_j - \kappa(\theta_2)) f(X_j)} \\ &= \exp(-\theta_1 U_\tau + \kappa(\theta_1)\tau - \theta_2(U_m - U_\tau) + \kappa(\theta_2)(m - \tau)) \end{aligned}$$

with $f(X)$ as a density function of $N(\mu, \psi^2)$ and $\kappa(\theta) = (\mu\theta + \frac{\psi^2}{2}\theta^2)$. This change of measure process is done by changing the mean μ of the original measure to μ_1 and μ_2 in the new measure as described above.

Now we explain how to choose μ_1 and μ_2. Most of the variability in L comes from τ, the barrier-crossing time. Thus, we will try to choose $\theta_1 < 0$ and $\theta_2 > 0$ so that τ cancels out in L, thereby lowering the variability of L. For large b and c, the typical way a positive payoff occurs is the underlying asset price drops down from the initial price to barely hit the lower barrier and then move up to barely hit the strike price at the mth step. This implies $P(\tau < m; U_m > c) \approx P(U_\tau \approx -b, U_m \approx c)$. If we choose μ_1 and μ_2 so that $\kappa(\theta_1) = \kappa(\theta_2)$, then the likelihood ratio L reduces to $L = \exp(-(\theta_1 - \theta_2)U_\tau - \theta_2 U_m + m\kappa(\theta_2))$, which depends on τ only through $U_\tau \approx -b$ and thus eliminates the variability resulting from τ. Since $\kappa(\theta_1) = \kappa(\theta_2)$, it can be shown that $\mu_1 = -\mu_2 \equiv -\mu_*$ due to (2.24) and (2.25). To select θ_1 and θ_2, we consider the trajectory of U_i with drift μ_1 from the initial point until $-b$ is hit for the first time, and then the drift is μ_2 thereafter. Then, it can be shown that we need to select μ_* to satisfy

$$\frac{b}{\mu_*} + \frac{b+c}{\mu_*} = m,$$

which results in $\mu_* = (2b+c)/m$.

Boyle et al. (1997) present empirical results from simulations of a knock-in option with various parameters, and they obtain variance reductions of up to a factor of 1124. Glasserman (2004) obtains a similar result, and also notes that the variance ratio depends primarily on the rarity of the payoff and not otherwise on the maturity.

CHAPTER 3

QUANTILE ESTIMATION FOR CRUDE MONTE CARLO

We now turn our discussion to estimating a quantile. Let X be a real-valued random variable with CDF F. For a real-valued function G, define $G^{-1}(a) = \inf\{x : G(x) \geq a\}$. For a fixed $0 < p < 1$, we want to compute the pth quantile $\xi_p = F^{-1}(p)$. Suppose F is differentiable at ξ_p and $f(\xi_p) > 0$, where $f(x) = dF(x)/dx$.

We will estimate ξ_p using simulation. Crude Monte Carlo estimation of ξ_p entails first generating i.i.d. samples $X_1, X_2, \ldots, X_n$ from distribution F. Then we compute the *empirical distribution function* F_n as

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I(X_i \leq x) \tag{3.1}$$

as an estimator of $F(x)$, where $I(A)$ is the indicator function of a set A that assumes value 1 on A and 0 on A^c. We then compute the p-quantile estimator $\hat{\xi}_{p,n} = F_n^{-1}(p)$. An alternative way of computing $\hat{\xi}_{p,n}$ is in terms of order statistics. Sort the samples $X_1, X_2, \ldots, X_n$ into ascending order as $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$, where $X_{(i)}$ is the ith smallest of the samples. Then $\hat{\xi}_{p,n} = X_{(\lceil np \rceil)}$, where $\lceil \cdot \rceil$ is the round-up function.

Roughly speaking, we have that $F_n(x) \approx F(x)$ for all x for large sample sizes n, so $\hat{\xi}_{p,n} \approx \xi_p$. Thus, since $p = F(\xi_p)$, a Taylor approximation gives

$$p \approx F(\hat{\xi}_{p,n}) \approx F(\xi_p) + f(\xi_p)(\hat{\xi}_{p,n} - \xi_p) \approx F_n(\xi_p) + f(\xi_p)(\hat{\xi}_{p,n} - \xi_p)$$

since $F(\xi_p) \approx F_n(\xi_p)$. Hence, $\hat{\xi}_{p,n} \approx \xi_p - (F_n(\xi_p) - p)/f(\xi_p)$.

Bahadur (1966) makes rigorous the above heuristic argument. Specifically, assuming that $f(\xi_p) > 0$ and that the second derivative of F is bounded in a neighborhood of ξ_p, he proves the following, which has become known as a *Bahadur representation*:

$$\hat{\xi}_{p,n} = \xi_p - \frac{F_n(\xi_p) - p}{f(\xi_p)} + R_n, \tag{3.2}$$

where almost surely (a.s.),

$$R_n = O(n^{-3/4}(\log n)^{1/2}(\log\log n)^{1/4}) \text{ as } n \to \infty. \tag{3.3}$$

The notation "$Y_n = O(g(n))$ a.s." means that there exists a set Ω_0 such that $P(\Omega_0) = 1$ and for each $\omega \in \Omega_0$, there exists a constant $B(\omega)$ such that $|Y_n(\omega)| \leq B(\omega)g(n)$, for n sufficiently large. Kiefer (1967) shows the exact order for R_n is $O(n^{-3/4}(\log\log n)^{3/4})$.

It is well known (e.g., Section 2.3.3 of Serfling (1980)) that $\sqrt{n}(\hat{\xi}_{p,n} - \xi_p)$ converges in distribution as $n \to \infty$ to a normal random variable with mean 0 and variance $p(1-p)/f^2(\xi_p)$, and so does $\sqrt{n}(p - F_n(\xi_p))/f(\xi_p)$ since $F_n(\xi_p)$ is the average of i.i.d. indicator functions. But Bahadur's representation goes further by showing the difference between these two quantities approaches 0 a.s. and provides the rate at which the difference vanishes.

For $p_n = p + O(n^{-1/2})$, define $\hat{\xi}_{p_n,n} = F_n^{-1}(p_n)$. Assuming only that $f(\xi_p) > 0$, Ghosh (1971) shows that

$$\hat{\xi}_{p_n,n} = \xi_p - \frac{F_n(\xi_p) - p_n}{f(\xi_p)} + R_n' \tag{3.4}$$

with R_n' in (3.4) satisfying

$$\sqrt{n}R_n' \xrightarrow{P} 0 \text{ as } n \to \infty, \tag{3.5}$$

where $\xrightarrow{P}$ denotes convergence in probability (p. 330 of Billingsley (1995)). This weaker form of the Bahadur representation in (3.4) and (3.5), which we call a *Bahadur-Ghosh representation*, suffices for most applications, including ours.

One consequence of the Bahadur-Ghosh representation for crude Monte Carlo is that it implies a CLT for the quantile estimator $\hat{\xi}_{p,n} = F_n^{-1}(p)$, as shown in Theorem 10.3

of David and Nagaraja (2003). Taking $p_n = p$ in (3.4), we can write

$$\sqrt{n}(\hat{\xi}_{p,n} - \xi_p) = \frac{\sqrt{n}}{f(\xi_p)}(p - F_n(\xi_p)) + \sqrt{n}R_n'. \tag{3.6}$$

Let $\xrightarrow{L}$ denote convergence in distribution (Billingsley (1995), Section 25), and define $N(a,b^2)$ as a normal distribution with mean a and variance b^2. That $F_n(\xi_p)$ is the sample average of $I(X_i \leq \xi_p)$, $i = 1,2,\ldots,n$, which are i.i.d. with mean p and variance $p(1-p)$, implies that the first term in the right-hand side (RHS) of (3.6) converges in distribution to $N(0, p(1-p)/f^2(\xi_p))$ as $n \to \infty$. Moreover, the second term on the RHS of (3.6) vanishes in probability as $n \to \infty$ by (3.5). Hence, $\sqrt{n}(\hat{\xi}_{p,n} - \xi_p) \xrightarrow{L} N\left(0, p(1-p)/f^2(\xi_p)\right)$ as $n \to \infty$ by Slutsky's theorem (Serfling (1980), p. 19).

CHAPTER 4

BAHADUR-GHOSH REPRESENTATION WHEN APPLYING VRTS

Because crude Monte Carlo is sometimes inefficient for estimating quantiles, we may try to obtain improved quantile estimators by applying a variance-reduction technique (VRT). VRTs often change the way samples are generated, or collect additional data, and this leads to different estimators of the CDF. We then invert the resulting estimated CDF to obtain a quantile estimator. We now establish a general mathematical framework that will allow us to show that a Bahadur-Ghosh representation holds when applying different VRTs. We will then apply this framework in subsequent chapters to examine importance sampling (IS), stratified sampling (SS), antithetic variates (AV), control variates (CV) and certain combinations of them.

Let $\tilde{F}_n$ denote a generic estimator of the CDF F obtained when using a VRT, where n is the "computational budget," which we define differently for various simulation methods. For example, when applying IS, n is the number of samples generated from a new distribution F_* obtained from a change of measure (see Chapter 5 for details). When employing AV, n denotes the number of antithetic pairs (Chapter 6). For CV, n is the number of pairs of output and control collected (Chapter 7).

Now set $\tilde{\xi}_{p_n,n} = \tilde{F}_n^{-1}(p_n)$ as the VRT estimator of the p_n-quantile for $p_n = p + O(n^{-1/2})$. Analogous to the Bahadur-Ghosh representation in (3.4) for crude Monte Carlo, we write

$$\tilde{\xi}_{p_n,n} = \xi_p - \frac{\tilde{F}_n(\xi_p) - p_n}{f(\xi_p)} + \tilde{R}_n. \tag{4.1}$$

To obtain a result similar to (3.5), we require that $\tilde{F}_n$ satisfies the following assumptions.

Assumption A1 *$P(M_n) \to 1$ as $n \to \infty$, where M_n is the event that $\tilde{F}_n(x)$ is monotonically increasing in x.*

This assumption allows for the estimated CDF to not necessarily be monotonically increasing in x, but the probability of this occurring must vanish as n increases. For many (but not all) VRTs, $\tilde{F}_n(x)$ will always be monotonically increasing in x for each n, so Assumption A1 will trivially hold.

Assumption A2 *For every $a_n = O(n^{-1/2})$,*

$$\frac{\sqrt{n}}{f(\xi_p)}\left[(F(\xi_p+a_n)-F(\xi_p))-(\tilde{F}_n(\xi_p+a_n)-\tilde{F}_n(\xi_p))\right]\xrightarrow{P}0$$

as $n \to \infty$.

This assumption requires that the scaled difference in the actual CDF and the CDF estimator, both evaluated over an interval of length of order $n^{-1/2}$ with an endpoint ξ_p, vanishes in probability as $n \to \infty$. We provide a set of sufficient conditions for Assumption A2 in Section 4.1.2.

We also require that a CLT holds for the CDF estimator at ξ_p.

Assumption A3 $\sqrt{n}\left[\tilde{F}_n(\xi_p)-F(\xi_p)\right]\xrightarrow{L}N(0,\psi_p^2)$ *as $n \to \infty$ for some $0 < \psi_p < \infty$.*

We will show in the later chapters that A1–A3 hold for the VRTs we consider. In the case of crude Monte Carlo, where F_n in (3.1) replaces $\tilde{F}_n$, Assumption A1 holds since $F_n(x)$ is monotonically increasing in x for each n. Moreover, Ghosh (1971) (also see David and Nagaraja (2003), p. 287) shows that Assumptions A2 and A3 hold by exploiting the fact that $nF_n(x)$ has a binomial distribution with parameters n and $F(x)$. Also, for crude Monte Carlo, $\psi_p^2 = \mathrm{Var}[I(X \le \xi_p)] = p(1-p)$ in Assumption A3.

The following theorem shows that for a CDF estimator obtained when applying VRTs and satisfying our assumptions, a Bahadur-Ghosh representation holds for the resulting quantile estimator.

Theorem 1 *Suppose $\tilde{F}_n$ satisfies Assumptions A1–A3. If $f(\xi_p) > 0$ and $p_n - p = O(n^{-1/2})$, then (4.1) holds with*

$$\sqrt{n}\tilde{R}_n \xrightarrow{P} 0 \text{ as } n \to \infty. \tag{4.2}$$

A simple consequence of Theorem 1 is that $\tilde{\xi}_{p,n} = \tilde{F}_n^{-1}(p)$ is a consistent estimator of ξ_p, which we can see as follows. Assumption A3 implies $\tilde{F}_n(\xi_p) \xrightarrow{P} F(\xi_p) = p$ by Theorem 2.3.4 of Lehmann (1999). Hence, (4.1) ensures

$$\tilde{\xi}_{p,n} = \xi_p + \frac{p - \tilde{F}_n(\xi_p)}{f(\xi_p)} + \tilde{R}_n \xrightarrow{P} \xi_p \tag{4.3}$$

as $n \to \infty$ by Theorem 1 and Slutsky's theorem. Moreover, Theorem 1 also implies the following CLT for the VRT estimator of the quantile.

Theorem 2 *Suppose $\tilde{F}_n$ satisfies Assumptions A1–A3. If $f(\xi_p) > 0$, then*

$$\frac{\sqrt{n}}{\kappa_p}(\tilde{\xi}_{p,n} - \xi_p) \xrightarrow{L} N(0,1) \tag{4.4}$$

as $n \to \infty$, where $\kappa_p = \psi_p \phi_p$, ψ_p is defined in Assumption A3, and $\phi_p = 1/f(\xi_p)$.

For crude Monte Carlo or any VRT, the parameter κ_p in (4.4) always has the same basic form of $\psi_p \phi_p$. The value of ψ_p from Assumption A3 depends on the particular VRT used (and equals $\sqrt{p(1-p)}$ for crude Monte Carlo), but ϕ_p does not change. Thus, efficient quantile estimation typically focuses on applying a VRT to reduce ψ_p.

The CLT in Theorem 2 provides a way to construct confidence intervals for a quantile estimated using VRTs, provided we have consistent estimators of ψ_p and ϕ_p. To handle ϕ_p, we first note that $\frac{d}{dp}F^{-1}(p) = 1/f(\xi_p) = \phi_p$ by the chain rule of differentiation, and we propose below some finite-difference estimators (e.g., Section 7.1 of Glasserman (2004)) of ϕ_p. Let $c \neq 0$ be any constant, and define $p_n = p + cn^{-1/2}$ and $p_n' = p - cn^{-1/2}$. Then define the estimators

$$\tilde{\phi}_{p,n,1}(c) = \frac{\sqrt{n}}{c}(\tilde{\xi}_{p_n,n} - \tilde{\xi}_{p,n}), \tag{4.5}$$

$$\tilde{\phi}_{p,n,2}(c) = \frac{\sqrt{n}}{2c}(\tilde{\xi}_{p_n,n} - \tilde{\xi}_{p_n',n}). \tag{4.6}$$

Note that $\tilde{\phi}_{p,n,1}(c)$ is a forward (resp., backward) finite-difference estimator of ϕ_p when $c > 0$ (resp., $c < 0$) since $\tilde{\phi}_{p,n,1}(c) = [\tilde{F}_n^{-1}(p + cn^{-1/2}) - \tilde{F}_n^{-1}(p)]/(cn^{-1/2})$. Similarly, $\tilde{\phi}_{p,n,2}(c)$ is a central finite difference. To define additional estimators of ϕ_p, let $c_1, \ldots, c_r$ and $w_1, \ldots, w_r$ be any nonzero constants (some possibly negative) with $\sum_{j=1}^r w_j = 1$. Then define estimators

$$\bar{\phi}_{p,n,i}(c_1, \ldots, c_r) = \sum_{j=1}^{r} w_j \tilde{\phi}_{p,n,i}(c_j), \text{ for } i = 1, 2, \tag{4.7}$$

which are weighted combinations of the previous finite-difference estimators. The following theorem shows that all of our estimators of ϕ_p are consistent. In addition, if we also have a consistent estimator $\tilde{\psi}_{p,n}$ of ψ_p in Assumption A3, then we can consistently estimate $\kappa_p = \psi_p \phi_p$ in (4.4) by taking the product of the consistent estimators of ψ_p and ϕ_p, and the CLT in (4.4) still holds when κ_p is replaced by its consistent estimator.

Theorem 3 *Assume the conditions of Theorem 2 hold. Then for any nonzero constants c and $c_1, \ldots, c_r$,*

$$\tilde{\phi}_{p,n,i}(c) \xrightarrow{P} \phi_p, \tag{4.8}$$

$$\bar{\phi}_{p,n,i}(c_1, \ldots, c_r) \xrightarrow{P} \phi_p, \tag{4.9}$$

as $n \to \infty$, for $i = 1, 2$. Moreover, if $\tilde{\psi}_{p,n} \xrightarrow{P} \psi_p$ as $n \to \infty$, then

$$\frac{\sqrt{n}}{\tilde{\kappa}_{p,n}}(\tilde{\xi}_{p,n} - \xi_p) \xrightarrow{L} N(0,1) \tag{4.10}$$

as $n \to \infty$, with $\tilde{\kappa}_{p,n} = \tilde{\psi}_{p,n}\tilde{\phi}_{p,n,i}(c)$ or $\tilde{\kappa}_{p,n} = \tilde{\psi}_{p,n}\bar{\phi}_{p,n,i}(c_1, \ldots, c_r)$ for $i = 1, 2$.

Hong (2009), Liu and Hong (2009) and Fu et al. (2009) develop consistent estimators for derivatives of quantiles with respect to certain model parameters, but their methods do not apply for estimating $\phi_p = \frac{d}{dp}F^{-1}(p)$ and/or when using VRTs. When applying crude

Monte Carlo (i.e., i.i.d. sampling), Bloch and Gastwirth (1968) and Bofinger (1975) show that estimators analogous to $\tilde{\phi}_{p,n,i}(c)$, $i = 1,2$, in (4.5) and (4.6) consistently estimate ϕ_p. Moreover, Babu (1986) considers estimators that are weighted combinations as in (4.7) for i.i.d. sampling. All of their consistency proofs rely on representing each i.i.d. sample X_i as $X_i = F^{-1}(U_i)$, where U_i is uniformly distributed on the unit interval. However, these arguments do not generalize when applying VRTs such as importance sampling, so we require a different approach to establish (4.8) and (4.9). In particular, Corollary 2.5.2 of Serfling (1980) provides a method that exploits the a.s. Bahadur representation from (3.2) and (3.3) for i.i.d. sampling to consistently estimate ϕ_p, and we modify this idea to work instead with a Bahadur-Ghosh representation and VRTs.

We now provide an algorithm that shows how to use (4.10) to construct an asymptotically valid confidence interval for the quantile ξ_p when applying a generic VRT.

Algorithm

Goal: Construct a point estimate and $100(1-\alpha)\%$ confidence interval for ξ_p.

Given: CDF estimator $\tilde{F}_n$ based on a computational budget of n; $r \geq 1$ nonzero constants $c_1, c_2, \ldots, c_r$ and $w_1, w_2, \ldots, w_r$ with $\sum_{j=1}^{r} w_j = 1$; and a consistent estimator $\tilde{\psi}_{p,n}$ of ψ_p.

Steps:

1. Compute the point estimator $\tilde{\xi}_{p,n} = \tilde{F}_n^{-1}(p)$ of ξ_p.
2. Compute $\tilde{\kappa}_{p,n} = \tilde{\psi}_{p,n}\bar{\phi}_{p,n,i}(c_1, \ldots, c_r)$ for either $i = 1$ or $i = 2$, where $\bar{\phi}_{p,n,i}(c_1, \ldots, c_r)$ is defined in (4.7).
3. An asymptotically valid $100(1-\alpha)\%$ confidence interval for ξ_p is then $(\tilde{\xi}_{p,n} \pm z_{1-\alpha/2}\tilde{\kappa}_{p,n}/\sqrt{n})$, where $z_\beta = \Phi^{-1}(\beta)$ and Φ is the CDF of a $N(0,1)$ random variable.

In the following chapters we present explicit formulae for the estimators $\tilde{F}_n$ and $\tilde{\psi}_{p,n}$ in the above algorithm for different VRTs, and these estimators can then be used to compute $\tilde{\xi}_{p,n}$ and $\bar{\phi}_{p,n,i}(c_1,\ldots,c_r)$. To simplify notation, we will continue to use the same variables $\tilde{F}_n$, $\tilde{\xi}_{p,n}$, $\tilde{\kappa}_{p,n}$, $\tilde{\psi}_{p,n}$ and $\bar{\phi}_{p,n,i}(c_1,\ldots,c_r)$ in each case rather than develop new notation for each different VRT.

4.1 Proofs

4.1.1 Proof of Theorem 1

To prove Theorem 1, we will first use the following lemma established by Ghosh (1971); also see pp. 286–287 of David and Nagaraja (2003) for more details. We then transform $\tilde{R}_n$ in (4.1) in terms of the two sets of variates described in the lemma to complete the proof as required.

Lemma 1 *Let (V_n, W_n), $n = 1,2,\ldots$, be a sequence of pairs of random variables such that*

1. *for all $\delta > 0$, there exists $\gamma \equiv \gamma(\delta)$ and $n_0 \equiv n_0(\gamma,\delta)$ such that $P\{|W_n| > \gamma\} < \delta$ for $n \geq n_0$ (i.e, $W_n = O_p(1)$);*

2. *for every y and every $\varepsilon > 0$,*

   (a) $\lim_{n\to\infty} P\{V_n \leq y, W_n \geq y+\varepsilon\} = 0$,

   (b) $\lim_{n\to\infty} P\{V_n \geq y, W_n \leq y+\varepsilon\} = 0$.

Then

$$V_n - W_n \xrightarrow{P} 0 \quad as \quad n \to \infty. \tag{4.11}$$

First define $\eta_n = (p_n - p)/f(\xi_p)$. Set

$$\begin{aligned} V_n &= \sqrt{n}(\tilde{\xi}_{p_n,n} - \xi_p - \eta_n), \\ W_n &= \frac{\sqrt{n}}{f(\xi_p)}[p - \tilde{F}_n(\xi_p)], \end{aligned} \tag{4.12}$$

and note that (4.1) implies

$$\sqrt{n}\tilde{R}_n = V_n - W_n. \tag{4.13}$$

We now show that (V_n, W_n) satisfies conditions 1 and 2 of Lemma 1 to establish our theorem.

First, Assumption A3 implies condition 1 of Lemma 1 by Theorem 2.3.2 of Lehmann (1999). We next prove condition 2(a) of Lemma 1 holds. Recall M_n is the event that $\tilde{F}_n(x)$ is monotonically increasing in x, and let M_n^c be its complement. Then

$$\begin{aligned}\{V_n \le y\} &= \{\tilde{\xi}_{p_n,n} \le \xi_p + \eta_n + yn^{-1/2}, M_n\} \cup \{V_n \le y, M_n^c\} \\ &\subseteq \{\tilde{F}_n(\xi_p + \eta_n + yn^{-1/2}) \ge p_n\} \cup M_n^c \\ &= \{Z_n \le y_n\} \cup M_n^c,\end{aligned}$$

where

$$\begin{aligned}Z_n &= \frac{\sqrt{n}}{f(\xi_p)}\left[F(\xi_p + \eta_n + yn^{-1/2}) - \tilde{F}_n(\xi_p + \eta_n + yn^{-1/2})\right], \\ y_n &= \frac{\sqrt{n}}{f(\xi_p)}\left[F(\xi_p + \eta_n + yn^{-1/2}) - p_n\right].\end{aligned}$$

Fix $\varepsilon > 0$, and since

$$P\{V_n \le y, W_n \ge y + \varepsilon\} \le P\{Z_n \le y_n, W_n \ge y + \varepsilon\} + P(M_n^c), \tag{4.14}$$

establishing condition 2(a) of Lemma 1 is equivalent to proving the first term on the RHS of (4.14) approaches 0 as $n \to \infty$ because of Assumption A1. Since F is assumed to be differentiable at ξ_p, Young's form of Taylor's theorem (Hardy (1952), p. 278) implies

$$\begin{aligned}y_n &= \frac{\sqrt{n}}{f(\xi_p)}\left\{F(\xi_p) + (\eta_n + yn^{-1/2})[f(\xi_p) + o(1)] - p_n\right\} \\ &= y + \frac{\sqrt{n}}{f(\xi_p)}\left\{p + p_n - p + \left(\frac{p_n - p}{f(\xi_p)} + yn^{-1/2}\right)o(1) - p_n\right\}\end{aligned}$$

since $F(\xi_p) = p$. It is thus clear that $y_n \to y$ as $n \to \infty$ since $p_n - p = O(n^{-1/2})$. Therefore, there exists n_0 such that $|y - y_n| < \varepsilon/2$ for all $n \geq n_0$, so taking the difference of the two inequalities in the first term of the RHS of (4.14) gives

$$\begin{aligned} P\{Z_n \leq y_n, W_n \geq y+\varepsilon\} &\leq P\{|W_n - Z_n| \geq \varepsilon + y - y_n\} \\ &\leq P\left\{|W_n - Z_n| \geq \frac{\varepsilon}{2}\right\} \text{ for } n \geq n_0. \end{aligned}$$

Hence, to show the left-hand side of (4.14) vanishes as $n \to \infty$, it suffices to prove

$$Z_n - W_n \xrightarrow{P} 0 \text{ as } n \to \infty. \tag{4.15}$$

Note that

$$Z_n - W_n = \frac{\sqrt{n}}{f(\xi_p)} \left[(F(\xi_p + \eta_n + yn^{-1/2}) - F(\xi_p)) - (\tilde{F}_n(\xi_p + \eta_n + yn^{-1/2}) - \tilde{F}_n(\xi_p)) \right],$$

and we have $p_n - p = O(n^{-1/2})$ by assumption, so $\eta_n + yn^{-1/2} = O(n^{-1/2})$. Consequently, (4.15) holds by Assumption A2. Thus, the (V_n, W_n) pair satisfies condition 2(a) in Lemma 1; condition 2(b) of the lemma may be similarly established, so (4.11) holds. Recalling (4.13) then completes the proof.

4.1.2 Sufficient Conditions for Assumption A2

As an alternative to directly showing Assumption A2 holds, we now provide a set of sufficient conditions for the assumption.

Condition C1 *For every $a_n = O(n^{-1/2})$, $E[\tilde{F}_n(\xi_p + a_n) - \tilde{F}_n(\xi_p)] = F(\xi_p + a_n) - F(\xi_p) + r_n(a_n)/n^{1/2}$, where $r_n(a_n) \to 0$ as $n \to \infty$.*

This condition requires the expectation of the difference in the CDF estimator at ξ_p and a perturbation from ξ_p of order $n^{-1/2}$ to equal the difference of the CDF at the two values, plus a bias term that goes to 0 sufficiently fast. Note that Condition C1 holds when $\tilde{F}_n(x)$ is an unbiased estimator of $F(x)$ for all x.

Condition C2 *For every* $a_n = O(n^{-1/2})$, $E\left[\tilde{F}_n(\xi_p + a_n) - \tilde{F}_n(\xi_p)\right]^2 = [F(\xi_p + a_n) - F(\xi_p)]^2 + s_n(a_n)/n$, *where* $s_n(a_n) \to 0$ *as* $n \to \infty$.

This condition considers the second moment of the difference in the CDF estimator at ξ_p and a perturbation from ξ_p of order $n^{-1/2}$. The condition requires that the second moment can be expressed as the square of the difference in probabilities due to this perturbation with a remainder approaching 0 sufficiently fast.

Proposition 1 *Conditions C1 and C2 together imply Assumption A2.*

PROOF. Let $a_n = O(n^{-1/2})$. Define W_n as in (4.12), and let

$$Z_n = \frac{\sqrt{n}}{f(\xi_p)}\left[F(\xi_p + a_n) - \tilde{F}_n(\xi_p + a_n)\right].$$

To establish Assumption A2, it suffices to prove $E[(Z_n - W_n)^2] \to 0$ as $n \to \infty$; e.g., see Theorem 2.1.1 of Lehmann (1999). Let $d_n = F(\xi_p + a_n) - F(\xi_p)$ and $D_n = \tilde{F}_n(\xi_p + a_n) - \tilde{F}_n(\xi_p)$. Then

$$E[(Z_n - W_n)^2] = \frac{n}{f^2(\xi_p)}\left[d_n^2 - 2d_nE(D_n) + E(D_n^2)\right]. \tag{4.16}$$

Conditions C1 and C2 imply $E[D_n] = d_n + r_n(a_n)/\sqrt{n}$ and $E[D_n^2] = d_n^2 + s_n(a_n)/n$. Putting these into (4.16) yields

$$E[(Z_n - W_n)^2] = \frac{-2\sqrt{n}}{f^2(\xi_p)}d_nr_n(a_n) + \frac{s_n(a_n)}{f^2(\xi_p)}. \tag{4.17}$$

According to Young's form of Taylor's theorem, $F(\xi_p + a_n) - F(\xi_p) = (f(\xi_p) + o(1))a_n$. Since $a_n = O(n^{-1/2})$, we have $d_n = F(\xi_p + a_n) - F(\xi_p) = O(n^{-1/2})$. Now we use the properties of r_n and s_n in Conditions C1 and C2, which ensure (4.17) converges to 0 as $n \to \infty$, so Assumption A2 holds.

4.1.3 Proof of Theorem 2

By (4.1), we have

$$\frac{\sqrt{n}}{\kappa_p}(\tilde{\xi}_{p,n}-\xi_p)=\frac{\sqrt{n}}{\psi_p}(p-\tilde{F}_n(\xi_p))+\frac{\sqrt{n}}{\kappa_p}\tilde{R}_n. \tag{4.18}$$

The first term on the RHS of (4.18) converges in distribution to a standard normal by Assumption A3, and the second term converges in probability to 0 by (4.2). Thus, the result follows from Slutsky's theorem.

4.1.4 Proof of Theorem 3

By Theorem 1 and (4.1), we have

$$\begin{aligned}\tilde{\xi}_{p,n}&=\xi_p-\frac{\tilde{F}_n(\xi_p)-p}{f(\xi_p)}+\tilde{R}_{n,1},\\ \tilde{\xi}_{p_n,n}&=\xi_p-\frac{\tilde{F}_n(\xi_p)-p_n}{f(\xi_p)}+\tilde{R}_{n,2},\end{aligned}$$

where $\sqrt{n}\tilde{R}_{n,i}\xrightarrow{P}0$ as $n\to\infty$ for $i=1,2$. Thus, since $p_n=p+cn^{-1/2}$,

$$\tilde{\phi}_{p,n,1}=\frac{\sqrt{n}}{c}(\tilde{\xi}_{p_n,n}-\tilde{\xi}_{p,n})=\frac{\sqrt{n}}{c}\left[\frac{cn^{-1/2}}{f(\xi_p)}+\tilde{R}_{n,2}-\tilde{R}_{n,1}\right]\xrightarrow{P}\frac{1}{f(\xi_p)}=\phi_p,$$

as $n\to\infty$, which completes the proof of (4.8) for $i=1$. We can similarly prove (4.8) for $i=2$. Also, (4.9) then follows from Slutsky's theorem since $\sum_{j=1}^k w_j=1$.

To prove (4.10), we only consider when $\tilde{\kappa}_{p,n}=\tilde{\psi}_{p,n}\bar{\phi}_{p,n,i}(c_1,\ldots,c_r)$ since $\tilde{\kappa}_{p,n}=\tilde{\psi}_{p,n}\tilde{\phi}_{p,n,i}(c)$ is just a special case with $r=1$. Recall $\tilde{\psi}_{p,n}\xrightarrow{P}\psi_p$ by assumption, and also (4.9) holds, where in both cases the limits are deterministic. Hence, Slutsky's theorem implies $\tilde{\kappa}_{p,n}\xrightarrow{P}\kappa_p$ as $n\to\infty$, where κ_p is deterministic. Then combining this with (4.4) using Slutsky's theorem establishes (4.10).

CHAPTER 5

QUANTILE ESTIMATION USING IS+SS

IS and SS are two VRTs used to improve the efficiency of simulations, and combining them may further enhance the effect. Before describing a combined IS+SS quantile estimator developed by Glasserman et al. (2000b), we start by applying just IS alone without SS, as in Glynn (1996).

We first explain how to apply IS in the simple case when the output X has CDF F and density function f. Let F_* be another CDF, and let f_* be the density function of F_* with the property that for each t, $f(t) > 0$ implies that $f_*(t) > 0$. For example, if F is normal with mean α and variance σ^2, then we can choose F_* to also be normal but with different mean α_* and variance σ_*^2. Define E_* to be expectation under CDF F_*. Also, define $L(t) = f(t)/f_*(t)$ to be the likelihood ratio at t. Then we can write

$$
\begin{aligned}
F(x) = E\left[I(X \leq x)\right] &= \int I(t \leq x)\, f(t)\, dt = \int I(t \leq x)\, \frac{f(t)}{f_*(t)}\, f_*(t)\, dt \\
&= \int I(t \leq x)\, L(t)\, f_*(t)\, dt = E_*\left[I(X \leq x)\, L(X)\right].
\end{aligned}
$$

The above suggests that to estimate $F(x)$ using IS, we generate i.i.d. samples $X_1, \ldots, X_n$ of X from CDF F_* and average $I(X_i \leq x)L(X_i)$, $i = 1, \ldots, n$.

As explained in Glynn and Iglehart (1989), IS applies more generally than the situation we just described. Let P be the original probability measure governing the stochastic system or process being studied, and let P_* be another probability measure such that for each (measurable) event A, $P(A) > 0$ implies $P_*(A) > 0$; i.e., P is absolutely continuous (p. 422 of Billingsley (1999)) with respect to P_*. Define E_* as the expectation operator under the IS probability measure P_*, and define the likelihood ratio $L = dP/dP_*$, which is also called the Radon-Nikodym derivative of P with respect to P_* (p. 423 of Billingsley

(1999)). Then we have

$$F(x) = E\left[I(X \le x)\right] = \int I(X \le x)\, dP = \int I(X \le x)\, L\, dP_*$$
$$= E_*\left[I(X \le x)\, L\right], \tag{5.1}$$

which is known as applying a change of measure. This motivates estimating $F(x)$ as follows. Generate i.i.d. samples $(X_1, L_1), \ldots, (X_n, L_n)$ of (X, L) using P_*, and the IS estimator of F is then

$$\tilde{F}_{n,\mathrm{IS}}(x) = \frac{1}{n} \sum_{i=1}^{n} I(X_i \le x)\, L_i. \tag{5.2}$$

Inverting $\tilde{F}_{n,\mathrm{IS}}$ results in the IS quantile estimator.

Recall the crude Monte Carlo estimator of F is given in (3.1), where the X_i in (3.1) are generated using the original measure P induced by CDF F. By Theorem 2, the key to applying IS for estimating ξ_p is choosing P_* so that $\mathrm{Var}_*[I(X \le \xi_p)\, L] < \mathrm{Var}[I(X \le \xi_p)] = p(1-p)$ to achieve a variance reduction (relative to crude Monte Carlo), where Var_* denotes variance under measure P_*. Glynn (1996) and Glasserman et al. (2000b) present particular choices of P_* for various settings.

To additionally incorporate stratified sampling, we identify a *stratification variable* Y such that X and Y are dependent. We partition the support of Y into $k < \infty$ strata $S_1, \ldots, S_k$ such that each $\lambda_i \equiv P_*\{Y \in S_i\} > 0$ is known and $\sum_{i=1}^{k} \lambda_i = 1$. For example, the strata may be disjoint intervals, and Sections 4.3 and 9.2.3 of Glasserman (2004) discuss several other choices for selecting the strata. Therefore, by (5.1), we can write

$$F(x) = \sum_{i=1}^{k} \lambda_i E_*\left[I(X \le x)\, L \mid Y \in S_i\right]. \tag{5.3}$$

Note that we derived (5.3) by first applying IS and then using stratification, so Y is distributed under the IS measure P_*. (Instead applying SS first and then IS leads to a different representation for F and thus a different estimator; see Glasserman et al. (2000a) for details.)

IS+SS estimation of F entails replacing each conditional expectation in (5.3) with an average of samples from the corresponding stratum. We now provide details on this approach as developed in Glasserman et al. (2000a). Define the sample size in each stratum i as $n_i = n\gamma_i$, where the $\gamma_i > 0$ are user-specified constants satisfying $\sum_{i=1}^{k} \gamma_i = 1$. (We later discuss possible choices for γ_i.) For simplicity, we assume that n_i is always an integer, so the total number of samples across all strata is $\sum_{i=1}^{k} n_i = \sum_{i=1}^{k} n\gamma_i = n$. For each stratum $i = 1, \ldots, k$, we use the IS measure P_* to draw n_i samples Y_{ij}, $j = 1, \ldots, n_i$, of Y conditioned to lie in S_i. Then for each $j = 1, \ldots, n_i$, generate X_{ij} as a sample of X having the conditional IS distribution of X given $Y = Y_{ij}$, and let L_{ij} be the corresponding likelihood ratio. We thus have n_i i.i.d. samples (X_{ij}, Y_{ij}, L_{ij}), $j = 1, \ldots, n_i$, of (X, Y, L) from stratum i. (Glasserman et al. (2000b) employ a "bin tossing" method to generate samples of the triple (X_{ij}, Y_{ij}, L_{ij}).) The (X_{ij}, Y_{ij}, L_{ij}) sample triples across strata are generated independently. Then the IS+SS estimator of the CDF is

$$\tilde{F}_n(x) = \sum_{i=1}^{k} \lambda_i \frac{1}{n_i} \sum_{j=1}^{n_i} I(X_{ij} \leq x) L_{ij}. \tag{5.4}$$

We allow for $P_* = P$, in which case the likelihood ratio $L \equiv 1$ and we do not apply IS. Also, the number of strata may be $k = 1$, in which case there is no stratified sampling. Hence, the following results for IS+SS encompass crude Monte Carlo, IS-only and SS-only as special cases.

Theorem 4 *Suppose $f(\xi_p) > 0$, and for each stratum i, suppose there exists $\varepsilon > 0$ and $\delta > 0$ such that $E_*[I(X_{ij} < \xi_p + \delta) L_{ij}^{2+\varepsilon}] < \infty$. Let $\tilde{F}_n$ be the IS+SS estimator of F defined in (5.4). Then $\tilde{F}_n$ satisfies Assumptions A1–A3, where $\psi_p^2 = \sum_{i=1}^{k} \lambda_i^2 \zeta_i^2 / \gamma_i$ is the variance constant in Assumption A3 with*

$$\zeta_i^2 = E_* \left[I(X_{ij} \leq \xi_p) L_{ij}^2 \right] - P^2(X \leq \xi_p \,|\, Y \in S_i). \tag{5.5}$$

Thus, Theorem 1 implies $\tilde{\xi}_{p_n,n} = \tilde{F}_n^{-1}(p_n)$ *with* $p_n - p = O(n^{-1/2})$ *satisfies the Bahadur-Ghosh representation in (4.1) and (4.2).*

We recently found out that independently of our work, Sun and Hong (2010) establish that the IS-only quantile estimator obtained by inverting $\tilde{F}_{n,\mathrm{IS}}$ in (5.2) satisfies an a.s. Bahadur representation analogous to (3.2) and (3.3) using a different proof technique and under a stronger set of assumptions than we use. Specifically, they further assume that the density f is positive and continuously differentiable in a neighborhood of ξ_p and that the likelihood ratio $L(x)$ is bounded in a neighborhood of ξ_p. Also, they do not consider IS+SS (nor AV and CV), as we do. Moreover, they examine only the case of fixed p and not perturbed p_n, the latter of which is essential for our approach for developing confidence intervals for ξ_p.

The next result shows that the IS+SS quantile estimator satisfies a CLT.

Theorem 5 *Suppose* $f(\xi_p) > 0$, *and for each stratum i, suppose there exists* $\varepsilon > 0$ *and* $\delta > 0$ *such that* $E_*[I(X_{ij} < \xi_p + \delta)L_{ij}^{2+\varepsilon}] < \infty$. *Then* $\tilde{\xi}_{p,n} = \tilde{F}_n^{-1}(p)$ *with* $\tilde{F}_n$ *defined in (5.4) satisfies*

$$\frac{\sqrt{n}}{\kappa_p}(\tilde{\xi}_{p,n} - \xi_p) \xrightarrow{L} N(0,1), \tag{5.6}$$

$$\frac{\sqrt{n}}{\tilde{\kappa}_{p,n}}(\tilde{\xi}_{p,n} - \xi_p) \xrightarrow{L} N(0,1), \tag{5.7}$$

as $n \to \infty$, *where* $\kappa_p = \psi_p \phi_p$, $\phi_p = 1/f(\xi_p)$, ψ_p *is defined in Theorem 4, the estimator* $\tilde{\kappa}_{p,n} = \tilde{\psi}_{p,n}\bar{\phi}_{p,n,i}(c_1,\ldots,c_r)$, $\bar{\phi}_{p,n,i}(c_1,\ldots,c_r)$ *is from (4.7) with nonzero constants* $c_1,\ldots,c_r$ *and* $w_1,\ldots,w_r$ *satisfying* $\sum_{j=1}^r w_j = 1$ *for* $i = 1$ *or* 2, $\tilde{\psi}_{p,n}^2 = \sum_{i=1}^k \lambda_i^2 \tilde{\zeta}_{i,n}^2/\gamma_i$, *and*

$$\tilde{\zeta}_{i,n}^2 = \left(\frac{1}{n_i}\sum_{j=1}^{n_i} I(X_{ij} \le \tilde{\xi}_{p,n})L_{ij}^2\right) - \left(\frac{1}{n_i}\sum_{j=1}^{n_i} I(X_{ij} \le \tilde{\xi}_{p,n})L_{ij}\right)^2. \tag{5.8}$$

We now discuss possible choices for the stratum allocation weights γ_i. Setting $\gamma_i = \lambda_i$ for each stratum i guarantees that the CDF estimator $\tilde{F}_n(x)$ in (5.4) has no greater variance than $\tilde{F}_{n,\mathrm{IS}}(x)$ in (5.2), which implies the IS+SS quantile estimator has no greater variance than the IS quantile estimator does; e.g., see p. 217 of Glasserman (2004). We can do even better by choosing γ_i to minimize $\psi_p^2 = \sum_{i=1}^k \lambda_i^2 \zeta_i^2/\gamma_i$ subject to $\sum_{i=1}^k \gamma_i = 1$ and $\gamma_i \geq 0$. The stratum allocation weights solving this optimization problem are given by $\gamma_i^* = \lambda_i \zeta_i / (\sum_{j=1}^k \lambda_j \zeta_j)$; e.g., see p. 217 of Glasserman (2004). Since the ζ_i are typically unknown, we might first estimate them from pilot runs, and then use these values to estimate the γ_i^*.

The right tail of $\tilde{F}_{n,\mathrm{IS}}$ in (5.2) may not behave as a proper CDF since it is possible (and indeed likely) for $\lim_{x\to\infty} \tilde{F}_{n,\mathrm{IS}}(x) = a$ with $a < 1$ or $a > 1$. To avoid such a situation, Glynn (1996) also proposes another IS estimator of the CDF, $\tilde{F}'_{n,\mathrm{IS}}(x) = 1 - \frac{1}{n}\sum_{i=1}^n I(X_i > x)L(X_i)$, which can be more effective when estimating a quantile for $p \approx 1$. (However, we may instead have $\lim_{x\to-\infty} \tilde{F}'_{n,\mathrm{IS}}(x) = b$ with $b < 0$ or $b > 0$, so $\tilde{F}'_{n,\mathrm{IS}}$ may not be appropriate when estimating a quantile for $p \approx 0$.) Glasserman et al. (2000b) develop the corresponding IS+SS estimator of F:

$$\tilde{F}'_n(x) = 1 - \sum_{i=1}^k \lambda_i \frac{1}{n_i} \sum_{j=1}^{n_i} I(X_{ij} > x)L_{ij}. \tag{5.9}$$

The following two theorems, in which primed variables replace non-primed variables from before, show that quantile estimators based on inverting $\tilde{F}'_n$ satisfy a Bahadur-Ghosh representation and CLTs. They can be shown by straightforward modifications of the proofs of Theorems 4 and 5.

Theorem 6 *Suppose $f(\xi_p) > 0$, and for each stratum i, suppose there exists $\varepsilon > 0$ and $\delta > 0$ such that $E_*[I(X_{ij} > \xi_p - \delta)L_{ij}^{2+\varepsilon}] < \infty$. Let $\tilde{F}'_n$ be the IS+SS estimator of F defined in (5.9). Then $\tilde{F}'_n$ satisfies Assumptions A1–A3, where $\psi_p'^2 = \sum_{i=1}^k \lambda_i^2 \zeta_i'^2/\gamma_i$ is the variance constant in Assumption A3 with $\zeta_i'^2 = E_*\left[I(X_{ij} > \xi_p)L_{ij}^2\right] - P^2(X > \xi_p \,|\, Y \in S_i)$. Thus, Theorem 1 implies $\tilde{\xi}'_{p_n,n} = \tilde{F}_n'^{-1}(p_n)$ with $p_n - p = O(n^{-1/2})$ satisfies the Bahadur-Ghosh representation in (4.1) and (4.2).*

Theorem 7 *Suppose $f(\xi_p) > 0$, and for each stratum i, suppose there exists $\varepsilon > 0$ and $\delta > 0$ such that $E_*[I(X_{ij} > \xi_p - \delta)L_{ij}^{2+\varepsilon}] < \infty$. Suppose $\tilde{\xi}'_{p,n} = \tilde{F}_n'^{-1}(p)$ with $\tilde{F}'_n$ defined in (5.9) satisfies*

$$\frac{\sqrt{n}}{\kappa'_p}(\tilde{\xi}'_{p,n} - \xi_p) \xrightarrow{L} N(0,1), \tag{5.10}$$

$$\frac{\sqrt{n}}{\tilde{\kappa}'_{p,n}}(\tilde{\xi}'_{p,n} - \xi_p) \xrightarrow{L} N(0,1), \tag{5.11}$$

as $n \to \infty$, where $\kappa'_p = \psi'_p \phi_p$, $\phi_p = 1/f(\xi_p)$, ψ'_p is defined in Theorem 6, the estimator $\tilde{\kappa}'_{p,n} = \tilde{\psi}'_{p,n}\tilde{\phi}'_{p,n,i}(c_1,\ldots,c_r)$, $\tilde{\phi}'_{p,n,i}(c_1,\ldots,c_r)$ uses $\tilde{F}'_n$ in (4.7) with nonzero constants $c_1,\ldots,c_r$ and $w_1,\ldots,w_r$ satisfying $\sum_{j=1}^{r} w_j = 1$ for $i = 1$ or 2, $\tilde{\psi}'^2_{p,n} = \sum_{i=1}^{k}\lambda_i^2\tilde{\zeta}'^2_{i,n}/\gamma_i$, and $\tilde{\zeta}'^2_{i,n} = \left(\frac{1}{n_i}\sum_{j=1}^{n_i} I(X_{ij} > \tilde{\xi}'_{p,n})L_{ij}^2\right) - \left(\frac{1}{n_i}\sum_{j=1}^{n_i} I(X_{ij} > \tilde{\xi}'_{p,n})L_{ij}\right)^2$.

Glasserman et al. (2000b) also prove that the quantile estimator $\tilde{\xi}'_{p,n}$ obtained using IS+SS satisfies the CLT in (5.10) (but they do not consider the CLT in (5.11) with estimated variance). Also, Glynn (1996) establishes CLTs analogous to (5.6) and (5.10) (but not (5.7) and (5.11) with estimated variance) for the IS-only quantile estimator $\tilde{\xi}_{p,n}$ equal to $\tilde{F}_{n,\mathrm{IS}}^{-1}(p)$ or $\tilde{F}'^{-1}_{n,\mathrm{IS}}(p)$. Both Glasserman et al. (2000b) and Glynn (1996) apply the Berry-Esséen theorem (e.g., p. 33 of Serfling (1980)) in their proofs, and consequently, they require the likelihood ratio L_{ij} to have a finite third moment (under the IS measure). Our proof of (5.6) uses a different approach employing the Bahadur-Ghosh representation, which allows us to relax the moment condition on the likelihood ratio to instead require $E_*[I(X_{ij} \leq \xi_p + \delta)L_{ij}^{2+\varepsilon}] < \infty$ for some $\varepsilon > 0$ and $\delta > 0$. Moreover, under the stronger assumptions discussed earlier, Sun and Hong (2010) establish that the IS-only quantile estimator $\tilde{F}_{n,\mathrm{IS}}^{-1}(p)$ obeys the CLT in (5.6), but they do not consider (5.7) with estimated variance nor IS+SS.

We now explain how to invert the estimated CDF $\tilde{F}_n$ in (5.4). We are given λ_i, n_i, X_{ij} and L_{ij}, and recall the total number of samples across all strata is n. For each $i = 1,\ldots,k$, and $j = 1,\ldots,n_i$, define $A_m = X_{ij}$ and $B_m = L_{ij}\lambda_i/n_i$, where $m = \sum_{\ell=1}^{i-1} n_\ell + j$. Then sort $A_1, A_2, \ldots, A_n$ in ascending order as $A_{(1)} \leq A_{(2)} \leq \cdots \leq A_{(n)}$, and let $B^{(i)}$ correspond to $A_{(i)}$.

For fixed $0 < q < 1$, define the qth quantile estimator $\tilde{\xi}_{q,n}$ to be $\tilde{F}_n^{-1}(q) = A_{(i_q)}$, where i_q is the smallest integer for which $\sum_{m=1}^{i_q} B^{(m)} \geq q$. Similarly, for the estimated CDF $\tilde{F}_n'$ in (5.9), we compute the qth quantile estimator $\tilde{\xi}_{q,n}'$ to be $\tilde{F}_n'^{-1}(q) = A_{(i_q')}$, where i_q' is the smallest integer for which $\sum_{m=i_q'+1}^{n} B^{(m)} \leq 1-q$.

5.1 Proofs

5.1.1 Proof of Theorem 4

Recall the IS+SS estimator $\tilde{F}_n(x)$ of $F(x)$ given in (5.4). Since L_{ij}, $I(X_{ij} \leq x)$, n_i and λ_i are all nonnegative, $\tilde{F}_n(x)$ is monotonically increasing in x for each n, so Assumption A1 is satisfied.

Now we will show that Assumption A2 holds by verifying Conditions C1 and C2 and applying Proposition 1. For all $x \in \Re$, we have

$$E_*[\tilde{F}_n(x)] = E_*\left[\sum_{i=1}^{k} \lambda_i \frac{1}{n_i} \sum_{j=1}^{n_i} I(X_{ij} \leq x) L_{ij}\right] = \sum_{i=1}^{k} \lambda_i \frac{1}{n_i} \sum_{j=1}^{n_i} E_*\left[I(X_{ij} \leq x) L_{ij}\right]$$
$$= \sum_{i=1}^{k} \lambda_i E_*\left[I(X \leq x) L \mid Y \in S_i\right] = F(x) \tag{5.12}$$

by (5.3), so $\tilde{F}_n(x)$ is unbiased for all x. Hence, Condition C1 is satisfied with $r_n(\cdot) = 0$ in this case.

We now show Condition C2 holds. Let $a_n = O(n^{-1/2})$, and define $D_n = \tilde{F}_n(\xi_p + a_n) - \tilde{F}_n(\xi_p)$. Set $\rho_n = \min(\xi_p, \xi_p + a_n)$ and $\rho_n' = \max(\xi_p, \xi_p + a_n)$, so

$$E_*[D_n^2] = E_*\left(\sum_{i=1}^{k} \lambda_i \frac{1}{n_i} \sum_{j=1}^{n_i} I(\rho_n < X_{ij} \leq \rho_n') L_{ij}\right)^2 = A + B, \tag{5.13}$$

where

$$A = E_*\left[\sum_{i=1}^{k} \lambda_i^2 \left(\frac{1}{n_i} \sum_{j=1}^{n_i} I(\rho_n < X_{ij} \leq \rho_n') L_{ij}\right)^2\right]$$

and by the independence of (X_{ij}, Y_{ij}, L_{ij}) and $(X_{i'j'}, Y_{i'j'}, L_{i'j'})$ for $i \neq i'$,

$$\begin{aligned} B &= \sum_{i=1}^{k} \sum_{\substack{i'=1 \\ i' \neq i}}^{k} \lambda_i \frac{1}{n_i} \sum_{j=1}^{n_i} E_* \left[I(\rho_n < X_{ij} \leq \rho_n') L_{ij} \right] \lambda_{i'} \frac{1}{n_{i'}} \sum_{j'=1}^{n_{i'}} E_* \left[I(\rho_n < X_{i'j'} \leq \rho_n') L_{i'j'} \right] \\ &= \sum_{i=1}^{k} \sum_{\substack{i'=1 \\ i' \neq i}}^{k} \lambda_i E_*[I(\rho_n < X \leq \rho_n') L \mid Y \in S_i] \lambda_{i'} E_*[I(\rho_n < X \leq \rho_n') L \mid Y \in S_{i'}]. \end{aligned}$$

We can then express $A = A_1 + A_2$, where

$$A_1 = \sum_{i=1}^{k} \frac{\lambda_i^2}{n_i^2} \sum_{j=1}^{n_i} E_* \left[I(\rho_n < X_{ij} \leq \rho_n') L_{ij}^2 \right] = \sum_{i=1}^{k} \frac{\lambda_i^2}{n_i} E_* \left[I(\rho_n < X_{i1} \leq \rho_n') L_{i1}^2 \right] \tag{5.14}$$

and by the independence of (X_{ij}, Y_{ij}, L_{ij}) and $(X_{ij'}, Y_{ij'}, L_{ij'})$ for $j \neq j'$,

$$\begin{aligned} A_2 &= \sum_{i=1}^{k} \frac{\lambda_i^2}{n_i^2} \sum_{j=1}^{n_i} \sum_{\substack{j'=1 \\ j' \neq j}}^{n_i} E_* \left[I(\rho_n < X_{ij} \leq \rho_n') L_{ij} \right] E_* \left[I(\rho_n < X_{ij'} \leq \rho_n') L_{ij'} \right] \\ &= \sum_{i=1}^{k} \frac{n_i(n_i - 1)}{n_i^2} \lambda_i^2 E_*^2 \left[I(\rho_n < X \leq \rho_n') L \mid Y \in S_i \right]. \end{aligned}$$

We then write $A_2 = A_{21} - A_{22}$, where

$$\begin{aligned} A_{21} &= \sum_{i=1}^{k} \lambda_i^2 E_*^2 \left[I(\rho_n < X \leq \rho_n') L \mid Y \in S_i \right], \\ A_{22} &= \sum_{i=1}^{k} \frac{\lambda_i^2}{n_i} E_*^2 \left[I(\rho_n < X \leq \rho_n') L \mid Y \in S_i \right]. \end{aligned} \tag{5.15}$$

Then, (5.1) and (5.3) imply

$$\begin{aligned} A_{21} + B &= \left(\sum_{i=1}^{k} \lambda_i E_* \left[I(\rho_n < X \leq \rho_n') L \mid Y \in S_i \right] \right)^2 \\ &= E_*^2 \left[I(\rho_n < X \leq \rho_n') L \right] = P^2(\rho_n < X \leq \rho_n'). \end{aligned} \tag{5.16}$$

Hence, substituting (5.14)–(5.16) into (5.13) yields

$$E_*[D_n^2] = [F(\xi_p + a_n) - F(\xi_p)]^2 + A_1 - A_{22}. \tag{5.17}$$

Now we need to check if $s_n(a_n) \equiv nA_1 - nA_{22} \to 0$ as $n \to \infty$, as required by Condition C2. We assumed $\upsilon_i \equiv E_*[I(X_{ij} < \xi_p + \delta)L_{ij}^{2+\varepsilon}] < \infty$ for each stratum i, and let $\tau = \max_{i=1,\dots,k} \upsilon_i^{1/(1+\varepsilon)}$, which is finite since $k < \infty$. Recall $n_i = \gamma_i n$, and note that $I^2(\cdot) = I(\cdot)$. Then applying a change of measure and Hölder's inequality to (5.14) yield

$$\begin{aligned}
A_1 &= \frac{1}{n}\sum_{i=1}^{k} \frac{\lambda_i^2}{\gamma_i} E\left[I^2(\rho_n < X_{i1} \le \rho_n')L_{i1}\right] \\
&\le \frac{1}{n}\sum_{i=1}^{k} \frac{\lambda_i^2}{\gamma_i} E^{\varepsilon/(1+\varepsilon)}\left[I^{(1+\varepsilon)/\varepsilon}(\rho_n < X_{i1} \le \rho_n')\right] E^{1/(1+\varepsilon)}\left[I^{1+\varepsilon}(\rho_n < X_{i1} \le \rho_n')L_{i1}^{1+\varepsilon}\right] \\
&= \frac{1}{n}\sum_{i=1}^{k} \frac{\lambda_i^2}{\gamma_i} P^{\varepsilon/(1+\varepsilon)}(\rho_n < X \le \rho_n' \mid Y \in S_i) E_*^{1/(1+\varepsilon)}\left[I(\rho_n < X_{i1} \le \rho_n')L_{i1}^{2+\varepsilon}\right] \\
&\le \frac{\tau}{n}\sum_{i=1}^{k} \frac{\lambda_i^{(2+\varepsilon)/(1+\varepsilon)}}{\gamma_i} P^{\varepsilon/(1+\varepsilon)}(\rho_n < X \le \rho_n', Y \in S_i)
\end{aligned}$$

for n sufficiently large since then $I(\rho_n < X_{ij} \le \rho_n') \le I(X_{ij} \le \xi_p + \delta)$ because $\rho_n' = \max(\xi_p, \xi_p + a_n)$ and $a_n = O(n^{-1/2})$. Similarly, we rewrite (5.15) as

$$A_{22} = \frac{1}{n}\sum_{i=1}^{k} \frac{1}{\gamma_i} P^2(\rho_n < X \le \rho_n', Y \in S_i),$$

which leads to

$$\begin{aligned}
|s_n(a_n)| &\le |nA_1| + |nA_{22}| \\
&\le \sum_{i=1}^{k} \frac{1}{\gamma_i} P^{\varepsilon/(1+\varepsilon)}(\rho_n < X \le \rho_n', Y \in S_i) \left[\lambda_i^{\left(\frac{2+\varepsilon}{1+\varepsilon}\right)}\tau + P^{\left(\frac{2+\varepsilon}{1+\varepsilon}\right)}(\rho_n < X \le \rho_n', Y \in S_i)\right].
\end{aligned}$$

Because the differentiability of F at ξ_p implies F is continuous at ξ_p, it follows that $P(\rho_n < X \le \rho_n', Y \in S_i) \le P(\rho_n < X \le \rho_n') \to 0$ since $\rho_n \to \xi_p$ and $\rho_n' \to \xi_p$ as $n \to \infty$. Thus,

$s_n(a_n) \to 0$ as $n \to \infty$ because $\gamma_i > 0$, $\lambda_i \leq 1$ and $\tau < \infty$, and then Condition C2 follows from (5.17). Consequently, Assumption A2 holds by Proposition 1.

Lastly, we need to show Assumption A3 holds. Recall that $n_i = n\gamma_i$, so

$$\begin{aligned}\sqrt{n}\left[\tilde{F}_n(\xi_p) - F(\xi_p)\right] &= \sqrt{n}\left\{\sum_{i=1}^{k}\frac{\lambda_i}{n_i}\sum_{j=1}^{n_i} I(X_{ij} \leq \xi_p)L_{ij} - \sum_{i=1}^{k}\lambda_i E_*\left[I(X_{i1} \leq \xi_p)L_{i1}\right]\right\} \\ &= \sum_{i=1}^{k}\frac{\lambda_i}{\sqrt{\gamma_i}}G_{i,n}, \qquad (5.18)\end{aligned}$$

where $G_{i,n} = \sqrt{n_i}\left[\frac{1}{n_i}\sum_{j=1}^{n_i} I(X_{ij} \leq \xi_p)L_{ij} - P(X_{i1} \leq \xi_p)\right]$. Also, $I(X_{ij} \leq \xi_p)L_{ij}$, $j = 1, 2, \ldots, n_i$, are i.i.d. with finite variance under P_* since $\upsilon_i < \infty$. Hence, $G_{i,n} \xrightarrow{L} N(0, \zeta_i^2) \equiv N_i$ as $n \to \infty$ for each i, where $\zeta_i^2 = \mathrm{Var}_*[I(X_{ij} \leq \xi_p)L_{ij}]$, which equals (5.5) from a change of measure.

The samples across strata are independent, so the independence of $G_{i,n}$, $i = 1, \ldots, k$, implies the independence of N_i, $i = 1, \ldots, k$. It follows that $(G_{i,n}, i = 1, \ldots, k) \xrightarrow{L} (N_i, i = 1, \ldots, k)$ as $n \to \infty$ by Theorem 11.4.4 of Whitt (2002). Then apply the continuous mapping theorem to (5.18) and obtain $\sqrt{n}\left(\tilde{F}_n(\xi_p) - F(\xi_p)\right) \xrightarrow{L} N\left(0, \sum_{i=1}^{k}\lambda_i^2\zeta_i^2/\gamma_i\right)$, which completes the proof.

5.1.2 Proof of Theorem 5

By applying Theorem 2, we see that (5.6) follows from Theorem 4. To establish (5.7), we will show

$$\tilde{\psi}_{p,n} \xrightarrow{P} \psi_p \qquad (5.19)$$

as $n \to \infty$ so that we can apply (4.10) from Theorem 3. Recall that $\psi_p = \sum_{i=1}^{k}\lambda_i^2\zeta_i^2/\gamma_i$ with ζ_i^2 defined in (5.5). Also, we defined $\tilde{\zeta}_{i,n}^2$ in (5.8) as an estimator of ζ_i^2. Now define $Z_{i,n}(x) = \frac{1}{n_i}\sum_{j=1}^{n_i} I(X_{ij} \leq x)L_{ij}^2$ and $z_i(x) = E_*[I(X_{ij} \leq x)L_{ij}^2]$, so $Z_{i,n}(\tilde{\xi}_{p,n})$ is the first term in $\tilde{\zeta}_{i,n}^2$ and $z_i(\xi_p)$ is the first term in ζ_i^2. To prove (5.19), we first show that for each $i = 1, \ldots, k$,

$$Z_{i,n}(\tilde{\xi}_{p,n}) \xrightarrow{P} z_i(\xi_p) \qquad (5.20)$$

as $n \to \infty$ under the IS measure P_*. Fix an arbitrary $a_0 > 0$, and establishing (5.20) requires proving that $P_*\left\{|Z_{i,n}(\tilde{\xi}_{p,n}) - z_i(\xi_p)| > a_0\right\} \to 0$ as $n \to \infty$. Because we assumed that $\upsilon_i \equiv E_*[I(X_{ij} \le \xi_p + \delta)L_{ij}^{2+\varepsilon}] = E[I(X_{ij} \le \xi_p + \delta)L_{ij}^{1+\varepsilon}] < \infty$ for some positive ε and δ and also that F is differentiable at ξ_p, there exists $0 < \delta' < \delta$ such that

$$P\{\xi_p < X \le \xi_p + \delta'\} \le \lambda_i \frac{(a_0/2)^{(1+\varepsilon)/\varepsilon}}{\upsilon^{1/\varepsilon}}, \tag{5.21}$$

where we recall $\lambda_i = P_*[Y \in S_i] > 0$. Then

$$\begin{aligned} P_*\left\{|Z_{i,n}(\tilde{\xi}_{p,n}) - z_i(\xi_p)| > a_0\right\} &= P_*\left\{|Z_{i,n}(\tilde{\xi}_{p,n}) - z_i(\xi_p)| > a_0, |\tilde{\xi}_{p,n} - \xi_p| \le \delta'\right\} \\ &\quad + P_*\left\{|Z_{i,n}(\tilde{\xi}_{p,n}) - z_i(\xi_p)| > a_0, |\tilde{\xi}_{p,n} - \xi_p| > \delta'\right\} \\ &\equiv q_{1,n} + q_{2,n}, \end{aligned} \tag{5.22}$$

so we want to show $q_{1,n} \to 0$ and $q_{2,n} \to 0$ as $n \to \infty$ to establish (5.20). Since $\tilde{\xi}_{p,n} \xrightarrow{P} \xi_p$ by (4.3),

$$q_{2,n} \le P_*\left\{|\tilde{\xi}_{p,n} - \xi_p| > \delta'\right\} \to 0 \tag{5.23}$$

as $n \to \infty$.

We now handle $q_{1,n}$ in (5.22). Because $Z_{i,n}(x)$ is monotonically increasing in x, we have that $|\tilde{\xi}_{p,n} - \xi_p| \le \delta'$ implies $\max(|Z_{i,n}(\xi_p + \delta') - z_i(\xi_p)|, |Z_{i,n}(\xi_p - \delta') - z_i(\xi_p)|) \ge |Z_{i,n}(\tilde{\xi}_{p,n}) - z_i(\xi_p)|$. Hence,

$$\begin{aligned} q_{1,n} &\le P_*\left\{\left|Z_{i,n}(\xi_p + \delta') - z_i(\xi_p)\right| > a_0, |\tilde{\xi}_{p,n} - \xi| \le \delta'\right\} \\ &\quad + P_*\left\{\left|Z_{i,n}(\xi_p - \delta') - z_i(\xi_p)\right| > a_0, |\tilde{\xi}_{p,n} - \xi| \le \delta'\right\} \\ &\equiv r_{1,n} + r_{2,n}. \end{aligned} \tag{5.24}$$

Observe that

$$
\begin{aligned}
r_{1,n} &\le P_*\left\{\left|Z_{i,n}(\xi_p+\delta') - z_i(\xi_p)\right| > a_0\right\} \\
&\le P_*\left\{\left|Z_{i,n}(\xi_p+\delta') - z_i(\xi_p+\delta')\right| + \left|z_i(\xi_p+\delta') - z_i(\xi_p)\right| > a_0\right\}
\end{aligned}
\tag{5.25}
$$

by the triangle inequality. Also, a change of measure and the fact $I^2(\cdot) = I(\cdot)$ yield

$$
\begin{aligned}
|z_i(\xi_p+\delta') - z_i(\xi_p)| &= |E[I(X_{ij} \le \xi_p+\delta')L_{ij}] - E[I(X_{ij} \le \xi_p)L_{ij}]| \\
&= E[I^2(\xi_p < X_{ij} \le \xi_p+\delta')L_{ij}] \\
&\le P^{\varepsilon/(1+\varepsilon)}\{\xi_p < X_{ij} \le \xi_p+\delta'\}\, E^{1/(1+\varepsilon)}[I^{1+\varepsilon}(\xi_p < X_{ij} \le \xi_p+\delta')L_{ij}^{1+\varepsilon}] \\
&\le \left(\frac{P\{\xi_p < X \le \xi_p+\delta', Y \in S_i\}}{\lambda_i}\right)^{\varepsilon/(1+\varepsilon)} \upsilon^{1/(1+\varepsilon)} \\
&\le \left(\frac{P\{\xi_p < X \le \xi_p+\delta'\}}{\lambda_i}\right)^{\varepsilon/(1+\varepsilon)} \upsilon^{1/(1+\varepsilon)} \\
&\le a_0/2,
\end{aligned}
\tag{5.26}
$$

where the third step follows from Hölder's inequality, the fourth step holds because $I(\xi_p < X_{ij} \le \xi_p+\delta') \le I(X_{ij} \le \xi_p+\delta)$ as $\delta' < \delta$, and (5.21) implies the last step. Thus, using (5.26) in (5.25) gives

$$
r_{1,n} \le P_*\left\{\left|Z_{i,n}(\xi_p+\delta') - z_i(\xi_p+\delta')\right| > a_0/2\right\}. \tag{5.27}
$$

In the definition of $Z_{i,n}(\xi_p+\delta')$, the summands $I(X_{ij} \le \xi_p+\delta')L_{ij}^2$, $j = 1,2,\ldots,n_i$, are i.i.d., and each summand has mean

$$
\begin{aligned}
z_i(\xi_p+\delta') = E_*[I(X_{ij} \le \xi_p+\delta')L_{ij}^2] &= E[I(X_{ij} \le \xi_p+\delta')L_{ij}] \\
&\le P^{\varepsilon/(1+\varepsilon)}\{X_{ij} \le \xi_p+\delta'\}\, \upsilon^{1/(1+\varepsilon)} < \infty
\end{aligned}
$$

by Hölder's inequality. Thus, the weak law of large numbers implies the RHS of (5.27) converges to 0 as $n \to \infty$, so $r_{1,n} \to 0$ as $n \to \infty$. We can similarly prove that $r_{2,n}$ in (5.24) satisfies $r_{2,n} \to 0$, so $q_{1,n} \to 0$ by (5.24). Using this together with (5.22) and (5.23) then establishes (5.20). For the second term in (5.8), similar arguments show that $\frac{1}{n_i}\sum_{j=1}^{n_i} I(X_{ij} \leq \tilde{\xi}_{p,n})L_{ij} \xrightarrow{P} E_*[I(X_{ij} \leq \xi_p)L_{ij}] = P\{X \leq \xi_p \mid Y \in S_i\}$ as $n \to \infty$, so (5.19) holds by the continuous-mapping theorem and Slutsky's theorem, completing the proof.

CHAPTER 6

QUANTILE ESTIMATION USING AV

In the case of estimating the mean of a random output X having CDF F, the basic idea of AV is to generate two copies X and X' of the output having CDF F in such a way that X and X' are negatively correlated, and we average the two outputs. Since $\mathrm{Var}((X+X')/2) = [\mathrm{Var}(X)+\mathrm{Cov}(X,X')]/2 \leq \mathrm{Var}(X)/2$ when $\mathrm{Cov}(X,X') \leq 0$, AV reduces variance compared to when X and X' are independent. There are various ways in which we can generate negatively correlated X and X' with the same marginal distribution F. For example, suppose that the output X can be expressed as $X = h(U_1,\ldots,U_d)$ for some function h, where $U_1,\ldots,U_d$ are i.i.d. uniform random variables on the unit interval. Then $X' = h(1-U_1,\ldots,1-U_d)$ has the same distribution as X since $1-U_i$ is also uniform on $[0,1]$. If h is monotonic in each of its arguments, then X and X' are negatively correlated; e.g., see Section 8.1 of Ross (1997).

In general, we implement AV to estimate a quantile of output X by generating (X_i,X_i'), $i = 1,2,\ldots,n$, as i.i.d. antithetic pairs, where X_i and X_i' each have marginal distribution F and X_i and X_i' are negatively correlated. Then the AV estimator of the CDF F is

$$\tilde{F}_n(x) = \frac{1}{2}\left[\frac{1}{n}\sum_{i=1}^{n} I(X_i \leq x) + \frac{1}{n}\sum_{i=1}^{n} I(X_i' \leq x)\right] = \frac{1}{2}\left[F_n(x)+F_n'(x)\right], \tag{6.1}$$

where F_n is defined in (3.1) and $F_n' = (1/n)\sum_{i=1}^{n} I(X_i' \leq x)$. Inverting $\tilde{F}_n$ yields the CV quantile estimator, which the following shows satisfies a Bahadur-Ghosh representation.

Theorem 8 *Suppose $f(\xi_p) > 0$, and let $\tilde{F}_n$ be the AV estimator of F defined in (6.1). Then $\tilde{F}_n$ satisfies Assumptions A1–A3, where*

$$\psi_p^2 = \frac{1}{2}\left[p(1-2p) + P\{X \leq \xi_p, X' \leq \xi_p\}\right] \tag{6.2}$$

is the variance constant in Assumption A3 and (X,X') is an antithetic pair. Thus, Theorem 1 *implies* $\tilde{\xi}_{p_n,n} = \tilde{F}_n^{-1}(p_n)$ *with* $p_n - p = O(n^{-1/2})$ *satisfies the Bahadur-Ghosh representation in (4.1) and (4.2).*

The next result shows that the AV quantile estimator satisfies a CLT.

Theorem 9 *If* $f(\xi_p) > 0$*, then* $\tilde{\xi}_{p,n} = \tilde{F}_n^{-1}(p)$ *with* $\tilde{F}_n$ *defined in (6.1) satisfies*

$$\frac{\sqrt{n}}{\kappa_p}(\tilde{\xi}_{p,n} - \xi_p) \xrightarrow{L} N(0,1), \tag{6.3}$$

$$\frac{\sqrt{n}}{\tilde{\kappa}_{p,n}}(\tilde{\xi}_{p,n} - \xi_p) \xrightarrow{L} N(0,1), \tag{6.4}$$

as $n \to \infty$*, where* $\kappa_p = \psi_p \phi_p$*,* $\phi_p = 1/f(\xi_p)$*,* ψ_p *is defined in (6.2),* $\tilde{\kappa}_{p,n} = \tilde{\psi}_{p,n}\bar{\phi}_{p,n,i}(c_1,\ldots,c_r)$*,* $\bar{\phi}_{p,n,i}(c_1,\ldots,c_r)$ *is from (4.7) with nonzero constants* $c_1,\ldots,c_r$ *and* $w_1,\ldots,w_r$ *satisfying* $\sum_{j=1}^r w_j = 1$ *for* $i = 1$ *or 2, and*

$$\tilde{\psi}_{p,n}^2 = \frac{1}{2}\left[p(1-2p) + \frac{1}{n}\sum_{i=1}^n I(X_i \le \tilde{\xi}_{p,n}, X_i' \le \tilde{\xi}_{p,n})\right]. \tag{6.5}$$

We now describe how to invert the AV CDF estimator $\tilde{F}_n$ in (6.1). For each $i = 1,\ldots,n$, define $A_{2i-1} = X_i$ and $A_{2i} = X_i'$. Then sort $A_1, A_2, \ldots, A_{2n}$ in ascending order as $A_{(1)} \le A_{(2)} \le \cdots \le A_{(2n)}$. For fixed $0 < q < 1$, define the qth quantile estimator $\tilde{\xi}_{q,n}$ to be $\tilde{F}_n^{-1}(q) = A_{(\lceil 2nq \rceil)}$, where $\lceil \cdot \rceil$ denotes the round-up function.

6.1 Proofs

6.1.1 Proof of Theorem 8

Recall the AV estimator $\tilde{F}_n(x)$ of $F(x)$ given in (6.1). Since both $I(X_i \le x)$ and $I(X_i' \le x)$ are nonnegative, $\tilde{F}_n(x)$ is monotonically increasing in x for each n, so Assumption A1 holds.

Next we will show Assumption A2 is satisfied by verifying Conditions C1 and C2 and applying Proposition 1. Observe that $F_n(x)$ is a special case of the IS+SS estimator with a single stratum (i.e., no SS) and IS measure $P_* = P$ (so the likelihood ratio $L \equiv 1$, which means no IS). Also, $F'_n(x) \stackrel{D}{=} F_n(x)$ since each $X'_i \stackrel{D}{=} X_i$, where $\stackrel{D}{=}$ denotes equality in distribution, so

$$E(\tilde{F}_n(x)) = \frac{1}{2}[E(F_n(x)) + E(F'_n(x))] = F(x)$$

by (5.12). Therefore, the AV CDF estimator is unbiased, which implies Condition C1.

We now show Condition C2 holds. Set $a_n = O(n^{-1/2})$, and (6.1) implies

$$E\left[\tilde{F}_n(\xi_p + a_n) - \tilde{F}_n(\xi_p)\right]^2 = \frac{1}{4}(A_1 + A_2 + A_3), \tag{6.6}$$

where

$$\begin{aligned}
A_1 &= E\left[F_n(\xi_p + a_n) - F_n(\xi_p)\right]^2, \\
A_2 &= E\left[F'_n(\xi_p + a_n) - F'_n(\xi_p)\right]^2, \\
A_3 &= 2E\left(\left[F_n(\xi_p + a_n) - F_n(\xi_p)\right]\left[F'_n(\xi_p + a_n) - F'_n(\xi_p)\right]\right).
\end{aligned}$$

As a special case of the IS+SS CDF estimator, $F_n(x)$ also satisfies Condition C2, as shown in the proof of Theorem 4. Therefore, since $F'_n(x) \stackrel{D}{=} F_n(x)$ for all x, we have

$$A_1 = A_2 = d_n^2 + \frac{s_n(a_n)}{n} \tag{6.7}$$

with $s_n(a_n) \to 0$ as $n \to \infty$, where $d_n = F(\xi_p + a_n) - F(\xi_p)$. To handle A_3, let $\rho_n = \min(\xi_p, \xi_p + a_n)$ and $\rho'_n = \max(\xi_p, \xi_p + a_n)$. Then

$$A_3 = \frac{2}{n^2} E\left[\left(\sum_{i=1}^{n} I(\rho_n < X_i \le \rho'_n)\right)\left(\sum_{j=1}^{n} I(\rho_n < X'_j \le \rho'_n)\right)\right] = \frac{2}{n^2}(B_1 + B_2), \tag{6.8}$$

where

$$B_1 = \sum_{i=1}^{n} E\left[I(\rho_n < X_i \leq \rho_n')I(\rho_n < X_i' \leq \rho_n')\right]$$
$$= nP(\rho_n < X \leq \rho_n', \rho_n < X' \leq \rho_n') \tag{6.9}$$

with (X,X') an antithetic pair and

$$B_2 = E\left[\sum_{i \neq j} I(\rho_n < X_i \leq \rho_n')I(\rho_n < X_j' \leq \rho_n')\right] = n(n-1)d_n^2 \tag{6.10}$$

since X_i and X_j' are independent for $i \neq j$. Then, (6.6)–(6.10) imply

$$E\left[\tilde{F}_n(\xi_p + a_n) - \tilde{F}_n(\xi_p)\right]^2 = d_n^2 + \frac{1}{2n}\left[s_n(a_n) + P(\rho_n < X \leq \rho_n', \rho_n < X' \leq \rho_n') - d_n^2\right]$$
$$\equiv d_n^2 + \frac{1}{2n}t_n.$$

Since $P(\rho_n < X \leq \rho_n', \rho_n < X' \leq \rho_n') \leq P(\rho_n < X \leq \rho_n') = |d_n|$, the triangle inequality yields

$$|t_n| \leq |s_n(a_n)| + P(\rho_n < X \leq \rho_n') + d_n^2 = |s_n(a_n)| + |d_n| + d_n^2.$$

Because the differentiability of F at ξ_p implies F is continuous at ξ_p, it follows that $d_n \to 0$ as $n \to \infty$ since $\rho_n \to \xi_p$ and $\rho_n' \to \xi_p$ as $n \to \infty$. Thus, $t_n \to 0$ as $n \to \infty$ since $s_n(a_n) \to 0$, which shows Condition C2 is satisfied. Hence, Assumption A2 follows by Proposition 1.

Lastly, we need to show Assumption A3 holds. We can rewrite (6.1) as $\tilde{F}_n(\xi_p) = (1/n)\sum_{i=1}^{n} M_i$, where $M_i = [I(X_i \leq \xi_p) + I(X_i' \leq \xi_p)]/2$, $i = 1,2,\ldots,n$, are i.i.d. since (X_i, X_i'), $i = 1,2,\ldots,n$, are i.i.d. Since $E[M_i] = F(\xi_p)$, the CLT in Assumption A3 holds, and the variance ψ_p^2 in the CLT is given by

$$\begin{aligned}\mathrm{Var}(M_i) &= \frac{1}{4}\left\{\mathrm{Var}[I(X_i \le \xi_p)] + \mathrm{Var}[I(X_i' \le \xi_p)] + 2\mathrm{Cov}[I(X_i \le \xi_p), I(X_i' \le \xi_p)]\right\} \\ &= \frac{1}{2}\left[p(1-2p) + P(X \le \xi_p, X' \le \xi_p)\right],\end{aligned}$$

which completes the proof.

6.1.2 Proof of Theorem 9

By applying Theorem 2, we see that (6.3) follows from Theorem 8. To establish (6.4), we will show that $\tilde{\psi}_{p,n}$ in (6.5) satisfies

$$\tilde{\psi}_{p,n} \xrightarrow{P} \psi_p \tag{6.11}$$

as $n \to \infty$ so that we can employ (4.10) from Theorem 3. Now we define $Z_n(x) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \le x, X_i' \le x)$ and $z(x) = E\left[I(X_i \le x, X_i' \le x)\right]$, so $Z_n(\tilde{\xi}_{p,n})$ is the only random term in $\tilde{\psi}_{p,n}^2$. To prove (6.11), we just need to show that $Z_n(\tilde{\xi}_{p,n}) \xrightarrow{P} z(\xi_p)$ as $n \to \infty$. But we can establish this by applying arguments similar to those used to show (5.20), so the proof is complete.

CHAPTER 7

QUANTILE ESTIMATION USING CV

In many simulations, one often knows the mean of an auxiliary random variable that is generated in the process of generating the output random variable. For example, in a simulation of a queueing system, one knows the distributions (and typically also the means) of the interarrival times and service times. The method of control variates (CV) reduces variance by exploiting this knowledge.

Suppose that (X,C) is a correlated pair of random variables, where we are interested in estimating the pth quantile of X. We assume that C has known mean ν and finite variance, and we will use C as a *control variate*. To avoid trivialities, we assume that $\mathrm{Var}[C] > 0$. Since $X' \equiv I(X \le x) - \beta(C - \nu)$ has mean $F(x)$ for any constant β, we can average i.i.d. samples of X' to obtain an unbiased estimator of $F(x)$. Specifically, we generate i.i.d. samples (X_i, C_i), $i = 1, 2, \ldots, n$, of the pair (X,C). Letting β be any constant, we can define an unbiased estimator of the CDF F of X as

$$\tilde{F}'_{n,\beta}(x) = \frac{1}{n}\sum_{i=1}^{n}\left[I(X_i \le x) - \beta(C_i - \nu)\right] \tag{7.1}$$

$$= F_n(x) - \beta(\bar{C}_n - \nu), \tag{7.2}$$

where F_n is the empirical CDF defined in (3.1) and $\bar{C}_n = (1/n)\sum_{i=1}^{n} C_i$.

Clearly, the variance of $\tilde{F}'_{n,\beta}(x)$ depends on the value of β. It can be shown (e.g., p. 186 of Glasserman (2004)) that the choice of β that minimizes the variance is

$$\beta_*(x) = \frac{\mathrm{Cov}[I(X \le x), C]}{\mathrm{Var}[C]},$$

which depends on x. However, one typically does not know the value of $\mathrm{Cov}[I(X \le x), C]$, so it must be estimated. We thus estimate $\beta_*(x)$ via

$$\hat{\beta}_n(x) = \frac{[(1/n)\sum_{i=1}^n I(X_i \leq x)C_i] - F_n(x)\bar{C}_n}{(1/n)\sum_{j=1}^n (C_j - \bar{C}_n)^2}. \tag{7.3}$$

Replacing β in (7.2) with $\hat{\beta}_n(x)$ gives us the CV estimator of the CDF F as

$$\tilde{F}_n(x) = F_n(x) - \hat{\beta}_n(x)(\bar{C}_n - \nu), \tag{7.4}$$

which is typically no longer unbiased because of the correlation of $\hat{\beta}_n(x)$ and $\bar{C}_n$. We obtain the CV quantile estimator by inverting $\tilde{F}_n$ in (7.4), and the following result shows the quantile estimator satisfies a Bahadur-Ghosh representation.

Theorem 10 *Suppose $f(\xi_p) > 0$, and let C be a control variate with $0 < Var[C] < \infty$. Let $\tilde{F}_n$ be the CV estimator of F defined in (7.4). Then $\tilde{F}_n$ satisfies Assumptions A1–A3, where*

$$\psi_p^2 = p(1-p) + \beta_*^2(\xi_p)Var[C] - 2\beta_*(\xi_p)Cov[I(X \leq \xi_p), C] \tag{7.5}$$

is the variance constant in Assumption A3. Thus, Theorem 1 implies $\tilde{\xi}_{p_n,n} = \tilde{F}_n^{-1}(p_n)$ with $p_n - p = O(n^{-1/2})$ satisfies the Bahadur-Ghosh representation in (4.1) and (4.2).

The CV quantile estimator satisfies a CLT, as shown next. We omit the proof since it can be established by applying arguments similar to those employed in the proof of Theorem 5.

Theorem 11 *Under the assumptions of Theorem 10, $\tilde{\xi}_{p,n} = \tilde{F}_n^{-1}(p)$ satisfies*

$$\frac{\sqrt{n}}{\kappa_p}(\tilde{\xi}_{p,n} - \xi_p) \xrightarrow{L} N(0,1),$$
$$\frac{\sqrt{n}}{\tilde{\kappa}_{p,n}}(\tilde{\xi}_{p,n} - \xi_p) \xrightarrow{L} N(0,1),$$

as $n \to \infty$, where $\kappa_p = \psi_p \phi_p$, $\phi_p = 1/f(\xi_p)$, ψ_p is defined in (7.5), $\tilde{\kappa}_{p,n} = \tilde{\psi}_{p,n}\bar{\phi}_{p,n,i}(c_1, \ldots, c_r)$,

$\bar{\phi}_{p,n,i}(c_1,\ldots,c_r)$ *is from (4.7) with nonzero constants* $c_1,\ldots,c_r$ *and* $w_1,\ldots,w_r$ *satisfying* $\sum_{j=1}^{r} w_j = 1$ *for* $i = 1$ *or* 2*, and*

$$\tilde{\psi}_{p,n}^2 = p(1-p) + \frac{\hat{\beta}_n^2(\tilde{\xi}_{p,n})}{n}\sum_{i=1}^{n}(C_i - \nu)^2 - 2\hat{\beta}_n(\tilde{\xi}_{p,n})\left(\frac{1}{n}\sum_{j=1}^{n} I(X_j \le \tilde{\xi}_{p,n})C_j - p\nu\right).$$

Inverting $\tilde{F}_n$ in (7.4) initially appears complicated by the fact that $\hat{\beta}_n(x)$ depends on x. However, Hesterberg and Nelson (1998) show that $\tilde{F}_n(x)$ can be rewritten as

$$\tilde{F}_n(x) = \sum_{i=1}^{n} T_i I(X_i \le x), \tag{7.6}$$

where

$$T_i = \frac{1}{n} + \frac{(\bar{C}_n - C_i)(\bar{C}_n - \nu)}{\sum_{j=1}^{n}(C_j - \bar{C}_n)^2}, \tag{7.7}$$

which does not depend on x. If $T_i \ge 0$ for each i, then it is clear from (7.6) that $\tilde{F}_n(x)$ is monotonically increasing in x. (It is possible for T_i to be negative, but Hesterberg and Nelson (1998) note that is unlikely, in a sense they make precise.) The advantage of the representation of $\tilde{F}_n$ in (7.6) is that it allows evaluating $\tilde{F}_n(x)$ at different values of x without needing to recompute $\hat{\beta}_n(x)$ in (7.3) each time. Also, we can invert $\tilde{F}_n$ as follows. We first sort $X_1, X_2, \ldots, X_n$ in ascending order as $X_{(1)} \le X_{(2)} \le \cdots \le X_{(n)}$, and let $T^{(i)}$ correspond to $X_{(i)}$. Then for $0 < q < 1$, we can compute the qth quantile estimator $\tilde{\xi}_{q,n}$ as $\tilde{F}_n^{-1}(q) = X_{(i_q)}$, where $i_q = \min\left\{j : \sum_{i=1}^{j} T^{(i)} \ge q\right\}$.

We now discuss a particular choice for a control variate C. Suppose that Y is a random variable correlated with the output X, and let G be the marginal CDF of Y. We can then define $C = I(Y \le y)$ for some constant y when $G(y)$ is known with $0 < G(y) < 1$. Thus, we collect i.i.d. pairs (X_i, Y_i), $i = 1, \ldots, n$, and define $C_i = I(Y_i \le y)$. It is straightforward to show in this case that $T_i \ge 0$ always holds, ensuring that $\tilde{F}_n(x)$ is monotonically increasing in x for each n. Moreover, Hesterberg and Nelson (1998) note that T_i in (7.7) becomes $T_i = P(Y \le y)/\sum_{i=1}^{n} I(Y_i \le y)$ if $Y_i \le y$, and $T_i = P(Y > y)/\sum_{i=1}^{n} I(Y_i > y)$ if $Y_i > y$.

When estimating the pth quantile $\xi_p = F^{-1}(p)$ of X, a natural choice for y is $y = G^{-1}(p)$ (assuming that this is known), but this is not required.

We can also apply CV with multiple controls $C^{(1)}, C^{(2)}, \ldots, C^{(m)}$, where each $C^{(j)}$ has known mean; see Hesterberg and Nelson (1998) for details. One possible choice is to specify constants $y_1, \ldots, y_m$ for which each $G(y_j)$ is known and set $C^{(j)} = I(Y \le y_j)$.

7.1 Proofs

7.1.1 Proof of Theorem 10

The alternative representation of $\tilde{F}_n$ in (7.6) shows that $\tilde{F}_n(x)$ is monotonically increasing in x when the weights T_i in (7.7) are nonnegative. As noted by Hesterberg and Nelson (1998), the probability of any T_i being negative is $o(n^{-1})$ since we assumed $\mathrm{Var}[C] < \infty$, so Assumption A1 holds.

To establish Assumption A2, let $a_n = O(n^{-1/2})$, and define

$$W_n = \frac{\sqrt{n}}{f(\xi_p)}[p - \tilde{F}_n(\xi_p)],$$
$$Z_n = \frac{\sqrt{n}}{f(\xi_p)}\left[F(\xi_p + a_n) - \tilde{F}_n(\xi_p + a_n)\right],$$

so we need to verify $Z_n - W_n \xrightarrow{P} 0$ as $n \to \infty$. Recalling (3.1) and (7.4), we can write

$$\begin{aligned} Z_n - W_n &= \frac{\sqrt{n}}{f(\xi_p)}[(F(\xi_p + a_n) - F(\xi_p)) - (F_n(\xi_p + a_n) - F_n(\xi_p))] \\ &+ \frac{\sqrt{n}}{f(\xi_p)}[\bar{C}_n - \nu][\hat{\beta}_n(\xi_p + a_n) - \hat{\beta}_n(\xi_p)] \\ &\equiv A_n + B_n. \end{aligned} \tag{7.8}$$

Since F_n is a special case of the IS+SS CDF estimator in (5.4) with $P_* = P$ (so the likelihood ratio $L \equiv 1$, which means no IS) and only $k = 1$ stratum (i.e., no SS), Theorem 4 shows that F_n satisfies Assumption A2, which is exactly that $A_n \xrightarrow{P} 0$ as $n \to \infty$. By Slutsky's theorem, it then suffices to show that $B_n \xrightarrow{P} 0$ as $n \to \infty$ to verify that Assumption A2 holds for CV.

Let $\sigma_C^2 = \mathrm{Var}[C]$. Since we assumed $0 < \sigma_C^2 < \infty$, the CLT then implies

$$\sqrt{n}(\bar{C}_n - \nu) \xrightarrow{L} N(0, \sigma_C^2) \tag{7.9}$$

as $n \to \infty$. Thus, if we prove that $Q_n \equiv \hat{\beta}_n(\xi_p + a_n) - \hat{\beta}_n(\xi_p) \xrightarrow{P} 0$ as $n \to \infty$, then $B_n \xrightarrow{P} 0$ by Slutsky's theorem since $f(\xi_p) > 0$ by assumption. Note that (7.3) implies

$$Q_n = \frac{\frac{1}{n}\sum_{i=1}^{n}[I(X_i \le \xi_p + a_n) - I(X_i \le \xi_p)](C_i - \bar{C}_n)}{\frac{1}{n}\sum_{k=1}^{n}(C_k - \bar{C}_n)^2} \equiv \frac{N_n}{D_n},$$

and $D_n \xrightarrow{P} \sigma_C^2 > 0$ as $n \to \infty$; e.g., see p. 69 of Serfling (1980). Thus, to establish that $Q_n \xrightarrow{P} 0$, it is sufficient to prove that $N_n \xrightarrow{P} 0$ as $n \to \infty$ by Slutsky's theorem. We accomplish this by next verifying that $E(|N_n|) \to 0$ as $n \to \infty$ and applying Theorem 1.3.2 of Serfling (1980).

Let $\rho_n = \min(\xi_p, \xi_p + a_n)$ and $\rho_n' = \max(\xi_p, \xi_p + a_n)$, so the triangle inequality yields

$$\begin{aligned} E(|N_n|) &= \frac{1}{n} E\left(\left|\sum_{i=1}^{n} I(\rho_n < X_i \le \rho_n')(C_i - \bar{C}_n)\right|\right) \\ &\le \frac{1}{n}\sum_{i=1}^{n} E\left[I(\rho_n < X_i \le \rho_n')\left|C_i - \bar{C}_n\right|\right] \\ &\le \frac{1}{n}\sum_{i=1}^{n} P^{1/2}(\rho_n < X_i \le \rho_n') E^{1/2}[(C_i - \bar{C}_n)^2] \end{aligned}$$

by the Cauchy-Schwarz inequality. Using the facts that the C_i, $i = 1, \ldots, n$, are i.i.d. and that $E[C^2] = \sigma_C^2 + \nu^2$, we can show that $E[(C_i - \bar{C}_n)^2] = (n-1)\sigma_C^2/n < \sigma_C^2$, so $E(|N_n|) < P^{1/2}(\rho_n < X_i \le \rho_n')\sigma_C$. Since $\rho_n \to \xi_p$ and $\rho_n' \to \xi_p$ as $n \to \infty$, we have that $P(\rho_n < X \le \rho_n') \to 0$ as $n \to \infty$ because F is differentiable at ξ_p. Hence, $E(|N_n|) \to 0$ as $n \to \infty$ since $\sigma_C < \infty$. Consequently, $B_n \xrightarrow{P} 0$ as $n \to \infty$, showing that Assumption A2 is satisfied by (7.8).

To prove Assumption A3 holds for CV, note that $\hat{\beta}_n(\xi_p) \xrightarrow{P} \beta_*(\xi_p)$ as $n \to \infty$ by the weak law of large numbers and Slutsky's theorem. Thus,

$$\sqrt{n}[F_n(\xi_p) - \beta_*(\xi_p)(\bar{C}_n - \nu)] - \sqrt{n}[F_n(\xi_p) - \hat{\beta}_n(\xi_p)(\bar{C}_n - \nu)]$$
$$= [\hat{\beta}_n(\xi_p) - \beta_*(\xi_p)]\sqrt{n}(\bar{C}_n - \nu) \xrightarrow{L} 0 \cdot N(0, \sigma_C^2) = 0$$

as $n \to \infty$ by (7.9) and Slutsky's theorem, so $\sqrt{n}[F_n(\xi_p) - \beta_*(\xi_p)(\bar{C}_n - \nu)]$ and $\sqrt{n}[F_n(\xi_p) - \hat{\beta}_n(\xi_p)(\bar{C}_n - \nu)]$ have the same weak limit as $n \to \infty$ by the converging-together lemma (e.g., Theorem 25.4 of Billingsley (1995)). Since the summands in (7.1) with β replaced with $\beta_*(\xi_p)$ are i.i.d. with finite variance (because we assumed $\mathrm{Var}[C] < \infty$), the CLT gives the weak limit as $N(0, \psi_p^2)$, with $\psi_p^2 = \mathrm{Var}[I(X \le \xi_p) - \beta_*(\xi_p)(C - \nu)]$, which works out to be (7.5).

CHAPTER 8

EMPIRICAL STUDY

The previous chapters developed confidence intervals for a quantile of a random variable X having CDF F, and we established the asymptotic validity of the intervals as the computational budget $n \to \infty$. However, in practice, only finite sample sizes can be used, so we now carry out an empirical study to see how well the confidence intervals perform with finite n. In particular, we ran simulations building confidence intervals having nominal confidence level $1-\alpha$ and observe how close the estimated coverages of the intervals are to $1-\alpha$ for different values of n. (The *coverage* of a confidence interval J_n for a parameter γ based on a computational budget of n is defined to be $P(\gamma \in J_n)$. In an ideal situation, the coverage equals the nominal level $1-\alpha$ for any n, but when J_n is asymptotically valid, this is only achieved as $n \to \infty$. In practice, for finite n, the coverage often differs from $1-\alpha$, sometimes significantly.)

Our experiments entail applying crude Monte Carlo (CMC), IS, IS+SS, AV, and CV on two stochastic models: a normal distribution (described in Section 8.1), and a stochastic activity network (Section 8.2). The goal is to study how the computational budget n and the "smoothing parameter" c used in the finite-difference estimators in (4.5) and (4.6) affect the coverage of the intervals.

8.1 Normal Distribution

The first set of experiments involves estimating the pth quantile ξ_p of a standard normal random variable X, so F is the standard normal CDF Φ. For IS, we obtain the IS distribution F_* by exponentially tilting F with tilting parameter θ, defined by $F_*(dx) = e^{\theta x - \zeta(\theta)} F(dx)$, where $\zeta(\theta) = \ln(E[e^{\theta X}]) = \theta^2/2$ is the cumulant generating function of X. It is straightforward to show that F_* is also normal with unit variance and mean $\zeta'(\theta) = \theta$, the derivative of $\zeta(\theta)$. To choose a value for θ, consider the following approximation applied by Glynn (1996): $P(X > x) \approx \exp(-x\theta_x + \zeta(\theta_x))$ for $x \gg 0$, where θ_x is the root of the equation $\zeta'(\theta_x) = x$, so $\theta_x = x$. Since we are interested in x satisfying $P(X > x) = 1 - p$ (i.e., the p quantile), we arrive at the equation $-\theta^2 + \theta^2/2 = \ln(1-p)$. Solving for θ gives $\zeta'(\theta) = \sqrt{-2\ln(1-p)}$ as the mean of F_*.

For IS+SS, consider a bivariate normal pair (X,Y), where the goal is to estimate the p-quantile of X and Y is used as a stratification variable. Under the original distribution, both X and Y have standard normal marginals, and the correlation is ρ. Under the IS measure, (X,Y) is again bivariate normal with marginal means $\zeta'(\theta)$, unit variances, and the same correlation ρ.

For AV, the distribution of X is $F = \Phi$, and the antithetic pair is $(X,X') = (X,-X)$. For CV, define (X,Y) as a bivariate normal pair with standard normal marginals and correlation ρ, and define the control $C = I(Y \leq G^{-1}(p))$ when estimating the pth quantile of X, where $G = \Phi$ is the marginal distribution of Y.

8.2 Stochastic Activity Network

The other model we considered is a stochastic activity network (SAN). SANs are often employed to model the time to complete a project and are useful in project planning. We consider a simple SAN with 5 activities, which was previously considered in Hsu and Nelson (1990). Figure 8.1 illustrates the model.

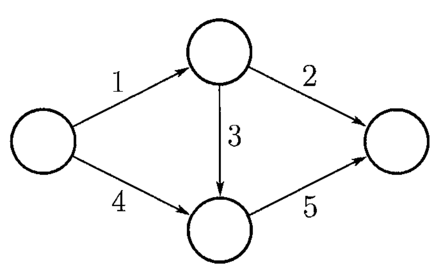

Figure 8.1 Stochastic activity network.

Let $A_1, A_2, \ldots, A_5$ be the durations of the five activities, which are i.i.d. exponentials with mean 1. Let f_i denote the density function of A_i, so $f_i(t) = e^{-t}$ for $t \geq 0$ for each $i = 1, 2, \ldots, 5$. There are 3 paths in the network. Let $B_1 = \{1,2\}$, $B_2 = \{1,3,5\}$ and $B_3 = \{4,5\}$, where B_j is the set of activities on path j. Let $m_j = |B_j|$, the number of activities on path j. Let $T_j = \sum_{i \in B_j} A_i$ be the length of path j, which has mean m_j. Let $X = \max(T_1, T_2, T_3)$ be the length of the longest path. Our goal is to estimate and construct confidence intervals for the pth quantile ξ_p of X. As noted by Hsu and Nelson (1990), the CDF of X is given by, for $x \geq 0$,

$$F(x) = 1 + (3 - 3x - x^2/2)e^{-x} + (-3 - 3x + x^2/2)e^{-2x} - e^{-3x},$$

and the density function of X is

$$f(x) = (-6 + 2x + x^2/2)e^{-x} + (3 + 7x - x^2)e^{-2x} + 3e^{-3x},$$

which is positive for all $x \geq 0$. Thus, we have $f(\xi_p) > 0$ for any $0 < p < 1$.

In the following sections, we provide details on how IS, IS+SS, and AV are applied for the SAN. For CV, we chose the control variate C as follows. Note that path 2 has the longest expectation, and we let $Y = T_2$ be the (random) length of this path. Let G denote the CDF of Y, which is Erlang-2. We then take $C = I(Y \leq G^{-1}(p))$ when estimating the pth quantile of X.

8.2.1 IS for SAN

We apply IS as in Juneja et al. (2007) by using a mixture of exponentially tilted distributions. Define f_i^θ to be the exponentially tilted version of f_i under tilting parameter θ, so $f_i^\theta(t) = e^{\theta t - \chi_i(\theta)} f_i(t)$, where $\chi_i(\theta) = \ln E[e^{\theta A_i}] = -\ln(1-\theta)$ is the cumulant generating function of A_i, which exists for $\theta < 1$. Note that $f_i^\theta(t) = (1-\theta)e^{-(1-\theta)t}$ for $t \geq 0$, and 0 otherwise; i.e., f_i^θ is the density of an exponential with rate $1-\theta$.

We will define the IS distribution to be a mixture of 3 distributions, each defined by exponentially tilting one path length T_j and not changing the distribution of activities not on that path. To do this, define positive constants $\alpha_1, \alpha_2, \alpha_3$ such that $\alpha_1+\alpha_2+\alpha_3 = 1$; these will be the mixture weights, and we will later discuss how we choose specific values for α_j. Let $\theta_1, \theta_2, \theta_3$ be positive constants, and θ_j will be the tilting parameter under the jth distribution in the mixture; we will define θ_j later. For each $j = 1,2,3$, define probability measure P_j such that A_i has density $f_i^{\theta_j}$ when $i \in B_j$ and density f_i when $i \notin B_j$, and $A_1, \ldots, A_5$ are mutually independent. Now define the IS measure P_* to be the mixture of the P_j using weights α_j; i.e., $P_*(A) = \sum_{j=1}^3 \alpha_j P_j(A)$ for any event A. The likelihood ratio is then

$$L = \left[\sum_{j=1}^{3} \alpha_j \exp\left(\theta_j T_j - \zeta_j(\theta_j)\right)\right]^{-1}, \tag{8.1}$$

where $\zeta_j(\theta) = \sum_{i\in B_j} \chi_i(\theta) = -m_j \ln(1-\theta)$ is the cumulant generating function of T_j.

We now discuss how to choose each tilting parameter θ_j using an approach described in Glynn (1996). Large-deviations theory suggests that under certain conditions

$$P(T_j > x) \approx \exp(-x\theta_x + \zeta_j(\theta_x)) \tag{8.2}$$

for $x \gg E[T_j] = m_j$, where θ_x is the root of the equation $\zeta_j'(\theta_x) = x$ and prime denotes derivative, so $\zeta_j'(\theta) = m_j/(1-\theta)$. Setting the RHS of (8.2) equal to $1-p$ yields $-\theta\zeta_j'(\theta) + \zeta_j(\theta) = \ln(1-p)$, or

$$-\frac{m_j\theta}{1-\theta} - m_j \ln(1-\theta) = \ln(1-p). \tag{8.3}$$

We then take θ_j to be the root of (8.3). Thus, we obtain $\zeta_j'(\theta_j) = m_j/(1-\theta_j)$ as a (crude) approximation for the pth quantile of T_j (under the original measure) when $p \approx 1$.

Since we will be estimating the pth quantile ξ_p of X for $p \approx 1$, we will use the IS CDF estimator $\tilde{F}_{n,\mathrm{IS}}'(x) = 1 - \frac{1}{n}\sum_{i=1}^n L_i I(X_i > x)$. The resulting quantile estimator $\tilde{\xi}_{p,n}' =$

$\tilde{F}'^{-1}_{n,\mathrm{IS}}(p)$ satisfies the CLT $\frac{\sqrt{n}}{\kappa_p}(\tilde{\xi}'_{p,n} - \xi_p) \xrightarrow{L} N(0,1)$ as $n \to \infty$, where $\kappa_p = \psi_p \phi_p$, $\phi_p = 1/f(\xi_p)$, and $\psi_p^2 = E_*[L^2 I(X > \xi_p)] - (1-p)^2$.

We now describe how to choose the IS mixture weights α_j by modifying a heuristic in Juneja et al. (2007). Let E_j denote expectation under measure P_j, and let E_* be expectation under measure P_*. Since $\zeta_j'(\theta_j)$ is roughly equal to the pth quantile of T_j and $X = \max_j T_j$, we now approximate ξ_p via $\bar{\xi}_p \equiv \max_j \zeta_j'(\theta_j)$, which leads to approximating the first term (the second moment) in ψ_p^2 by $E_*[L^2 I(X > \bar{\xi}_p)]$. Now we develop an upper bound for this quantity. On the event $\{T_j > \bar{\xi}_p\}$, we have that $L \leq K_j/\alpha_j$ for $\theta_j > 0$ by (8.1), where $K_j = \exp(-\theta_j \bar{\xi}_p + \zeta_j(\theta_j))$. Hence, since $\{X > \bar{\xi}_p\} = \cup_{j=1}^3 \{T_j > \bar{\xi}_p\}$, we get

$$E_*[L^2 I(X > \bar{\xi}_p)] \leq \left(\max_{j=1,2,3} \frac{K_j}{\alpha_j} \right)^2.$$

Choosing $\alpha_j \geq 0$ to minimize the above bound subject to $\sum_{j=1}^3 \alpha_j = 1$ gives $\alpha_j^* = K_j / \sum_{l=1}^3 K_l$.

We now put this all together in the algorithm below, in which all samples are generated independently and where $\mathrm{Exp}(\eta)$ denotes an exponential distribution with rate η.

IS algorithm for SAN example:

1. For each $j = 1,2,3$, define θ_j to be the root of (8.3), and define $\alpha_j = K_j / \sum_{l=1}^3 K_l$, where $K_l = (1-\theta_l)^{-m_l} \exp(-\theta_l \bar{\xi}_p)$ and $\bar{\xi}_p = \max_r m_r/(1-\theta_r)$. Also, define $\eta_j = 1 - \theta_j$. Set $N = 0$, which is the total number of samples collected thus far.

2. Let $N = N+1$, and generate $U_N \sim \mathrm{unif}[0,1]$.

3. If $U_N \leq \alpha_1$, then generate $A_1 \sim \mathrm{Exp}(\eta_1)$, $A_2 \sim \mathrm{Exp}(\eta_1)$, $A_3 \sim \mathrm{Exp}(1)$, $A_4 \sim \mathrm{Exp}(1)$, $A_5 \sim \mathrm{Exp}(1)$.

4. If $\alpha_1 < U_N \leq \alpha_1 + \alpha_2$, then generate $A_1 \sim \mathrm{Exp}(\eta_2)$, $A_2 \sim \mathrm{Exp}(1)$, $A_3 \sim \mathrm{Exp}(\eta_2)$, $A_4 \sim \mathrm{Exp}(1)$, $A_5 \sim \mathrm{Exp}(\eta_2)$.

5. If $\alpha_1 + \alpha_2 < U_N \leq \alpha_1 + \alpha_2 + \alpha_3$, then generate $A_1 \sim \mathrm{Exp}(1)$, $A_2 \sim \mathrm{Exp}(1)$, $A_3 \sim \mathrm{Exp}(1), A_4 \sim \mathrm{Exp}(\eta_3), A_5 \sim \mathrm{Exp}(\eta_3)$.

6. Compute $T_1 = A_1 + A_2$, $T_2 = A_1 + A_3 + A_5$, $T_3 = A_4 + A_5$, $X_N = \max(T_1, T_2, T_3)$, and the likelihood ratio L_N as in (8.1).

7. If $N < n$, then goto step 2. Otherwise, compute the IS estimator and confidence interval, and stop.

8.2.2 IS+SS for SAN

We will use $Y = T_2$ as a stratification variable, so we now want to compute the CDF G_* of Y under IS measure P_*. First let G_j be the CDF of Y under measure P_j, and define $\eta_j = 1 - \theta_j$. Under measure P_1, we have that A_1 is exponential with rate η_1 while A_3 and A_5 are both exponential with rate 1, with A_1, A_3, A_5 mutually independent. Thus, for $t > 0$ we have

$$
\begin{aligned}
G_1(t) &= P_1\{A_1 + A_3 + A_5 \leq t\} \\
&= E_1[P_1\{A_3 + A_5 \leq t - A_1 \mid A_1\}] \\
&= E_1[(1 - e^{-(t-A_1)} - (t - A_1)e^{-(t-A_1)}) I(A_1 \leq t)] \\
&= P_1\{A_1 \leq t\} - \int_0^t e^{-(t-x)} \eta_1 e^{-\eta_1 x} dx - \int_0^t (t-x) e^{-(t-x)} \eta_1 e^{-\eta_1 x} dx \\
&= 1 - e^{-\eta_1 t} \left(1 + \frac{\eta_1}{1-\eta_1} + \frac{\eta_1}{(1-\eta_1)^2}\right) + e^{-t} \frac{\eta_1}{1-\eta_1} \left(1 + t + \frac{1}{1-\eta_1}\right).
\end{aligned}
$$

Similarly, we can show that

$$
G_3(t) = 1 - e^{-\eta_3 t} \left(1 + \frac{\eta_3}{1-\eta_3} + \frac{\eta_3}{(1-\eta_3)^2}\right) + e^{-t} \frac{\eta_3}{1-\eta_3} \left(1 + t + \frac{1}{1-\eta_3}\right).
$$

Finally, under measure P_2, we have that A_1, A_3, A_5 are i.i.d. exponential with rate η_2, so T_2 has an Erlang-3 distribution with scale parameter η_2; i.e., $G_2(t) = 1 - e^{-\eta_2 t} - \eta_2 t e^{-\eta_2 t} -$

$(\eta_2 t)^2 e^{-\eta_2 t}/2$. Thus, the CDF of Y under IS measure P_* is given by $G_*(t) = \sum_{j=1}^3 \alpha_j G_j(t)$.

Define k strata $S_1, S_2, \ldots, S_k$ for the stratification variable Y, and let $\lambda_i = P_*\{Y \in S_i\}$. Also, define $\gamma_i > 0$ with $\sum_{i=1}^k \gamma_i = 1$, and let $n_i = n\gamma_i$. In our experiments, we generate stratified samples using the "bin tossing" method of Glasserman et al. (2000b).

Let $\mathrm{Exp}(\eta)$ denote an exponential distribution with rate η. We then have the following algorithm, in which all generated samples are independent.

IS+SS algorithm with bin tossing for SAN:

1. For each $j = 1,2,3$, define θ_j to be the root of (8.3), and define $\alpha_j = K_j / \sum_{l=1}^3 K_l$, where $K_l = (1-\theta_l)^{-m_l} \exp(-\theta_l \bar{\xi}_p)$ and $\bar{\xi}_p = \max_r m_r/(1-\theta_r)$. Also, define $\eta_j = 1-\theta_j$. Set $N_i = 0$ for $i = 1, \ldots, k$, where N_i is the number of samples in stratum i collected thus far. Let $N = 0$, which is the number of strata that have enough samples collected.

2. Let $N = N+1$, and generate $U_N \sim \mathrm{unif}[0,1]$.

3. If $U_N \le \alpha_1$, then generate $A_1 \sim \mathrm{Exp}(\eta_1)$, $A_2 \sim \mathrm{Exp}(\eta_2)$, $A_3 \sim \mathrm{Exp}(1)$, $A_4 \sim \mathrm{Exp}(1)$, $A_5 \sim \mathrm{Exp}(1)$.

4. If $\alpha_1 < U_N \le \alpha_1 + \alpha_2$, then generate $A_1 \sim \mathrm{Exp}(\eta_2)$, $A_2 \sim \mathrm{Exp}(1)$, $A_3 \sim \mathrm{Exp}(\eta_2)$, $A_4 \sim \mathrm{Exp}(1)$, $A_5 \sim \mathrm{Exp}(\eta_2)$.

5. If $\alpha_1 + \alpha_2 < U_N \le \alpha_1 + \alpha_2 + \alpha_3$, then generate $A_1 \sim \mathrm{Exp}(1)$, $A_2 \sim \mathrm{Exp}(1)$, $A_3 \sim \mathrm{Exp}(1)$, $A_4 \sim \mathrm{Exp}(\eta_3)$, $A_5 \sim \mathrm{Exp}(\eta_3)$.

6. Compute $T_1 = A_1 + A_2$, $T_2 = A_1 + A_3 + A_5$, $T_3 = A_4 + A_5$, $X = \max(T_1, T_2, T_3)$, $Y = T_2$, and L as in (8.1).

7. Find i such that $Y \in S_i$, and let $N_i = N_i + 1$. If $N_i \le n_i$, then set $X_{i,N_i} = X$, $Y_{i,N_i} = Y$, and $L_{i,N_i} = L$; otherwise, discard (X, Y, L).

8. If $N < n$, then goto step 2. Otherwise, compute the IS+SS estimator and confidence interval, and stop.

8.2.3 AV for SAN

To apply AV, we generate each sample X as follows. Let $U_1, \ldots, U_5$ be i.i.d. unif$[0,1]$ random numbers, and for each $i = 1, \ldots, 5$, set $A_i = -\ln(1-U_i)$ and $A_i' = -\ln(U_i)$. Then for $j = 1,2,3$, set $T_j = \sum_{i \in B_j} A_i$ and $T_j' = \sum_{i \in B_j} A_i'$. Finally, set $X = \max(T_1, T_2, T_3)$ and $X' = \max(T_1', T_2', T_3')$ as the antithetic pair of outputs. Since X is monotonic in each U_i, we have that X and X' are negatively correlated; see p. 181 of Ross (1997).

8.3 Choosing the Smoothing Parameter c

Recall $\tilde{\phi}_{p,n,2}(c)$ from (4.6), the centered finite-difference estimator of ϕ_p, where n is the computational budget. The estimator $\tilde{\phi}_{p,n,2}(c)$ scales the difference of quantile estimators at $p + c/\sqrt{n}$ and $p - c/\sqrt{n}$ for some smoothing parameter $c \neq 0$, and we can use $\tilde{\phi}_{p,n,2}(c)$ in the last step of the algorithm on p. 37 in Chapter 4 to construct a $100(1-\alpha)\%$ confidence interval for ξ_p. We now discuss some recommendations for choosing c.

One possible goal guiding the selection of c is to minimize the mean-square error (MSE) of $\tilde{\phi}_{p,n,2}(c)$ as an estimator of ϕ_p. Alternatively, we can choose c to minimize the coverage error of the resulting confidence interval for ξ_p. In the case of crude Monte Carlo, there has been some previous work carrying out asymptotic expansions to study these two problems. Bofinger (1975) shows that under certain regularity conditions, the MSE of $\tilde{\phi}_{p,n,2}(c)$ as $n \to \infty$ takes the form

$$\mathrm{MSE}_n(c) = \frac{c^4}{n^2}\left(\frac{Q'''(p)}{6}\right)^2 + \frac{1}{2cf^2(\xi_p)\sqrt{n}} + o(n^{-1/2}), \tag{8.4}$$

where $Q(p) = F^{-1}(p)$ and Q''' is its third derivative. Since the second term in (8.4) shrinks more slowly than the first, selecting c as large as possible will maximize the rate (to first order) at which $\mathrm{MSE}_n(c)$ decreases as n grows. As for the coverage error, define the $100(1-\alpha)\%$ confidence interval for ξ_p as $J_n(c) = (\tilde{\xi}_{p,n} \pm z_{1-\alpha/2}\sqrt{p(1-p)}\tilde{\phi}_{p,n,2}(c)/\sqrt{n})$ with $z_{1-\alpha/2} = \Phi^{-1}(1-\alpha/2)$. Under certain regularity conditions, Hall and Sheather (1988) show that the coverage of $J_n(c)$ for large n satisfies

$$P\{\xi_p \in J_n(c)\} = 1 - \alpha + 2\left[\frac{u_2(z_{1-\alpha/2})}{c\sqrt{n}} + \frac{c^2 u_3(z_{1-\alpha/2})}{n}\right]\varphi(z_{1-\alpha/2}) + o(n^{-1/2}), \quad (8.5)$$

where $u_2(x) = -x^3/4$, $u_3(x) = ax$ with $a = [3f'(\xi_p)^2 - f(\xi_p)f''(\xi_p)][6f(\xi_p)^4]^{-1}$, and φ is the standard normal density function. The bracketed term in (8.5) represents the first-order error term in the coverage. The first summand in this decreases more slowly than the other, so we should choose c as large as possible to maximize the rate at which the first-order error shrinks as $n \to \infty$.

When combining different values of c as in (4.7), we can use the following idea from Glasserman (2004). Again let $Q(x) = F^{-1}(x)$, so $\phi_p = Q'(p)$, where prime denotes derivative. Let $h = cn^{-1/2}$, and a Taylor expansion of $Q(p)$ yields

$$\frac{Q(p+h) - Q(p-h)}{2h} = Q'(p) + \frac{1}{6}Q'''(p)h^2 + O(h^4),$$

where odd powers of h cancel out. Similarly,

$$\frac{Q(p+2h) - Q(p-2h)}{4h} = Q'(p) + \frac{2}{3}Q'''(p)h^2 + O(h^4),$$

so we can cancel out the order h^2 term by combining the two above results as

$$\frac{4}{3}\left(\frac{Q(p+h) - Q(p-h)}{2h}\right) - \frac{1}{3}\left(\frac{Q(p+2h) - Q(p-2h)}{4h}\right) = Q'(p) + O(h^4).$$

This suggests that when combining $r = 2$ values of c in (4.7), selecting c_1 and c_2 with $c_2 = 2c_1$, $w_1 = 4/3$ and $w_2 = -1/3$ may lead to the estimator $\bar{\phi}_{p,n,2}(c_1, c_2)$ having low bias.

8.4 Discussion of Empirical Results

In our experiments, we constructed confidence intervals for the pth quantile ξ_p using CMC, IS, IS+SS, AV and CV, where the intervals have a nominal confidence level of $1-\alpha = 0.9$. In each case, we constructed a confidence interval using the algorithm on p. 37 in Chapter 4, which consistently estimates the variance constant κ^2. This requires specifying a value for the smoothing parameter c of the finite-difference estimators of ϕ_p in (4.5) and (4.6), or values for $c_1,\ldots,c_r$ and weights $w_1,\ldots,w_r$ for the combined estimator in (4.7). We experimented with different values for these parameters and the computational budget n to study their effect on the coverage.

The experiments used $n = 50$ and $n = 100 \times 4^j$ for $0 \le j \le 4$. Also, we took $c = 2^t$ for $-4 \le t \le 2$. In all cases we estimated coverage levels using $m = 10^3$ independent replications. Also, we applied common random numbers (CRN) when possible, which can lead to sharper comparisons. For example, in tables containing results for both CV and crude Monte Carlo, the simulated output (X, C) for CV shares the same value of X used in crude Monte Carlo. In our IS+SS experiments, we used $k = 5$ equiprobable strata defined by the intervals $S_i = (G^{-1}((i-1)/k), G^{-1}(i/k)]$ for $i = 1,\ldots,5$, where G is the CDF of the stratification variable Y, so each $\lambda_i = 1/k$. Also, we let $\gamma_i = \lambda_i$.

We first discuss how coverage converges as n increases for different values of c. Figures 8.2 and 8.3 are for IS-only on a normal for $p = 0.8$ and $p = 0.95$, respectively. Figures 8.4–8.7 are for IS+SS on the bivariate normal for $\rho = 0.5$ and 0.9 and for $p = 0.8$ and $p = 0.95$. Figures 8.8 and 8.9 are for AV on the normal for $p = 0.8$ and $p = 0.95$. Figures 8.10–8.13 are for CV on the bivariate normal with $\rho = 0.5$ and $\rho = 0.9$ for $p = 0.8$ and $p = 0.95$. Figures 8.14 and 8.15 are for CMC on the normal for $p = 0.8$ and $p = 0.95$.

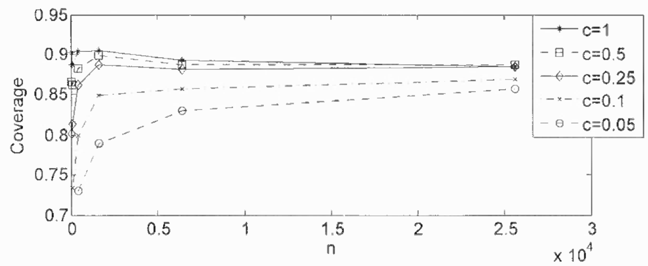

Figure 8.2 Coverage for IS-only for the normal for $p = 0.8$.

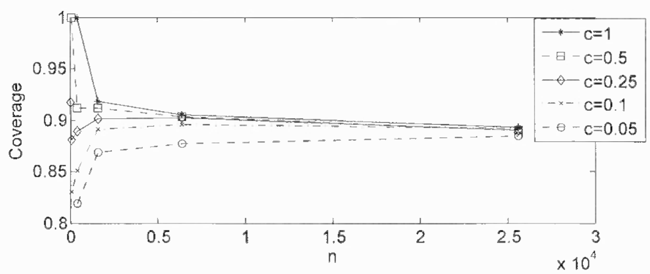

Figure 8.3 Coverage for IS-only for the normal for $p = 0.95$.

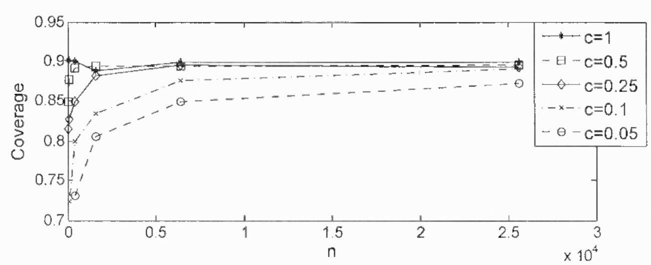

Figure 8.4 Coverage for IS+SS for $p = 0.8$ for the bivariate normal with $\rho = 0.5$.

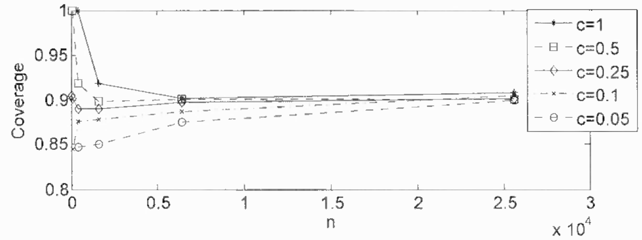

Figure 8.5 Coverage for IS+SS for $p = 0.95$ for the bivariate normal with $\rho = 0.5$.

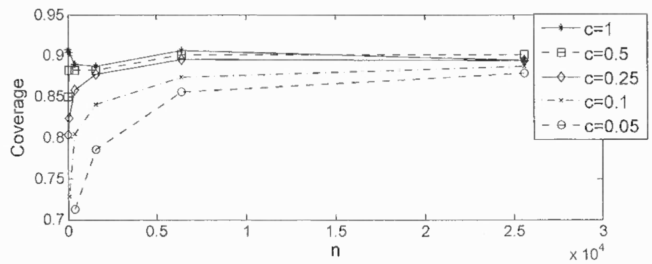

Figure 8.6 Coverage for IS+SS for $p = 0.8$ for the bivariate normal with $\rho = 0.9$.

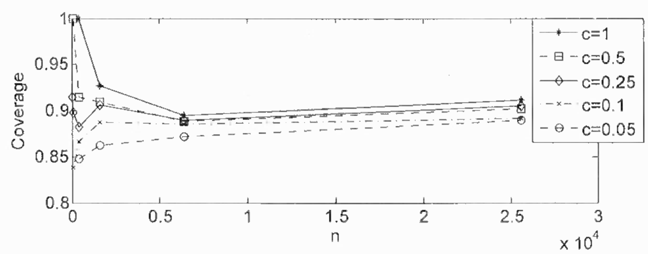

Figure 8.7 Coverage for IS+SS for $p = 0.95$ for the bivariate normal with $\rho = 0.9$.

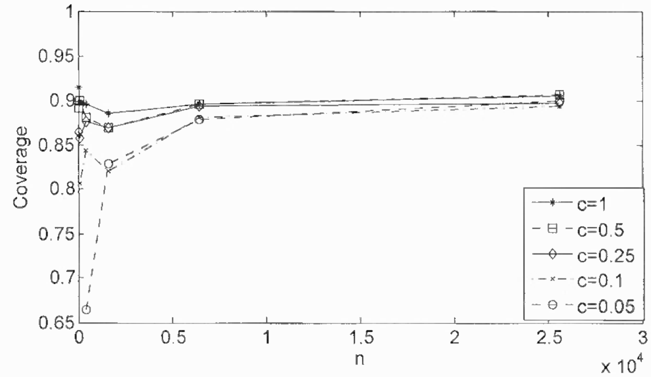

Figure 8.8 Coverage for AV for $p = 0.8$ for the normal.

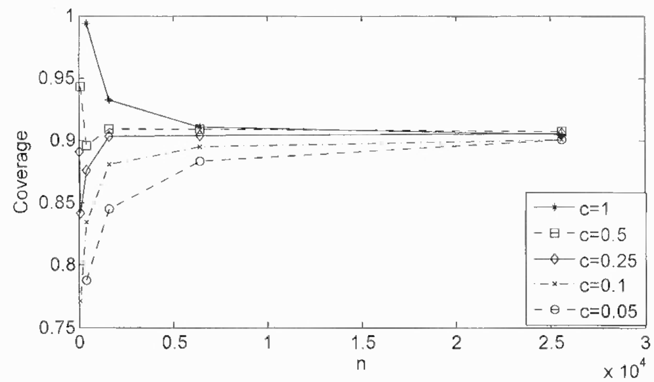

Figure 8.9 Coverage for AV for $p = 0.95$ for the normal.

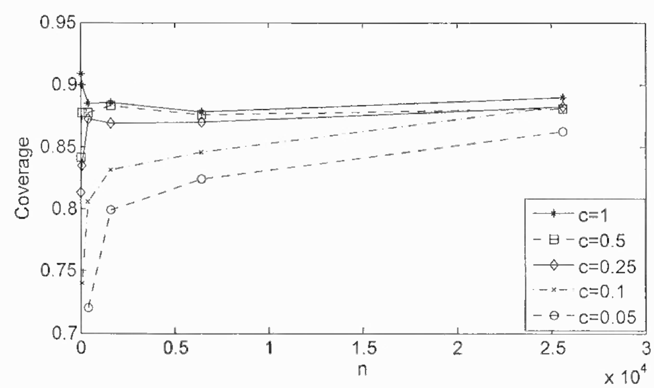

Figure 8.10 Coverage for CV for $p = 0.8$ for the bivariate normal with $\rho = 0.5$.

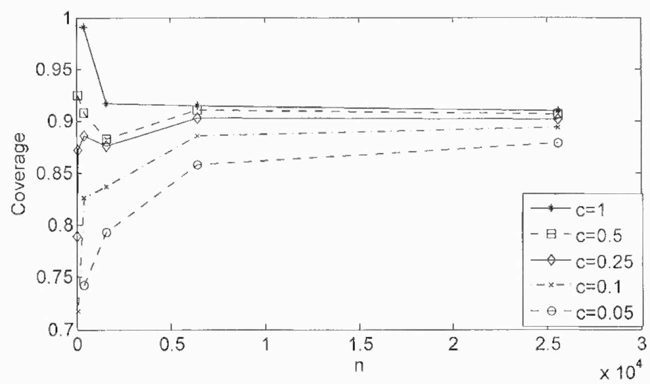

Figure 8.11 Coverage for CV for $p = 0.95$ for the bivariate normal with $\rho = 0.5$.

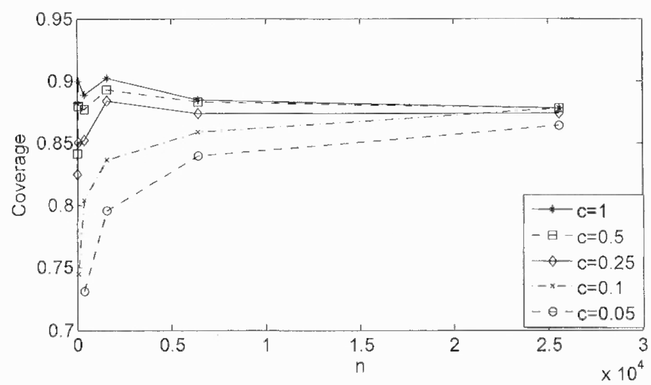

Figure 8.12 Coverage for CV for $p = 0.8$ for the bivariate normal with $\rho = 0.9$.

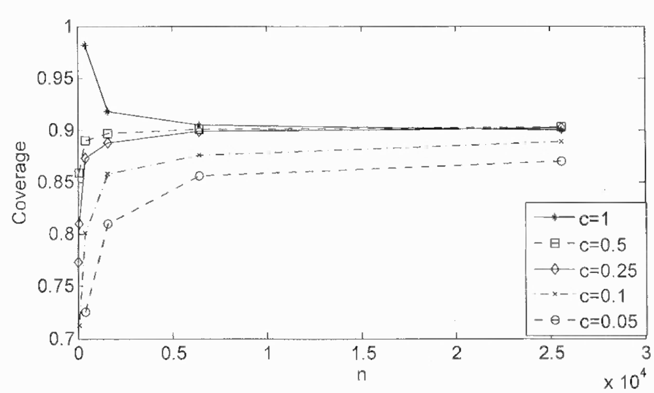

Figure 8.13 Coverage for CV for $p = 0.95$ for the bivariate normal with $\rho = 0.9$.

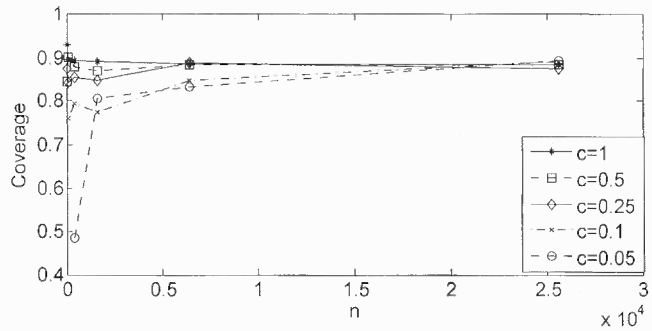

Figure 8.14 Coverage for CMC for $p = 0.8$ for the normal.

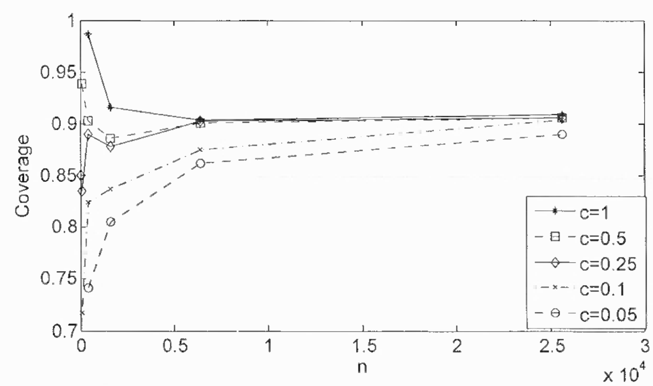

Figure 8.15 Coverage for CMC for $p = 0.95$ for the normal.

In general we see that as n increases for a fixed c and method, the coverage levels converge to the nominal level of 0.9 in each case. Also, for fixed n, larger values of c seem to lead to coverages that are closer to 0.9 than the smaller c for CMC and all of the VRTs. Moreover, as n increases, the convergence appears to be faster for large c. Thus, the observations in Section 8.3 for CMC that choosing large c is better appear to also be valid for VRTs.

Figures 8.16–8.20 are for IS-only, IS+SS, AV, CV and CMC on the SAN for $p = 0.95$. The same patterns appear in the results for the SAN that were previously exhibited for the normal case.

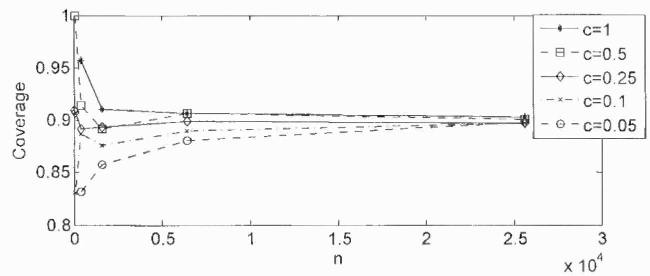

Figure 8.16 Coverage for IS-only for $p = 0.95$ for the SAN.

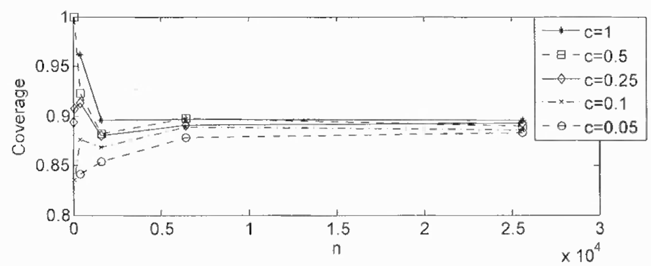

Figure 8.17 Coverage for IS+SS for $p = 0.95$ for the SAN.

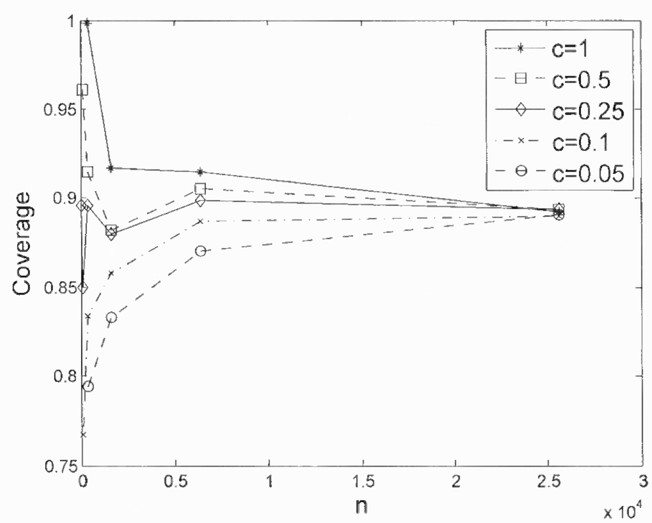

Figure 8.18 Coverage for AV for $p = 0.95$ for the SAN.

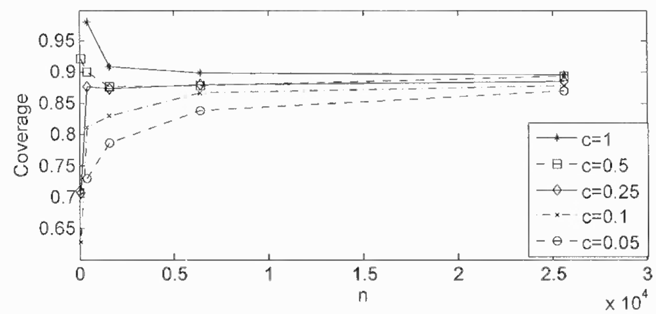

Figure 8.19 Coverage for CV for $p = 0.95$ for the SAN.

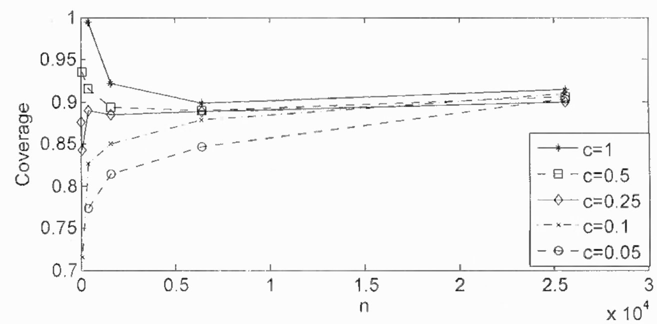

Figure 8.20 Coverage for CMC for $p = 0.95$ for the SAN.

Tables 8.1–8.40 provide more detailed results about coverage and average half-widths, where Tables 8.1–8.20 are for the normal or bivariate normal model, and Tables 8.21–8.40 contain results for the SAN. In each table the first column gives the computational budget n. The next three columns give the results for the centered, forward and backward finite-difference (FD) estimators of ϕ_p from (4.6) and (4.5).

Two boundary conditions of p_n in (4.6) and (4.5) need to be satisfied while estimating ϕ_p: 1) $p + c/\sqrt{n} \leq 1$, when $c > 0$; 2) $|n \times c/\sqrt{n}| \geq 1$, which, equivalently, means the perturbation position needs to be at least one. If one of them is not satisfied, then corresponding coverages and half-widths will display "NaN". For instance, when $p = 0.95$, $c = 1$ and $n < 400$, the coverages and half-widths display "NaN".

Table 8.1: Coverages for IS with $c = 0.1$ for the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.901	0.845	0.193
100	0.842	0.820	0.742	0.887	0.881	0.253
400	0.870	0.855	0.819	0.889	0.894	0.393
1600	0.881	0.874	0.854	0.896	0.901	0.525
6400	0.892	0.887	0.876	0.902	0.902	0.614
25600	0.897	0.896	0.893	0.899	0.904	0.705

Table 8.2: Average Half-Widths for IS with $c = 0.1$ for the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.271	0.161	0.757
100	0.119	0.120	0.129	0.185	0.119	0.661
400	0.061	0.062	0.064	0.074	0.061	0.636
1600	0.031	0.031	0.032	0.034	0.031	0.477
6400	0.016	0.016	0.016	0.017	0.016	0.337
25600	0.007	0.008	0.007	0.008	0.008	0.229

In terms of coverage, the backward FD estimator appears to do worse than the other two, and the centered FD estimator might perform slightly better than the forward estimator. This complements the MSE analysis in Section 7.1 of Glasserman (2004) showing that centered finite-difference estimators of the derivative of a mean have asymptotically smaller MSE than forward estimators.

Table 8.3: Coverage for IS with $c = 0.5$ for the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.881	0.861	0.203
100	1.000	1.000	0.727	0.873	0.860	0.281
400	0.912	0.959	0.809	0.884	0.889	0.381
1600	0.912	0.924	0.869	0.884	0.902	0.499
6400	0.903	0.917	0.885	0.898	0.897	0.628
25600	0.890	0.894	0.888	0.900	0.892	0.725

Table 8.4: Average Half-Widths for IS with $c = 0.5$ for the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.271	0.161	0.815
100	0.452	0.816	0.090	0.183	0.119	0.828
400	0.066	0.080	0.051	0.073	0.061	0.622
1600	0.031	0.035	0.028	0.036	0.031	0.444
6400	0.016	0.016	0.014	0.017	0.016	0.366
25600	0.008	0.008	0.007	0.008	0.008	0.243

Table 8.5: Coverages for IS+SS with $c = 0.5$ on the Bivariate Normal Distribution with $\rho = 0.9$ when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.782	0.867	0.184
100	1.000	1.000	0.756	0.848	0.873	0.250
400	0.915	0.953	0.812	0.884	0.890	0.407
1600	0.909	0.930	0.872	0.899	0.904	0.531
6400	0.888	0.904	0.871	0.894	0.885	0.611
25600	0.902	0.909	0.892	0.906	0.908	0.684

Table 8.6: Average Half-Widths for IS+SS with $c = 0.5$ for the Bivariate Normal with $\rho = 0.9$ When $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.271	0.146	0.971
100	0.411	0.741	0.081	0.166	0.107	0.798
400	0.059	0.072	0.047	0.064	0.055	0.619
1600	0.028	0.031	0.026	0.031	0.028	0.545
6400	0.014	0.015	0.013	0.015	0.014	0.296
25600	0.007	0.007	0.007	0.008	0.007	0.233

Table 8.7: Coverages for IS+SS with $c = 0.25$ on the Bivariate Normal Distribution with $\rho = 0.9$ when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean	Combining $c_1 = 0.25$ and $c_2 = 0.5$
50	0.915	0.965	0.731	0.782	0.867	0.184	NaN
100	0.898	0.927	0.779	0.848	0.873	0.250	0.236
400	0.882	0.910	0.822	0.884	0.890	0.407	0.870
1600	0.906	0.914	0.884	0.899	0.904	0.531	0.902
6400	0.890	0.895	0.877	0.894	0.885	0.611	0.887
25600	0.905	0.909	0.897	0.906	0.908	0.684	0.903

Table 8.8: Average Half-Widths for IS+SS with $c = 0.25$ for the Bivariate Normal with $\rho = 0.9$ When $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean	Combining $c_1 = 0.25$ and $c_2 = 0.5$
50	0.174	0.235	0.113	0.271	0.146	0.971	NaN
100	0.115	0.139	0.090	0.166	0.107	0.798	0.016
400	0.056	0.062	0.050	0.064	0.055	0.619	0.055
1600	0.028	0.030	0.025	0.031	0.028	0.545	0.028
6400	0.014	0.014	0.013	0.015	0.014	0.296	0.014
25600	0.007	0.007	0.007	0.008	0.007	0.233	0.007

Table 8.9: Coverages for AV with $c = 1$ for the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.850	0.791	1.000
100	NaN	NaN	NaN	0.459	0.853	1.000
400	0.994	1.000	0.751	0.746	0.886	1.000
1600	0.932	0.967	0.827	0.859	0.910	1.000
6400	0.911	0.946	0.875	0.889	0.909	1.000
25600	0.905	0.923	0.897	0.894	0.907	1.000

Table 8.10: Average Half-Widths for AV with $c = 1$ for the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.315	0.252	0.000
100	NaN	NaN	NaN	0.310	0.211	0.000
400	0.227	0.370	0.086	0.123	0.116	0.000
1600	0.063	0.077	0.050	0.064	0.059	0.000
6400	0.030	0.036	0.027	0.032	0.030	0.000
25600	0.015	0.016	0.014	0.016	0.015	0.000

Table 8.11: Coverages for AV with $c = 0.5$ for the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.850	0.791	1.000
100	0.943	0.979	0.696	0.459	0.853	1.000
400	0.896	0.950	0.804	0.746	0.886	1.000
1600	0.909	0.936	0.859	0.859	0.910	1.000
6400	0.909	0.929	0.885	0.889	0.909	1.000
25600	0.907	0.913	0.895	0.894	0.907	1.000

Table 8.12: Average Half-Widths for AV with $c = 0.5$ for the the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.315	0.252	0.000
100	0.321	0.493	0.149	0.310	0.211	0.000
400	0.123	0.150	0.096	0.123	0.116	0.000
1600	0.060	0.066	0.054	0.064	0.059	0.000
6400	0.030	0.032	0.028	0.032	0.030	0.000
25600	0.015	0.015	0.015	0.016	0.015	0.000

Table 8.13: Coverages for CV with $c = 1$ for the Bivariate Normal with $\rho = 0.9$ when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.113	0.728	0.913
100	NaN	NaN	NaN	0.169	0.740	0.905
400	0.982	0.999	0.758	0.777	0.889	0.906
1600	0.918	0.969	0.827	0.908	0.908	0.908
6400	0.905	0.927	0.865	0.903	0.905	0.892
25600	0.900	0.918	0.882	0.903	0.905	0.892

Table 8.14: Average Half-Widths for CV with $c = 1$ for the Normal with $\rho = 0.9$ when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.799	0.360	0.108
100	NaN	NaN	NaN	0.671	0.263	0.074
400	0.238	0.373	0.103	0.178	0.137	0.036
1600	0.073	0.086	0.058	0.078	0.068	0.018
6400	0.035	0.038	0.031	0.037	0.034	0.009
25600	0.017	0.019	0.016	0.019	0.017	0.005

Table 8.15: Coverages for CV with $c = 0.5$ for the Bivariate Normal with $\rho = 0.9$ when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.113	0.728	0.913
100	0.825	0.850	0.675	0.169	0.740	0.905
400	0.890	0.918	0.800	0.777	0.889	0.906
1600	0.897	0.917	0.851	0.908	0.908	0.908
6400	0.901	0.912	0.875	0.903	0.905	0.892
25600	0.903	0.906	0.891	0.903	0.905	0.892

Table 8.16: Average Half-Widths for CV with $c = 0.5$ for the Bivariate Normal with $\rho = 0.9$ when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.799	0.360	0.108
100	0.329	0.468	0.190	0.671	0.263	0.074
400	0.145	0.174	0.116	0.178	0.137	0.036
1600	0.069	0.076	0.062	0.078	0.068	0.018
6400	0.034	0.036	0.032	0.037	0.034	0.009
25600	0.017	0.018	0.016	0.019	0.017	0.005

Table 8.17: Coverages for CMC with $c = 1$ for the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.333	0.910	0.878
100	NaN	NaN	NaN	0.832	0.914	0.891
400	0.987	0.998	0.788	0.663	0.907	0.894
1600	0.916	0.958	0.818	0.811	0.901	0.911
6400	0.904	0.933	0.875	0.889	0.905	0.894
25600	0.909	0.922	0.888	0.891	0.916	0.890

Table 8.18: Average Half-Widths for CMC with $c = 1$ for the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.376	0.492	0.232
100	NaN	NaN	NaN	0.330	0.347	0.164
400	0.304	0.448	0.129	0.171	0.174	0.082
1600	0.092	0.111	0.073	0.091	0.087	0.041
6400	0.044	0.049	0.039	0.046	0.043	0.021
25600	0.022	0.023	0.020	0.023	0.022	0.010

Table 8.19: Coverages for CMC with $c = 0.5$ for the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.333	0.910	0.897
100	0.939	0.968	0.711	0.832	0.914	0.891
400	0.903	0.926	0.828	0.663	0.907	0.894
1600	0.886	0.900	0.841	0.811	0.901	0.911
6400	0.901	0.907	0.885	0.889	0.905	0.894
25600	0.906	0.914	0.897	0.891	0.916	0.890

Table 8.20: Average Half-Widths for CMC with $c = 0.5$ for the Normal when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.376	0.492	0.232
100	0.455	0.668	0.242	0.330	0.347	0.164
400	0.181	0.215	0.147	0.171	0.174	0.082
1600	0.088	0.097	0.079	0.091	0.087	0.041
6400	0.043	0.046	0.041	0.046	0.043	0.021
25600	0.022	0.022	0.021	0.023	0.022	0.010

Table 8.21: Coverages for IS with $c = 1$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.844	0.842	0.885
100	NaN	NaN	NaN	0.838	0.891	0.903
400	1.000	1.000	0.696	0.880	0.897	0.897
1600	0.958	0.997	0.789	0.903	0.895	0.895
6400	0.913	0.954	0.840	0.903	0.901	0.898
25600	0.906	0.931	0.871	0.912	0.901	0.887

Table 8.22: Average Half-Widths for IS with $c = 1$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.823	0.536	1.003
100	NaN	NaN	NaN	0.536	0.400	0.708
400	1.475	2.819	0.157	0.235	0.207	0.354
1600	0.132	0.185	0.089	0.114	0.104	0.178
6400	0.055	0.065	0.048	0.057	0.052	0.089
25600	0.026	0.029	0.025	0.029	0.026	0.045

Table 8.23: Coverages for IS with $c = 0.5$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.844	0.842	0.885
100	1.000	1.000	0.727	0.838	0.891	0.903
400	0.914	0.961	0.807	0.880	0.897	0.897
1600	0.892	0.937	0.844	0.903	0.895	0.895
6400	0.906	0.919	0.879	0.903	0.901	0.898
25600	0.901	0.909	0.889	0.912	0.901	0.887

Table 8.24: Average Half-Widths for IS with $c = 0.5$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.823	0.536	1.003
100	3.444	6.609	0.278	0.536	0.400	0.709
400	0.226	0.282	0.170	0.235	0.207	0.356
1600	0.106	0.120	0.093	0.114	0.104	0.178
6400	0.053	0.056	0.050	0.057	0.052	0.089
25600	0.026	0.029	0.025	0.029	0.026	0.045

Table 8.25: Coverages for IS+SS with $c = 0.5$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.849	0.825	0.845
100	1.000	1.000	0.742	0.876	0.881	0.866
400	0.923	0.965	0.817	0.904	0.899	0.892
1600	0.882	0.920	0.835	0.900	0.880	0.914
6400	0.898	0.911	0.876	0.904	0.902	0.898
25600	0.890	0.899	0.883	0.893	0.895	0.910

Table 8.26: Average Half-Widths for IS+SS with $c = 0.5$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.661	0.449	0.534
100	2.909	5.578	0.241	0.425	0.338	0.391
400	0.190	0.236	0.143	0.191	0.175	0.201
1600	0.090	0.100	0.079	0.097	0.088	0.101
6400	0.044	0.048	0.042	0.048	0.044	0.051
25600	0.022	0.023	0.022	0.024	0.022	0.025

Table 8.27: Coverages for IS+SS with $c = 0.25$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean	Combining $c_1 = 0.25$ and $c_2 = 0.5$
50	0.894	0.938	0.702	0.849	0.825	0.845	NaN
100	0.907	0.937	0.854	0.876	0.881	0.866	0.879
400	0.913	0.914	0.842	0.904	0.899	0.892	0.895
1600	0.880	0.895	0.845	0.900	0.880	0.914	0.881
6400	0.891	0.902	0.878	0.904	0.902	0.898	0.890
25600	0.893	0.890	0.885	0.893	0.895	0.910	0.890

Table 8.28: Average Half-Widths for IS+SS with $c = 0.25$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean	Combining $c_1 = 0.25$ and $c_2 = 0.5$
50	0.583	0.734	0.343	0.661	0.449	0.534	NaN
100	0.362	0.450	0.276	0.425	0.338	0.391	0.476
400	0.177	0.198	0.157	0.192	0.175	0.200	0.173
1600	0.088	0.093	0.083	0.097	0.088	0.101	0.088
6400	0.044	0.046	0.043	0.048	0.044	0.051	0.044
25600	0.022	0.022	0.022	0.024	0.022	0.025	0.022

Table 8.29: Coverages for AV with $c = 1$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.858	0.788	0.879
100	NaN	NaN	NaN	0.492	0.855	0.900
400	0.998	1.000	0.752	0.782	0.908	0.907
1600	0.917	0.971	0.802	0.864	0.888	0.906
6400	0.915	0.936	0.861	0.888	0.904	0.908
25600	0.892	0.909	0.879	0.893	0.895	0.910

Table 8.30: Average Half-Widths for AV with $c = 1$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.887	0.701	0.212
100	NaN	NaN	NaN	0.565	0.584	0.150
400	0.751	1.272	0.226	0.334	0.321	0.075
1600	0.178	0.222	0.134	0.176	0.164	0.038
6400	0.084	0.094	0.074	0.090	0.082	0.019
25600	0.041	0.044	0.039	0.044	0.041	0.009

Table 8.31: Coverages for AV with $c = 0.5$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.858	0.788	0.879
100	0.961	0.989	0.670	0.492	0.855	0.900
400	0.915	0.962	0.811	0.782	0.908	0.907
1600	0.882	0.927	0.831	0.864	0.888	0.906
6400	0.905	0.922	0.875	0.888	0.904	0.908
25600	0.894	0.903	0.886	0.893	0.895	0.910

Table 8.32: Average Half-Widths for AV with $c = 0.5$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.887	0.701	0.212
100	1.016	1.632	0.400	0.565	0.584	0.150
400	0.344	0.426	0.261	0.334	0.321	0.075
1600	0.167	0.186	0.148	0.176	0.164	0.038
6400	0.082	0.087	0.078	0.090	0.082	0.019
25600	0.041	0.043	0.040	0.044	0.041	0.009

Table 8.33: Coverages for CV with $c = 1$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.114	0.531	0.881
100	NaN	NaN	NaN	0.194	0.533	0.870
400	0.980	0.995	0.762	0.676	0.884	0.901
1600	0.908	0.956	0.806	0.877	0.886	0.890
6400	0.896	0.923	0.842	0.905	0.891	0.886
25600	0.898	0.909	0.879	0.896	0.895	0.891

Table 8.34: Average Half-Widths for CV with $c = 1$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.796	0.629	0.186
100	NaN	NaN	NaN	0.613	0.448	0.133
400	0.590	0.966	0.213	0.412	0.290	0.068
1600	0.165	0.204	0.125	0.176	0.150	0.034
6400	0.077	0.087	0.068	0.083	0.076	0.017
25600	0.038	0.041	0.036	0.041	0.038	0.008

Table 8.35: Coverages for CV with $c = 0.5$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.114	0.531	0.881
100	0.640	0.641	0.593	0.194	0.533	0.870
400	0.900	0.918	0.823	0.676	0.884	0.901
1600	0.877	0.905	0.823	0.877	0.886	0.890
6400	0.878	0.901	0.852	0.905	0.891	0.886
25600	0.894	0.896	0.884	0.896	0.895	0.891

Table 8.36: Average Half-Widths for CV with $c = 0.5$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.796	0.629	0.186
100	0.649	0.925	0.375	0.613	0.448	0.133
400	0.323	0.393	0.253	0.412	0.290	0.068
1600	0.153	0.171	0.135	0.176	0.150	0.034
6400	0.076	0.081	0.072	0.083	0.076	0.017
25600	0.038	0.039	0.036	0.041	0.038	0.008

Table 8.37: Coverages for CMC with $c = 1$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.420	0.918	0.897
100	NaN	NaN	NaN	0.860	0.905	0.886
400	0.994	1.000	0.756	0.867	0.908	0.896
1600	0.922	0.970	0.805	0.839	0.899	0.902
6400	0.898	0.927	0.855	0.893	0.893	0.914
25600	0.914	0.936	0.880	0.907	0.913	0.915

Table 8.38: Average Half-Widths for CMC with $c = 1$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ	Estimating Mean
50	NaN	NaN	NaN	0.933	1.346	0.392
100	NaN	NaN	NaN	0.920	0.952	0.279
400	0.964	1.595	0.332	0.452	0.476	0.140
1600	0.258	0.321	0.194	0.253	0.238	0.070
6400	0.121	0.136	0.106	0.129	0.119	0.035
25600	0.060	0.063	0.056	0.064	0.060	0.018

Table 8.39: Coverages for CMC with $c = 0.5$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.420	0.918	0.897
100	0.935	0.967	0.714	0.860	0.905	0.886
400	0.915	0.940	0.798	0.867	0.908	0.896
1600	0.893	0.922	0.826	0.839	0.899	0.902
6400	0.890	0.895	0.868	0.893	0.893	0.914
25600	0.905	0.915	0.889	0.907	0.913	0.915

Table 8.40: Average Half-Widths for CMC with $c = 0.5$ for the SAN when $p = 0.95$

n	Centered FD	Forward FD	Backward FD	Batching	Exact ϕ_p	Estimating Mean
50	NaN	NaN	NaN	0.933	1.346	0.392
100	1.373	2.105	0.638	0.920	0.952	0.279
400	0.503	0.623	0.386	0.452	0.476	0.140
1600	0.240	0.250	0.211	0.253	0.238	0.070
6400	0.120	0.124	0.112	0.129	0.119	0.035
25600	0.060	0.060	0.058	0.064	0.060	0.018

We also wanted to see the "penalty" for estimating ϕ_p to construct confidence intervals. To do this, we also constructed intervals using the exact value of ϕ_p. The columns in the tables labeled "Exact ϕ_p" contain these results. In general, we see that coverage levels are closer to the nominal level of 0.9 when the exact value of ϕ_p is used instead of being estimated.

For comparison, we also used batching (with $b = 10$ batches) as an alternative approach to construct confidence intervals; see the columns labeled "Batching" in the tables. (In batching we allocate a computational budget of n/b to each batch, with batches simulated independently. We form an estimator of the quantile from each batch, and then take the sample mean and sample variance of the resulting b quantile estimators to construct a confidence interval for ξ_p.) The empirical results show that batching generally leads to somewhat better coverage levels than the methods based on the finite-difference estimators of ϕ_p. However, the confidence intervals for batching have larger average half-widths.

We also constructed confidence intervals for the mean as another benchmark for comparison. For AV, as shown in Tables 8.9– 8.12, the coverage rates are all 1s for different sample sizes while the half-widths are all 0s, demonstrating applying AV to the standard normal is a special case, where the mean estimator is identical to the theoretical mean 0 and variance estimator equals to 0 after the antithetic pairs cancel out in constructing the mean estimator and the variance estimator. For CV, the coverage rates for the confidence intervals for the mean are close to the nominal level for all values of n, demonstrating that for the sample means, the CLT asymptotics appear to take effect fairly quickly and in general more rapidly than for the quantile estimators. For IS-only and IS+SS on the SAN, coverages when estimating the mean are again close to the nominal level. However, for IS-only and IS+SS for the normal and bivariate normal, coverage for the mean intervals are poor. This may be because the importance-sampling distribution was chosen to estimate the quantile, but this distribution is possibly very inappropriate for estimating the mean, leading to coverage problems.

Finally, some of the tables present results from applying the combined estimator of ϕ_p from (4.7). We combined $r = 2$ values of c, using the strategy described at the end of Section 8.3. From Tables 8.7, 8.8, 8.27 and 8.28, it is not clear if the combining estimator outperforms the single estimator of ϕ_p in terms of displaying a faster convergence and/or a smaller half-width. “NaN" represents data not available due to two reasons below: the sum of the probability and the positive perturbed term exceeds 1; the perturbed postion is less than 1.

8.5 Program Description and Testing

8.5.1 Program Description

The empirical study is a set of simulation experiments that require extensive numerical computation. Matlab, widely used by engineers and scientists for scientific computing and prototyping as stated by Kuncicky (2004), is a natural development tool. As an interpreted programming language, however, Matlab has inevitably shown low efficiency except for matrix-heavy computations. Therefore, all the experiments were carried out on high-performance computing clusters. Only the pilot study and some test cases were conducted on personal computers.

To implement CRN effectively, the entire program package consists of three functional components: CMC/AV/CV, IS and IS+SS applying CMC, AV, CV, IS and IS+SS to both the normal/bivariate normal and the SAN experiments. In the following, each file and subprogram/subroutine, whic is a Matlab m-file, will now be described in more details.

AV/CV/CMC

- AVCVCMC.m: a main program that integrates coverages and half-widths from the SAN experiments applying AV, CV and CMC in a 1×222 vector.
- AVCVCMC2.m: a main program that integrates coverages and half-widths from the normal/bivariate normal experiments applying AV, CV and CMC in a 1×222 vector.
- batch.m: divides a column vector into a fixed number of intervals and rearrange them in a matrix for batching experiments.
- CI1.m: an m-file that computes coverages and half-widths for the experiments using finite difference to estimate $1/f(\xi_p)$ according to (4.5) and (4.6).
- CI2.m: computes coverages and half-widths for the batching experiments.
- CMCBatch.m: computes the sample mean and sample variance of the quantile estimators from batching while applying CMC.
- ComputeBeta.m: computes $\hat{\beta}_n(x)$ for the experiments while applying CV according to (7.3).
- ComputeT.m: computes T_i for the experiments while applying CV according to (7.7).

- CV.m: computes estimators of ξ_p, ϕ_p and ψ_p for the experiments applying CV and integrates them in a 5×1 vector.

- CVBatch.m: performs batching while applying CV and computes the sample mean and the sample variance of the quantile estimators from batching.

- Empirical.m: performs as an indicator function.

- FBV.m: computes the forward and the backward finite difference in general.

- GetAV.m: computes estimators of ξ_p, ϕ_p and ψ_p for the experiments applying AV and integrates them in a 5×1 vector.

- GetCMC.m: computes estimators of ξ_p, ϕ_p and ψ_p for the experiments applying CMC and integrates them in a 5×1 vector.

- GetPsi.m: computes the estimator of ψ_p for the experiments applying AV according to (6.2).

- GetQt.m: computes the estimator of ξ_p for the experiments using finite difference to estimate $1/f(\xi_p)$ according to (4.5) and (4.6).

- sampling.m: generates CRN samples for the SAN experiments applying CMC, AV and CV. More specifically, the output is a 1×3 vector with the 1st entry as the longest path from the original distribution, 2nd entry as the longest path from the distribution applying AV, and 3rd entry as a second path from the original distribution.

- Size.m: calls sampling.m to generate n CRN samples for the SAN experiments applying AV, CV and CMC, respectively and integrates them in a $1 \times 3n$ vector.

- Size2.m: integrates n samples from the normal/bivariate normal experiments applying AV, CV and CMC, respectively in a $1 \times 3n$ vector.

- SV.m: computes an intermediate result of $\mathrm{Var}[C]$. in (7.5).

- TriMax.m: returns the greatest number if inputs are three numbers; returns a vector with the greatest number in each entry if inputs are three vectors; returns a matrix with the greatest number in each entry if inputs are three matrices.

- WEIGHT.m: an m-file that assigns weights to ϕ_p estimators for combining experiments according to (4.7).

The programs for both the SAN and the normal/bivariate normal experiments have the following structure.

Function $AVCVCMC(lambda, p, b, n, m)$ includes the input $lambda$ as the rate parameter of the exponential distribution, p as the probability of interest, b as the number of batches, n as the sample size, and m as the replication times of the simulation experiment.

1. Preprocessing: Define the theoretical quantile qt of X, the theoretical quantile q of the control variate C and the theoretical mean of X; preallocate memory for multiple vectors and matrices to store intermediate and end results.

2. Use sampling.m and Size.m to generate samples for AV, CV and CMC and integrate them in a row vector.

3. Use FBV.m to produce forward and backward finite difference of p_n.

4. Use GetCMC.m, GetAV.m and CV.m to compute estimates for ξ_p, ϕ_p with centered, forward and backward finite difference and ψ_p.

5. Use CI1.m to construct the confidence intervals for experiments applying AV, CV and CMC in terms of ϕ_p estimators with different c values.

6. Use WEIGHT.m and results from Step (4) to construct the confidence intervals in terms of combining ϕ_p estimators.

7. Use CMCBatch.m, AVBatch.m and CVBatch.m to construct confidence intervals for the batching experiment.

8. Use the samples from Step (3) to construct confidence intervals for the mean estimation.

9. Repeat Step (2)–(8) m times to complete the coverage experiment.

IS: Below are brief reviews of the Matlab m-files used for IS

- batch.m and trimax.m: same as batch.m in the AV/CV/CMC component.
- Compare.m: sets the flag of sampling for stratification.
- GetAlpha.m: computes α for the SAN experiments as specified in Section 8.2.1.
- GetK.m: computes κ for the SAN experiments as specified in Section 8.2.1.
- GetL.m: computes the likelihood ratio for the SAN experiments in (8.1) as specified in Section 8.2.1.
- GetL2.m: computes the likelihood ratio for the normal/bivariate normal experiments.
- GetPsi.m: computes ψ_p estimator for the normal/bivariate normal experiments.
- GetPsi1.m: computes ψ_p estimator for the SAN experiments according to Theorem 6.
- GetQt.m: computes ξ_p estimator for the experiments while applying IS.
- GetStr.m: performs stratification on normal/bivariate normal experiments.
- GetTheta.m: computes θ as specified in Section 8.2.1.
- GetXi.m: computes ζ_j for the experiments while applying IS as specified in Section 8.2.1.
- IS.m: computes coverages and half-widths for the SAN while applying IS.

- ISN.m: computes coverages and half-widths for the normal/bivariate normal experiments while applying IS.

The program for the SAN experiments has the following structure. The program for the normal/bivariate normal experiments has a similar structure except for the sampling part.

Function $IS(p, j, b, n, r)$ includes the input p as the probability of interest, j as the number of paths of the SAN, b as the number of batches, n as the sample size and r as the replication times of the simulation experiments.

1. Preprocessing: Define the theoretical quantile tq of X and the theoretical mean of X; preallocate memory for multiple vectors and matrices to store intermediate and end results; precalculate θ_j, κ_j and α_j by using GetTheta.m, GetXi.m, GetXi.m, trimax.m, GetK.m and GetAlpha.m for applying IS.

2. Use Matlab built-in function rand() to generate uniform random samples between 0 and 1 and then compare it with $\alpha(1,1)$, $\alpha(1,1)+\alpha(2,1)$ and 1 to decide on which path IS should be applied.

3. Use trimax.m and GetL.m to compute the longest path for the SAN and the corresponding likelihood ratio.

4. Use getQt.m to compute the ξ_p estimator.

5. Use GetPsi.m to compute the ψ_p estimator.

6. Construct confidence intervals where ϕ_p are estimated by centered/forward/backward finite difference with different c values.

7. Construct confidence intervals for combining ϕ_p estimators.

8. Use batch.m and GetQt.m to construct intervals for the batching experiment.

9. Construct intervals for mean estimation.

10. Repeat (2)–(9) r times to complete the coverage experiment.

IS+SS: Below are brief reviews of the Matlab m-files used for IS+SS.

- batch.m and trimax.m: same as that in AV/CV/CMC and IS components. Compare.m, GetAlpha.m, GetK.m, GetQt.m, GetXi.m: same as those in IS component.
- batch1.m: performs batching for the experiments while applying IS+SS.
- GetL2.m: computes the likelihood ratio for the experiments while applying IS+SS as specified in Section 8.2.2.
- GetPsi1.m: computes ψ_p estimator according to Theorem 6 for the experiments while applying IS+SS.
- GetVar.m: computes the sample variance of X for mean estimation while applying IS+SS.
- strata.m: performs stratification in an equiprobable way as specified in Section 8.4.
- STR.m: computes coverages and half-widths for the SAN experiments while applying IS+SS.
- STR2.m: computes coverages and half-widths for the normal/bivariate normal experiments while applying IS+SS.

The program for the SAN experiments has the following structure. The program for the normal/bivariate normal experiments has a similar structure except for the sampling part.

Function $STR(p, j, w, n, r, b)$ includes the input p as the probability of interest, j as the number of paths of the SAN, w as the number of strata, n as the sample size in each stratum, r as the replication times of the simulation experiments and b as the number of batches.

1. Preprocessing: define the theoretical quantile tq of X and the theoretical mean of X; preallocate memory for multiple vectors and matrices to store intermediate and end results.

2. Use strata.m to set up the stratification in an equiprobable way as specified in Section 8.4.

3. Use GetTheta.m, GetXi.m, trimax.m, GetK.m and GetAlpha.m to compute θ_j, κ_j and α_j for applying IS.

4. Same as Step (2) in the IS component.

5. Use compare.m to decide if more samples are needed to fill up each stratum. More specifically, a status variable $flag = 1$ is set initially before the loop of generating samples starts. The only condition for the loop to stop is $flag \neq 1$. If all the strata have n samples, compare.m will update $flag = 0$ and then the loop stops; otherwise, the loop continues to generate more samples.

6. Use trimax.m and GetL.m to compute the longest path for the SAN and the corresponding likelihood ratio.

7. Use GetQt.m and GetPsi1.m to construct confidence intervals in terms of different c values.

8. Construct confidence intervals for the combining experiment.

9. Use batch1.m and GetQt.m to construct confidence intervals for batching.

10. Construct confidence intervals for the mean estimation.

11. Repeat Step (4) –(10) to complete the coverage experiment.

8.5.2 Program Testing

System verification and validation (V&V) activities take place at each stage of the software process from requirements reviews through code inspection. Carrying out V&V is not only to check if the software conforms to its specification both from functional and non-functional requirements, but also to ensure that the system meets the "customer's" expectations. The ultimate goal, concluded by Sommerville (2004), is to establish the confidence that the system is a good fit for its intended use.

Since these programs are designed particularly for the simulation experiments, the formal V&V processes can be simplified to program testing, which is intended to discover and fix system defects at the program stage. Two fundamental testing activities, component testing (testing parts of the program) and system testing (testing the system as a whole), have been applied to test the validity of the program. Specifically, integration testing is the most widely used approach in finding the source of problems and identifying the components in the system that need to be further debugged and analyzed.

Take the AV/CV/CMC functional component as an example. Its subprograms can be traced back to three independent programs designed to compute coverages and half-widths for the experiments while applying AV, CV and CMC, respectively. Therefore, it is appropriate to test each unit program first and then validate the integration, i.e., a mixture of *top-down* (develop the overall skeleton of the system first and then add components to it) and *bottom-up* (integrate the components that provide common services first and then add the functional components incrementally) in this case.

For a specific function in this case, two testing methods have been heavily used: comparison testing and scenario-based testing. Comparison testing is to compare the results generated by programs and derived from calculations when the results can be computed analytically. As many intermediate results of our experiments fall in this category, comparison testing has been widely used. Scenario-based testing is to reveal defects by executing programs under predetermined test cases. When results can not be computed analytically, test cases are designed to increase the chances of exposing defects.

The following three sections are devoted to explaining in more details how these two methods were used to debug the program.

AV/CV/CMC

Since it is a highly integrated program, none of its functions can be tested thoroughly by comparison testing alone.

Two functions are chosen to demonstrate how scenario-based testing is built to validate their correctness.

- batch.m: batch(b,n,A), where b, n and A denotes the number of batches, the entire sample size of A, and the column vector, respectively. The output should be a $(n/b) \times b$ matrix.

  1. Input: $b = 5, n = 10, A = [1;2;3;4;5;6;7;8;9;10]$(a column vector of size 10)

     Output:[1 3 5 7 9; 2 4 6 8 10] (a 2×5 matrix displayed after the execution of the program)

     Conclusion: True as the column vector is divided into five batches with two samples each.

  2. Input: $b = 5, n = 3, A = [1;2;3]$ (a column vector of size 3)

     Output: Warning: Size vector should be a row vector with integer elements.

     Conclusion: True as batching cannot be operated on the column vector.

  3. Input: $b = 2, n = 2, A = [12]$

     Output:Attempted to access $A(2,1)$; index out of bounds because size(A)=[1,2]).

     Conclusion: True as the row vector does not satisfy the requirements of the input.

- GetQt.m: GetQt(n,p,F), where n, p and F denotes the row size of F, the probability of interest, and an $n \times 2$ matrix generated from sampling.

  1. Scenario(input): for the SAN experiment while applying CV, set sample size n is 100 and the probability of interest p is 0.95 and the matrix F generated from the first part in CV.m: F =

     [0.5947 0.0103; 0.7233 0.0103; 0.7914 0.0103; 1.0875 0.0103; 1.1129 0.0103; 1.1472 0.0103; 1.3395 0.0103; 1.3549 0.0103; 1.3602 0.0103; 1.4233 0.0103; 1.4727 0.0103; 1.4869 0.0103; 1.7637 0.0103; 1.7975 0.0103; 1.9006 0.0103; 1.9047 0.0103; 1.9606 0.0103; 1.9654 0.0103; 2.0174 0.0103; 2.0329 0.0103; 2.0540 0.0103; 2.0584 0.0103; 2.0594 0.0103; 2.0847 0.0103; 2.1145 0.0103; 2.1371 0.0103; 2.1584 0.0103; 2.2212 0.0103; 2.2237 0.0103; 2.2827 0.0103; 2.2883 0.0103; 2.3110 0.0103; 2.3120 0.0103; 2.3149 0.0103; 2.3463 0.0103; 2.3803 0.0103; 2.3860 0.0103; 2.4133 0.0103; 2.4304 0.0103; 2.4354 0.0103; 2.4924 0.0103; 2.5104 0.0103; 2.5588 0.0103; 2.6414 0.0103; 2.6523 0.0103; 2.6529 0.0103; 2.6541 0.0103; 2.6836 0.0103; 2.7436 0.0103; 2.8588 0.0103; 2.8790 0.0103; 2.9375 0.0103; 2.9818 0.0103; 2.9895 0.0103; 3.0427 0.0103; 3.0672 0.0103; 3.1139 0.0103; 3.1380 0.0103; 3.1613 0.0103; 3.1695 0.0103; 3.1807 0.0103; 3.1994 0.0103; 3.2490 0.0103; 3.2779 0.0103; 3.2933 0.0103; 3.2955 0.0103; 3.2960 0.0103; 3.4983 0.0103; 3.6056 0.0103; 3.7155 0.0103; 3.8570 0.0103; 3.9112 0.0103; 3.9810 0.0103; 4.1180 0.0103; 4.3221 0.0103; 4.3346 0.0103; 4.4352 0.0103; 4.4661 0.0103; 4.5445 0.0103; 4.6258 0.0103; 4.7700 0.0103; 4.7998 0.0103; 4.9725 0.0103; 5.2302 0.0103; 5.3147 0.0103; 5.3652 0.0103; 5.3775 0.0103; 5.4297 0.0103; 5.5484 0.0103; 5.5588 0.0103; 6.2997 0.0063; 6.3262 0.0063; 6.3413 0.0063; 6.3924 0.0063; 6.5728 0.0063; 7.1792 0.0103; 7.4684 0.0063; 7.4874 0.0063; 7.9237 0.0103; 8.1971 0.0063;]

     Output: 6.6528

     Conclusion: True as the sum of $F(:,2)$ (values in the second column) until the one corresponding 6.5728 for the first time exceeds 0.95.

IS

Running comparison testing can validate the correctness of several functions in IS functional component: GetAlpha.m, GetK.m, GetTheta.m and GetXi.m because these corresponding quantities can be calculated analytically as specified in the IS algorithm in Section 8.2.1.

Specifically, θ is the root of (8.3), which can be computed using solve(), an Matlab built-in function to calculate the roots of equations. The result appears to be [0.7399; 0.6819; 0.7399], which can be confirmed by plugging back in the function. Then, based on the IS algorithm as specified in Section 8.2.1, κ and α can be be computed and confirmed, which are [0.0138; 0.0500; 0.0138] and [0.1776; 0.6449; 0.1776]. Up to this point, the precalculation part before the sampling and quantile/mean estimation has been confirmed to be correct.

One function is chosen to demonstrate how scenario-based testing is performed to validate correctness.

- trimax.m: trimax(a,b,c), where a, b and c denotes three matrices or vectors the same size or numbers.

  1. Input: $a = [12;34]$, $b = [11;22]$, $c = [14;24]$ (three 2×2 matrix)

     Output: [1 2; 3 4]

     Conclusion: True as each entry in the new matrix is occupied by the greatest number of three input matrices.

  2. Input: $a = [1;3;2;4]$, $b = [2;1;2;5]$, $c = [2;2;2;2]$ (three column vectors of size 4)

     Output: [2;3;2;5]

Conclusion: True as each entry of the resultant vector is occupied by the greatest number of each position from the original vectors confirmed by observation.

3. Input: $a = [12]$, $b = [2;3]$ (one row vector of size 2 and one column vector of size 2)

   Output: Matrix dimensions must agree

   Conclusion: True as the input vectors do not satisfy the requirements of the function.

IS+SS

Running comparison testing can validate correctness of several functions GetAlpha.m, GetK.m, GetTheta.m and GetXi.m in IS+SS functional component because these quantities can be calculated analytically. The results from executing the program can be confirmed by that in the last section : θ=[0.7399; 0.6819; 0.7399], κ=[0.0138; 0.0500; 0.0138], α=[0.1776; 0.6449; 0.1776].

Two functions are chosen to demonstrate how scenario-based testing are performed to debug the program.

- strata.m: strata(theta, alpha, w), where θ=[0.7399; 0.6819; 0.7399], and α=[0.1776; 0.6449; 0.1776], which are both derived from early steps for the SAN and have been confirmed to be correct. w is the number of strata.

  1. Input: $w = 5$

     Output[3.6943;5.8424;8.3344;12.0055].

     Conclusion: true as entries of the resultant vector are exactly the same as the intervals $S_i = (G^{-1}((i-1)/k), G^{-1}(i/k)]$ for $i = 1, \ldots, 5$, where G is the CDF of the stratification variable Y as specified in the first part of Section 8.4.

  2. Input: $w = 1$

Output: 0

Conclusion: true as there is only one strata, the original vector.

3. Input: $w = 10$

   Output: [2.5457; 3.6943; 4.7578; 5.8424; 7.0115; 8.3344; 9.9217; 12.0055; 15.3016]

   Conclusion: true as entries of the resultant vector are exactly the same as the intervals $S_i = (G^{-1}((i-1)/k), G^{-1}(i/k)]$ for $i = 1, \ldots, 10$, where G is the CDF of the stratification variable Y as specified in the first part of Section 8.4.

- Compare.m: compare(C,w.n) where C, w and n denotes the current status of samples in each strutum, the number of strata and the sample size in a "full" stratum as specified earlier.

1. Input: $C = [10; 2; 1]$, $w = 3$, $n = 5$

   Output: 1

   Conclusion: True as the second and the third entry of the vector are not "full" yet.

2. Input: $C = [100; 100]$, $w = 2$, $n = 100$

   Output: 0

   Conclusion: True as both entries are considered "full" comparing with the predetermined benchmark 100.

8.6 Matlab Code

8.6.1 AV/CV/CMC Package

AVBatch.m

```
{
```

```
function bc2=AVBatch(b,n,CMC,AV,p)

% First subdivide CMC samples
% AV samples into b batches.
% Then invert them to obtain quantile estimators.
B1=batch(b,n,CMC);
B2=batch(b,n,AV);
qe=zeros(b,1);
for i=1:b
     E=[B1(:,i);B2(:,i)];
     F=sort(E);
     qe(i,1)=F(ceil(2*p*n/b),1);
end
avgqe=sum(single(qe))/b;
v=0;
for i=1:b
    v=v+(qe(i,1)-avgqe)^2;
end
varqe=v/(b-1);
bc2=[avgqe;varqe];

}
```

AVCVCMC.m

```
{
function result=AVCVCMC(lambda,p,b,n,m)

% Set up
wh=WEIGHT();
result=zeros(1,222);
t=6.6645;
q=6.295793;
mean=3.4583;

COUNTER1=zeros(1,90);
COUNTER2=zeros(1,90);
COUNTER3=zeros(1,15);
COUNTER4=zeros(1,15);
COUNTER5=zeros(1,3);
COUNTER6=zeros(1,3);
COUNTER7=zeros(1,3);
COUNTER8=zeros(1,3);
```

```
U2=zeros(1,15);
A1=zeros(5,10);
A2=zeros(5,10);
A3=zeros(5,10);
B1=zeros(2,30);
B2=zeros(2,30);
B3=zeros(2,30);
C1=zeros(1,90);
C2=zeros(1,90);
D1=zeros(5,1);
D2=zeros(5,1);
D3=zeros(5,1);
E1=zeros(2,3);
E2=zeros(2,3);
E3=zeros(2,3);
G1=zeros(1,9);
G2=zeros(1,9);
H1=zeros(2,1);
H2=zeros(2,1);
H3=zeros(2,1);
I=zeros(2,3);
J1=zeros(1,3);
J2=zeros(1,3);
L1=zeros(1,3);
L2=zeros(1,3);
R=zeros(30,1);
Phi=zeros(15,1);
Qt=zeros(15,1);
Psi=zeros(15,1);
INT=zeros(15,2);

for k=1:m

% Sampling

[CMC AV CT]=Size(n,lambda);

%Quantile estimators, Phi estimators and  Psi estimators

w=1;

Y=[0.025 0.05 0.075 0.15 0.1 0.2 0.3 0.6 0.5 1];
for c2=1:10
    c=Y(1,c2);
```

```
[fd bd]=FBV(p,c,n);
A1(:,w)=GetCMC(CMC,n,p,fd,bd,c);
A2(:,w)=GetAV(CMC,AV,p,n,bd,fd,c);
A3(:,w)=CV(n,p,q,c,CMC,CT,fd,bd);
w=w+1;
end

%center diffference for CMC/AV/CV
for t=1:10
    B1(:,t)=CI1(A1(2,t),A1(5,t),n,A1(1,t),qt);
    B2(:,t)=CI1(A2(2,t),A2(5,t),n,A2(1,t),qt);
    B3(:,t)=CI1(A3(2,t),A3(5,t),n,A3(1,t),qt);
end

%forward difference for CMC/AV/CV
for t=11:20
    B1(:,t)=CI1(A1(3,t-10),A1(5,t-10),n,A1(1,t-10),qt);
    B2(:,t)=CI1(A2(3,t-10),A2(5,t-10),n,A2(1,t-10),qt);
    B3(:,t)=CI1(A3(3,t-10),A3(5,t-10),n,A3(1,t-10),qt);
end

%backward difference for CMC/AV/CV
for t=21:30
    B1(:,t)=CI1(A1(4,t-20),A1(5,t-20),n,A1(1,t-20),qt);
    B2(:,t)=CI1(A2(4,t-20),A2(5,t-20),n,A2(1,t-20),qt);
    B3(:,t)=CI1(A3(4,t-20),A3(5,t-20),n,A3(1,t-20),qt);
end

%C1:half width, C2:Counter
%C1(1,1)--C1(1,10):CMC/CD/c-value
%C1(1,11)--C1(1,20):CMC/FD/c-value
%C1(1,21)--C1(1,30):CMC/BD/c-value
%C1(1,31)--C1(1,40):AV/CD/c-value
%C1(1,41)--C1(1,50):AV/FD/c-value
%C1(1,51)--C1(1,60):AV/BD/c-value
%C1(1,61)--C1(1,70):CV/CD/c-value
%C1(1,71)--C1(1,80):CV/FD/c-value
%C1(1,81)--C1(1,90):CV/BD/c-value

C1=[B1(1,:) B2(1,:) B3(1,:)];
C2=[B1(2,:) B2(2,:) B3(2,:)];

COUNTER1=COUNTER1+C1;
COUNTER2=COUNTER2+C2;
```

```
%Combined c-value
%
R(1:10,1)=A1(2,:)'.*wh;
R(11:20,1)=A2(2,:)'.*wh;
R(21:30,1)=A3(2,:)'.*wh;

y=1;
for x=1:15
    Phi(x,1)=R(y,1)+R(y+1,1);
    y=y+2;
end

Qt1=A1(1,1);
Qt2=A2(1,1);
Qt3=A3(1,1);
Psi1=A1(5,1);
Psi2=A2(5,1);
Psi3=A3(5,1);

%Combined c-value
%G1:half width; G2:Counter
%G1(1,1)--G1(1,3) CD/FD/BD for CMC
%G1(1,4)--G1(1,6) CD/FD/BD for AV
%G1(1,7)--G1(1,9) CD/FD/BD for CV
G1=[1.645*Phi(1:5,1)*Psi1/n;1.645*Phi(6:10,1)*Psi2/n;
1.645*Phi(11:15,1)*Psi3/n];
HB=[Qt1+G1(1:5,1);Qt2+G1(6:10,1);Qt3+G1(11:15,1)];
LB=[Qt1-G1(1:5,1);Qt2-G1(6:10,1);Qt3-G1(11:15,1)];
U1=G1';
for z=1:15
    if qt>=LB(z,1) && qt<=HB(z,1)
        U2(1,z)=1;
    end
end

COUNTER3=COUNTER3+U1;
COUNTER4=COUNTER4+U2;

%Setup for Batching, Mean estimation and Combined c

H1=CMCBatch(b,n,p,CMC);

H2=AVBatch(b,n,CMC,AV,p);
```

```
H3=CVBatch(b,n,CMC,CT,p,q);

I=[H1 H2 H3];

J1=1.833*sqrt(I(2,:))/sqrt(b);
hbound1=I(1,1)+J1(1,1);
lbound1=I(1,1)-J1(1,1);
hbound2=I(1,2)+J1(1,2);
lbound2=I(1,2)-J1(1,2);
hbound3=I(1,3)+J1(1,3);
lbound3=I(1,3)-J1(1,3);

if qt<hbound1 && qt>lbound1
    R1=1;
else
    R1=0;
end

if qt<hbound2 && qt>lbound2
    R2=1;
else
    R2=0;
end

if qt<hbound3 && qt>lbound3
    R3=1;
else
    R3=0;
end

J2=[R1 R2 R3];

COUNTER5=COUNTER5+J1;
COUNTER6=COUNTER6+J2;

%result3=Mean(CMC,AV,CT,n);

%Construct Confidence Interval
avg1=sum(single(CMC))/n;
avg2=sum((CMC+AV)/2)/n;
beta=corr(CMC,CT)*sqrt(var(CMC)*var(CT))/var(CT);
avg3=sum(CMC-beta*(CT-3))/n;
var1=var(CMC);
var2=var((CMC+AV)/2);
var3=var(CMC-beta*(CT-3));
```

```
hw1=1.645*sqrt(var1)/sqrt(n);
hw2=1.645*sqrt(var2)/sqrt(n);
hw3=1.645*sqrt(var3)/sqrt(n);
hb1=avg1+hw1;
lb1=avg1-hw1;
hb2=avg2+hw2;
lb2=avg2-hw2;
hb3=avg3+hw3;
lb3=avg3-hw3;

if mean<hb1 && mean>lb1;
    K1=1;
else
    K1=0;
end

if mean<hb2 && mean>lb2;
    K2=1;
else
    K2=0;
end

if mean<hb3 && mean>lb3;
    K3=1;
else
    K3=0;
end

L1(1,:)=[hw1 hw2 hw3];
L2(1,:)=[K1 K2 K3];

COUNTER7=COUNTER7+L1;
COUNTER8=COUNTER8+L2;

end

result=[COUNTER1/m COUNTER2/m COUNTER3/m COUNTER4/m
COUNTER5/m COUNTER6/m COUNTER7/m COUNTER8/m];

}
```

AVCVCMC2.m

```
{

function result=AVCVCMC2(p,b,n,m,r)

% Set up
wh=WEIGHT();
result=zeros(1,232);
qt=norminv(p,0,1);
q=norminv(p,0,1);
mean=0;

COUNTER1=zeros(1,90);
COUNTER2=zeros(1,90);
COUNTER3=zeros(1,15);
COUNTER4=zeros(1,15);
COUNTER5=zeros(1,3);
COUNTER6=zeros(1,3);
COUNTER7=zeros(1,3);
COUNTER8=zeros(1,3);
U2=zeros(1,15);
A1=zeros(5,10);
A2=zeros(5,10);
A3=zeros(5,10);
B1=zeros(2,30);
B2=zeros(2,30);
B3=zeros(2,30);
C1=zeros(1,90);
C2=zeros(1,90);
D1=zeros(5,1);
D2=zeros(5,1);
D3=zeros(5,1);
E1=zeros(2,3);
E2=zeros(2,3);
E3=zeros(2,3);
G1=zeros(1,9);
G2=zeros(1,9);
H1=zeros(2,1);
H2=zeros(2,1);
H3=zeros(2,1);
I=zeros(2,3);
J1=zeros(1,3);
J2=zeros(1,3);
L1=zeros(1,3);
L2=zeros(1,3);
R=zeros(30,1);
```

```
Phi=zeros(15,1);
Qt=zeros(15,1);
Psi=zeros(15,1);
INT=zeros(15,2);

for k=1:m

% Sampling
result=Size2(n,r);
CMC=result(1:n,1);
AV=result(n+1:2*n,1);
CT=result(2*n+1:3*n,1);

%Quantile estimators, Phi estimators and  Psi estimators

w=1;
Y=[0.025 0.05 0.075 0.15 0.1 0.2 0.3 0.6 0.5 1];
for c2=1:10
    c=Y(1,c2);

[fd bd]=FBV(p,c,n);
A1(:,w)=GetCMC(CMC,n,p,fd,bd,c);
A2(:,w)=GetAV(CMC,AV,p,n,bd,fd,c);
A3(:,w)=CV(n,p,q,c,CMC,CT,fd,bd);
w=w+1;
end

%center diffference for CMC/AV/CV
for t=1:10
    B1(:,t)=CI1(A1(2,t),A1(5,t),n,A1(1,t),qt);
    B2(:,t)=CI1(A2(2,t),A2(5,t),n,A2(1,t),qt);
    B3(:,t)=CI1(A3(2,t),A3(5,t),n,A3(1,t),qt);
end

%forward difference for CMC/AV/CV
for t=11:20
    B1(:,t)=CI1(A1(3,t-10),A1(5,t-10),n,A1(1,t-10),qt);
    B2(:,t)=CI1(A2(3,t-10),A2(5,t-10),n,A2(1,t-10),qt);
    B3(:,t)=CI1(A3(3,t-10),A3(5,t-10),n,A3(1,t-10),qt);
end

%backward difference for CMC/AV/CV
for t=21:30
    B1(:,t)=CI1(A1(4,t-20),A1(5,t-20),n,A1(1,t-20),qt);
    B2(:,t)=CI1(A2(4,t-20),A2(5,t-20),n,A2(1,t-20),qt);
```

```
    B3(:,t)=CI1(A3(4,t-20),A3(5,t-20),n,A3(1,t-20),qt);
end

%C1:half width, C2:Counter
%C1(1,1)--C1(1,10):CMC/CD/c-value
%C1(1,11)--C1(1,20):CMC/FD/c-value
%C1(1,21)--C1(1,30):CMC/BD/c-value
%C1(1,31)--C1(1,40):AV/CD/c-value
%C1(1,41)--C1(1,50):AV/FD/c-value
%C1(1,51)--C1(1,60):AV/BD/c-value
%C1(1,61)--C1(1,70):CV/CD/c-value
%C1(1,71)--C1(1,80):CV/FD/c-value
%C1(1,81)--C1(1,90):CV/BD/c-value

C1=[B1(1,:) B2(1,:) B3(1,:)];
C2=[B1(2,:) B2(2,:) B3(2,:)];
COUNTER1=COUNTER1+C1;
COUNTER2=COUNTER2+C2;

%Combined c-value
%
R(1:10,1)=A1(2,:)'.*wh;
R(11:20,1)=A2(2,:)'.*wh;
R(21:30,1)=A3(2,:)'.*wh;
y=1;
for x=1:15
    Phi(x,1)=R(y,1)+R(y+1,1);
    y=y+2;
end

Qt1=A1(1,1);
Qt2=A2(1,1);
Qt3=A3(1,1);
Psi1=A1(5,1);
Psi2=A2(5,1);
Psi3=A3(5,1);

%Combined c-value
%G1:half width; G2:Counter
%G1(1,1)--G1(1,3) CD/FD/BD for CMC
%G1(1,4)--G1(1,6) CD/FD/BD for AV
%G1(1,7)--G1(1,9) CD/FD/BD for CV
G1=[1.645*Phi(1:5,1)*Psi1/n;1.645*Phi(6:10,1)*Psi2/n;
1.645*Phi(11:15,1)*Psi3/n];
HB=[Qt1+G1(1:5,1);Qt2+G1(6:10,1);Qt3+G1(11:15,1)];
```

```
LB=[Qt1-G1(1:5,1);Qt2-G1(6:10,1);Qt3-G1(11:15,1)];
U1=G1';
for z=1:15
    if qt>=LB(z,1) && qt<=HB(z,1)
        U2(1,z)=1;
    end
end

COUNTER3=COUNTER3+U1;
COUNTER4=COUNTER4+U2;

%Setup for Batching, Mean estimation and Combined c

H1=CMCBatch(b,n,p,CMC);
H2=AVBatch(b,n,CMC,AV,p);
H3=CVBatch(b,n,CMC,CT,p,q);
I=[H1 H2 H3];
J1=1.833*sqrt(I(2,:))/sqrt(b);
hbound1=I(1,1)+J1(1,1);
lbound1=I(1,1)-J1(1,1);
hbound2=I(1,2)+J1(1,2);
lbound2=I(1,2)-J1(1,2);
hbound3=I(1,3)+J1(1,3);
lbound3=I(1,3)-J1(1,3);

if qt<=hbound1 && qt>=lbound1
    R1=1;
else
    R1=0;
end

if qt<=hbound2 && qt>=lbound2
    R2=1;
else
    R2=0;
end

if qt<=hbound3 && qt>=lbound3
    R3=1;
else
    R3=0;
end

J2=[R1 R2 R3];
COUNTER5=COUNTER5+J1;
```

```
COUNTER6=COUNTER6+J2;

%result3=Mean(CMC,AV,CT,n);
%Construct Confidence Interval
avg1=sum(single(CMC))/n;
avg2=sum((CMC+AV)/2)/n;
beta=corr(CMC,CT)/var(CT);
avg3=sum(CMC-beta*(CT-mean))/n;
var1=var(CMC);
var2=var((CMC+AV)/2);
var3=var(CMC-beta*(CT-mean));
hw1=1.645*sqrt(var1)/sqrt(n);
hw2=1.645*sqrt(var2)/sqrt(2*n);
hw3=1.645*sqrt(var3)/sqrt(n);
hb1=avg1+hw1;
lb1=avg1-hw1;
hb2=avg2+hw2;
lb2=avg2-hw2;
hb3=avg3+hw3;
lb3=avg3-hw3;

if mean<=hb1 && mean>=lb1;
    K1=1;
else
    K1=0;
end
if mean<=hb2 && mean>=lb2;
    K2=1;
else
    K2=0;
end
if mean<=hb3 && mean>=lb3;
    K3=1;
else
    K3=0;
end

L1(1,:)=[hw1 hw2 hw3];
L2(1,:)=[K1 K2 K3];

COUNTER7=COUNTER7+L1;
COUNTER8=COUNTER8+L2;
end
result=[COUNTER1/m COUNTER2/m COUNTER3/m COUNTER4/m
COUNTER5/m COUNTER6/m COUNTER7/m COUNTER8/m];
```

```
}
```

batch.m

```
{
function B=batch(b,n,A)
% Subdivide a vector of size n into
% b batches and place them into
% a n/b*b matrix.
B=zeros(n/b,b);
k=1;
for j=1:b
    for i=1:(n/b)
    B(i,j)=A(k,1);
    k=k+1;
    end
end
}
```

CI1.m

```
{
function A=CI1(phi,psi,n,qe,qt)
% the confidence interval setup
% due to the central limit theorem
% of a normal distribution.
hw=1.645*phi.*psi/sqrt(n);
lb=qe-hw;
hb=qe+hw;
s=0;
if qt<=hb && qt>=lb
    s=1;
end
A=[hw;s];

}
```

CI2.m

```
{
function B=CI2(var,b,qe,qt)
```

```
% the confidence interval setup
% for the student-t distribution.
hw=1.833*sqrt(var)/sqrt(b);
hb=qe+hw;
lb=qe-hw;
s=0;
if qt<=hb && qt>=lb
    s=1;
end
B=[hw;s];

}
```

CMCBatch.m

```
{
function bc=CMCBatch(b,n,p,CMC)
% Compute the quantiles
% and variance for
% CMC batchs.
B=batch(b,n,CMC);
qe=zeros(b,1);
for i=1:b
   N=sort(B(:,i));
   qe(i,1)=N(ceil(n*p/b),1);
end
avgqe=sum(single(qe))/b;
varqe=var(qe);
bc=[avgqe;varqe];
}
```

ComputeBeta.m

```
{
function beta=ComputeBeta(CMC,CT,n)
%Compute Beta
avg1=sum(single(CMC))/n;
avg2=sum(single(CT))/n;
s=0;
t=0;
for i=1:n
    s=s+(CMC(i,1)-avg1)*(CT(i,1)-avg2);
    t=t+(CT(i,1)-avg2)^2;
end
```

```
beta=s/t;

}
```

ComputeT.m

```
{
function T= ComputeT(n,p,q,avg,CT,svar)
%Compute T
T=zeros(n,1);
for i=1:n
    if CT(i,1)<q
        T(i,1)=1/n+(avg-1)*(avg-p)/svar;
    elseif CT(i,1)>q
        T(i,1)=1/n+avg*(avg-p)/svar;
    end
end

}
```

CV.m

```
{
function control=CV(n,p,q,c,CMC,CT,fd,bd)
%Set up for CV
%compute estimators for quantiles and variance constant
D=Empirical(CT,q,n);
avg=sum(single(D))/n;
svar=SV(n,D,avg);
%var(D);

%Compute T
T=ComputeT(n,p,q,avg,CT,svar);
%1/m+(avg-1)*(avg-p)/svar
%Order has been given before all the results.
%T=ComputeT2(p,s,D,m)

% Compute i_q
E=[CMC T];
F=sortrows(E);

% Quantile estimator
qe=GetQt(n,p,F);
```

```
% Compute theoretical Phi
%Phi=density(q);

% Estimate Phi
q1=GetQt(n,fd,F);
q2=GetQt(n,bd,F);
phi1=sqrt(n)*(q1-q2)/(2*c);
phi2=sqrt(n)*(q1-qe)/c;
phi3=sqrt(n)*(qe-q2)/c;

%Phi1=EstPhi(p,m,r,fv,bv,F)

%Compute theoretical beta and Psi
%Beta=GetBeta(p,q,mu,SIGMA);
%Psi=sqrt(PsiTh(p,q,mu,SIGMA))

%Estimate beta and Psi
Q=Empirical(CMC,qe,n);
Z=cov(Q(:,1),D(:,1));
Beta1=Z(1,2)/var(D(:,1));
psi=sqrt(var(Q(:,1)-Beta1*D));

control=[qe;phi1;phi2;phi3;psi];

}
```

CVBatch.m

```
{

function bc3=CVBatch(b,n,CMC,CT,p,q)
%subdivide CV samples of size n into b batchs
%each batch contains n/b samples
B1=batch(b,n,CMC);
B2=batch(b,n,CT);
qe=zeros(b,1);

for i=1:b
D=Empirical(B2(:,i),q,n/b);
avg=sum(single(D))*b/n;
svar=SV(n/b,D,avg);
%var(D);
```

```
%Compute T
T=ComputeT(n/b,p,q,avg,B2(:,i),svar);
%1/m+(avg-1)*(avg-p)/svar
%Order has been given before all the results.
%T=ComputeT2(p,s,D,m)

% Compute i_q
E=[B1(:,i) T];
F=sortrows(E);


% Quantile estimator
qe(i,1)=GetQt(n/b,p,F);

end

avgqe=sum(single(qe))/b;

v=0;
for i=1:b
    v=v+(qe(i,1)-avgqe)^2;
end
varqe=v/(b-1);

bc3=[avgqe;varqe];
}
```

Empirical.m

```
{
function Y=Empirical(CT,q,n)
% Indicator function
Y=zeros(n,1);
for i=1:n
    if CT(i,1)<q
        Y(i,1)=1;
    else
        Y(i,1)=0;
    end
end
}
```

FBV.m

```
{
function [fv bv]=FBV(p,c,n)

%Forward and backward variable
%n is sample size
if p+c/sqrt(n)>1
fv=1;
else
    fv=p+c/sqrt(n);
end
bv=p-c/sqrt(n);

}
```

GetAV.m

```
{
function anti=GetAV(CMC,AV,p,n,bd,fd,c)

% Quantile Estimator
E=[CMC;AV];
F=sort(E);
qe=F(ceil(2*p*n),1);

%Compute Phi estimator
%forward backward anc central
q1=F(ceil(2*bd*n),1);
q2=F(ceil(2*fd*n),1);
phi1=sqrt(n)*(q2-q1)/(2*c);
phi2=sqrt(n)*(q2-qe)/c;
phi3=sqrt(n)*(qe-q1)/c;

%Compute Psi estimator
psi= GetPsi(CMC,AV,qe,p,n);
anti=[qe;phi1;phi2;phi3;psi];
}
```

GetCMC.m

```
{
function crude=GetCMC(CMC,n,p,fd,bd,c)
```

```
%Quantile Estimator
N=sort(CMC);
qe=N(ceil(n*p),1);

%Estimate Phi
phi1=sqrt(n)*(N(ceil(n*fd),1)-N(ceil(n*bd),1))/(2*c);
phi2=sqrt(n)*(N(ceil(n*fd),1)-N(ceil(n*p),1))/c;
phi3=sqrt(n)*(N(ceil(n*p),1)-N(ceil(n*bd),1))/c;

%Estimate Psi
psi=sqrt(p*(1-p));

crude=[qe;phi1;phi2;phi3;psi];

}
```

GetPsi.m

```
{
function psi= GetPsi(CMC,AV,qe,p,n)
%Compute variance constant for AV
s=0;
for i=1:n
    if CMC(i,1)<qe && AV(i,1)<qe
        s=s+1;
    end
end

psi=sqrt((p*(1-2*p)+s/n)/2);

}
```

GetQt.m

```
{
function Qt= GetQt(n,p,F)
%Compute the quantile estimator
i=1;
S=F(1,2);
while i<n && S<p
    i=i+1;
```

```
    S=S+F(i,2);
end
Qt=F(i,1);

}
```

sampling.m

```
{
function [X Y C]=sampling(lambda)
%Generate samples for SAN
%using common random numbers
U=zeros(5,1);
A=zeros(5,1);
B=zeros(5,1);
T=zeros(3,1);
S=zeros(3,1);

for i=1:5
    U(i,1)=rand();
    A(i,1)=-log(1-U(i,1))/lambda;
    B(i,1)=-log(U(i,1))/lambda;
end

T(1,1)=A(1,1)+A(2,1);
T(2,1)=A(1,1)+A(3,1)+A(5,1);
T(3,1)=A(4,1)+A(5,1);
S(1,1)=B(1,1)+B(2,1);
S(2,1)=B(1,1)+B(3,1)+B(5,1);
S(3,1)=B(4,1)+B(5,1);
X=TriMax(T(1,1),T(2,1),T(3,1));
Y=TriMax(S(1,1),S(2,1),S(3,1));
C=T(2,1);
}
```

Size.m

```
{
function [CMC AV CT]=Size(n,lambda)
% Generate n samples for
% CMC, AV and CT (control variate)
```

```
CMC=zeros(n,1);
AV=zeros(n,1);
CT=zeros(n,1);
for i=1:n
    [CMC(i,1),AV(i,1),CT(i,1)]=sampling(lambda);
end

}
```

Size2.m

```
{

function result=Size2(n,r)
% Generate CRN (common random numbers)
% samples for CMC, AV and CT (control variates)
CMC=zeros(n,1);
AV=zeros(n,1);
CT=zeros(n,1);
MU=[0 0];
SIGMA=[1 r;r 1];
result=zeros(3*n,1);

A=mvnrnd(MU,SIGMA,n);
CMC=A(:,1);
CT=A(:,2);
AV=-CMC;
result=[CMC;AV;CT];
}
```

SV.m

```
{
function svar=SV(n,D,avg)
% intermediate step to compute variance
svar=0;
for j=1:n
    svar=svar+(D(j,1)-avg)^2;
end

}
```

Trimax.m

```
{
function Y=TriMax(A,B,C)
% compare three vectors/matrix/numbers
X=max(A,B);
Y=max(X,C);
end

}
```

8.6.2 IS Package and IS+SS Package

GetAlpha.m

```
{
function alpha=GetAlpha(K,j)
% Compute alpha
alpha=zeros(j,1);
for i=1:j
alpha(i,1)=K(i,1)/sum(single(K));
end
}
```

GetK.m

```
{
function K= GetK(theta,MaxXi,m,j)
%compute K
K=zeros(j,1);
for i=1:j
K(i,1)=(1-theta(i,1))^(-m(i,1))*exp(-theta(i,1)*MaxXi);
end
end
}
```

GetL.m

```
{
function LR= GetL(theta,alpha,j,m,T,N)
%Compute likelihood ratio
S=alpha(1,1)*exp(theta(1,1)*T(1,N)+m(1,1)*log(1-theta(1,1)))
```

```
+alpha(2,1)*exp(theta(2,1)*T(2,N)+m(2,1)*log(1-theta(2,1)))
+alpha(3,1)*exp(theta(3,1)*T(3,N)+m(3,1)*log(1-theta(3,1)));
LR=1/S;
end

}
```

GetPsi.m

```
{
function w=GetPsi(LR,M,n,p,Qt)
% Compute psi estimator
s=0;
for k=1:n
   if M(k,1)>Qt
       s=s+LR(k,1)^2;
   end;
end
w=sqrt(s/n-(1-p)^2);

}
```

GetPsi1.m

```
{
function Psi=GetPsi1(O,P,w,n,Qt)
% compute psi estimator
Psi2=0;
Xi2=zeros(w,1);
lambda=1/w;
gamma=1/w;

for j=1:w
S=0;
T=0;
for i=1:n
    if O(i,j)>Qt
    S=S+P(i,j)^2;
    T=T+P(i,j);
    end
end
Xi2(j,1)=S/n-(T/n)^2;
Psi2=Psi2+Xi2(j,1)*(lambda^2)/gamma;
```

```
end
Psi=sqrt(Psi2);
}
```

GetQt.m

```
{
function Qt=GetQt(M,LR,p,n)
% Compute quantile estimator
A=[M LR];
B=sortrows(A);
S=B(n,2);
v=n;
while v>1 && S<(1-p)*n
    v=v-1;
    S=S+B(v,2);
end
Qt=B(v,1);
}
```

GetXi.m

```
{
function v=GetXi(m,theta,j)
%compute xi
v=zeros(j,1);
%for i=1:j
%    v(i,1)=m(i,1)/(1-theta(i,1));
%end

v=m./(1-theta);
}
```

IS.m

```
{

function result=IS(p,j,b,n,r)
%main function for IS for SAN
% Set up and preprocessing
tq=6.6645;%theoretical value
```

```
mean=3.4583;%theoretical value
result=zeros(1,78);
H=zeros(1,32);
I=zeros(1,32);

for q=1:r
N=0;
m=[2;3;2];
%theta=GetTheta(m,p,j)
theta=[0.7399;0.6819;0.7399];
v=GetXi(m,theta,j);
MaxXi=trimax(v(1,1),v(2,1),v(3,1));
K=GetK(theta,MaxXi,m,j);
alpha=GetAlpha(K,j);
% Sampling
M=zeros(n,1);
T=zeros(3,n);
LR=zeros(n,1);
A=zeros(5,1);
while N<n
    N=N+1;
    UN=rand();
    if UN<alpha(1,1)
        A(1,1)=exprnd(1/(1-theta(1,1)));
        A(2,1)=exprnd(1/(1-theta(1,1)));
        A(3,1)=exprnd(1);
        A(4,1)=exprnd(1);
        A(5,1)=exprnd(1);
    elseif alpha(1,1)<UN && UN<(alpha(1,1)+alpha(2,1))
        A(1,1)=exprnd(1/(1-theta(2,1)));
        A(3,1)=exprnd(1/(1-theta(2,1)));
        A(5,1)=exprnd(1/(1-theta(2,1)));
        A(2,1)=exprnd(1);
        A(4,1)=exprnd(1);
    elseif (alpha(1,1)+alpha(2,1))<UN && UN<1
        A(4,1)=exprnd(1/(1-theta(3,1)));
        A(5,1)=exprnd(1/(1-theta(3,1)));
        A(1,1)=exprnd(1);
        A(2,1)=exprnd(1);
        A(3,1)=exprnd(1);
    end
    T(2,N)=A(1,1)+A(3,1)+A(5,1);
    T(3,N)=A(4,1)+A(5,1);
    T(1,N)=A(1,1)+A(2,1);
    M(N,1)=trimax(T(1,N),T(2,N),T(3,N));
```

```
   LR(N,1)=GetL(theta,alpha,j,m,T,N);
end
%L=sum(single(LR))/n
Qt=GetQt(M,LR,p,n);

%estimate phi
Y=[0.05 0.1 0.25 0.5 0.75 1.5 1 2];
for o=1:8
    c1=Y(1,o);
fd=p+c1/sqrt(n);
bd=p-c1/sqrt(n);
q1=GetQt(M,LR,fd,n);
q2=GetQt(M,LR,bd,n);
Phi=26.5396;
Phi1(o,1)=sqrt(n)*(q1-q2)/(2*c1);
Phi2=sqrt(n)*(q1-Qt)/c1;
Phi3=sqrt(n)*(Qt-q2)/c1;

%estimate psi
Psi=GetPsi(LR,M,n,p,Qt);

%Construct Confidence Interval
hw(q,o)=1.645*Phi*Psi/sqrt(n);
hb=Qt+hw(q,o);
lb=Qt-hw(q,o);

% Construct confidence intervals
hw1(q,o)=1.645*Phi1(o,1)*Psi/sqrt(n);
hb1=Qt+hw1(q,o);
lb1=Qt-hw1(q,o);
hw2(q,o)=1.645*Phi2*Psi/sqrt(n);
hb2=Qt+hw2(q,o);
lb2=Qt-hw2(q,o);
hw3(q,o)=1.645*Phi3*Psi/sqrt(n);
hb3=Qt+hw3(q,o);
lb3=Qt-hw3(q,o);

if tq>lb && tq<hb
    U(q,o)=1;
else
    U(q,o)=0;
end


if tq>lb1 && tq<hb1
```

```
    U1(q,o)=1;
else
    U1(q,o)=0;
end

if tq>lb2 && tq<hb2
    U2(q,o)=1;
else
    U2(q,o)=0;
end

if tq>lb3 && tq<hb3
    U3(q,o)=1;
else
    U3(q,o)=0;
end

end
H=H+[U(q,:) U1(q,:) U2(q,:) U3(q,:)];
I=I+[hw(q,:) hw1(q,:) hw2(q,:) hw3(q,:)];

%Combined
Phi6=Phi1(1,1)*4/3-Phi1(2,1)/3;
Phi7=Phi1(3,1)*4/3-Phi1(4,1)/3;
Phi8=Phi1(5,1)*4/3-Phi1(6,1)/3;
Phi9=Phi1(7,1)*4/3-Phi1(8,1)/3;
Phi10=(Phi6+Phi7+Phi8+Phi9)/4;

hw6(q,1)=1.645*Phi6*Psi/sqrt(n);
hb6=Qt+hw6(q,1);
lb6=Qt-hw6(q,1);
hw7(q,1)=1.645*Phi7*Psi/sqrt(n);
hb7=Qt+hw7(q,1);
lb7=Qt-hw7(q,1);
hw8(q,1)=1.645*Phi8*Psi/sqrt(n);
hb8=Qt+hw8(q,1);
lb8=Qt-hw8(q,1);
hw9(q,1)=1.645*Phi9*Psi/sqrt(n);
hb9=Qt+hw9(q,1);
lb9=Qt-hw9(q,1);
hw10(q,1)=1.645*Phi10*Psi/sqrt(n);
hb10=Qt+hw10(q,1);
lb10=Qt-hw10(q,1);

if tq>lb6 && tq<hb6
```

```
    U6(q,1)=1;
else
    U6(q,1)=0;
end
if tq>lb7 && tq<hb7
    U7(q,1)=1;
else
    U7(q,1)=0;
end
if tq>lb8 && tq<hb8
    U8(q,1)=1;
else
    U8(q,1)=0;
end
if tq>lb9 && tq<hb9
    U9(q,1)=1;
else
    U9(q,1)=0;
end
if tq>lb10 && tq<hb10
    U10(q,1)=1;
else
    U10(q,1)=0;
end

%Batching
Qe2=zeros(b,1);
Bth=batch(b,n,M(:,1));
Lth=batch(b,n,LR(:,1));
for o=1:b
    Qe2(o,1)=GetQt(Bth(:,o),Lth(:,o),p,n/b);
end
avg=sum(single(Qe2))/b;
svar=var(Qe2);
hw4(q,1)=1.833*sqrt(svar)/sqrt(b);
hb4(q,1)=avg+hw4(q,1);
lb4(q,1)=avg-hw4(q,1);
if tq>lb4(q,1) && tq<hb4(q,1)
    U4(q,1)=1;
else
    U4(q,1)=0;
end

%Mean estimation
mavg=sum(M(:,1).*LR(:,1))/n;
```

```
mvar=var(M(:,1).*LR(:,1));
hw5(q,1)=1.645*sqrt(mvar)/sqrt(n);
hb5(q,1)=mavg+hw5(q,1);
lb5(q,1)=mavg-hw5(q,1);
if mean<hb5(q,1) && mean>lb5(q,1)
    U5(q,1)=1;
else
    U5(q,1)=0;
end
end
L=H/r;
Q=I/r;
D=[sum(U4)/r sum(U5)/r sum(U6)/r sum(U7)/r
sum(U8)/r sum(U9)/r sum(U10)/r];
E=[sum(hw4)/r sum(hw5)/r sum(hw6)/r sum(hw7)/r
sum(hw8)/r sum(hw9)/r sum(hw10)/r];
result=[L Q D E];
}
```

trimax.m

```
{
function d= trimax(a,b,c)
%compare three vectors/matrix/numbers
d=a;
if a<b
    d=b;
end
if d<c
    d=c;
end
}
```

GetL2.m

```
{
function LR=GetL2(theta,X,mu,sigma)
% compute likelihood ratio
LR=exp(-theta*X+mu*theta+sigma^2*theta^2/2);

}
```

GetTheta.m

```
{
function result=GetTheta(mu,sigma,p)
%Compute theta
syms t;
g=-t*(mu+sigma^2*t)+mu*t+(sigma^2)*(t^2)/2-log(1-p);
result=abs(solve(g));
}
```

ISN.m

```
{
function result=ISN(p,mu,sigma,b,n,r)
%IS for normal distribution
tq=norminv(p,mu,sigma);%0.6 quantile
%tq=norminv(0.9,mu1,sigma1); %0.9 quantile
mean=0;%theoretical value
H=zeros(1,32);
I=zeros(1,32);
result=zeros(1,78);
for q=1:r
N=0;
theta=1.7941;
%theta=1.3537;%0.6 quantile
%theta=2.1460;% 0.9 quantile
M=zeros(n,1);
LR=zeros(n,1);
while N<n
   N=N+1;
   M(N,1)=normrnd(mu+sigma^2*theta,sigma);
   LR(N,1)=GetL2(theta,M(N,1),mu,sigma);
end
%L=sum(single(LR))/n
Qt=GetQt(M,LR,p,n);
Y=[0.05 0.1 0.25 0.5 1 2 4 8];
for z=1:8
    c=Y(1,z);
fd=p+c/sqrt(n);
bd=p-c/sqrt(n);
%estimate phi
```

```
q1=GetQt(M,LR,fd,n);
q2=GetQt(M,LR,bd,n);
Phi1(z,1)=sqrt(n)*(q1-q2)/(2*c);
Phi2=sqrt(n)*(q1-Qt)/c;
Phi3=sqrt(n)*(Qt-q2)/c;
Phi=3.5719;%1/f(\xi_p)for p=0.8
%estimate psi
Psi=GetPsi(LR,M,n,p,Qt);
%Construct Confidence Interval
hw(q,z)=1.645*Phi*Psi/sqrt(n);
hb=Qt+hw(q,z);
lb=Qt-hw(q,z);
if tq>lb && tq<hb
    U(q,z)=1;
else
    U(q,z)=0;
end
hw1(q,z)=1.645*Phi1(z,1)*Psi/sqrt(n);
hb1=Qt+hw1(q,z);
lb1=Qt-hw1(q,z);
if tq>lb1 && tq<hb1
    U1(q,z)=1;
else
    U1(q,z)=0;
end
hw2(q,z)=1.645*Phi2*Psi/sqrt(n);
hb2=Qt+hw2(q,z);
lb2=Qt-hw2(q,z);
if tq>lb2 && tq<hb2
    U2(q,z)=1;
else
    U2(q,z)=0;
end

hw3(q,z)=1.645*Phi3*Psi/sqrt(n);
hb3=Qt+hw3(q,z);
lb3=Qt-hw3(q,z);
if tq>lb3 && tq<hb3
    U3(q,z)=1;
else
    U3(q,z)=0;
end
end
H=H+[U(q,:) U1(q,:) U2(q,:) U3(q,:)];
I=I+[hw(q,:) hw1(q,:) hw2(q,:) hw3(q,:)];
```

```
%Combined
Phi6=Phi1(1,1)*4/3-Phi1(2,1)/3;
Phi7=Phi1(3,1)*4/3-Phi1(4,1)/3;
Phi8=Phi1(5,1)*4/3-Phi1(6,1)/3;
Phi9=Phi1(7,1)*4/3-Phi1(8,1)/3;
Phi10=(Phi6+Phi7+Phi8+Phi9)/4;

hw6(q,1)=1.645*Phi6*Psi/sqrt(n);
hb6=Qt+hw6(q,1);
lb6=Qt-hw6(q,1);
hw7(q,1)=1.645*Phi7*Psi/sqrt(n);
hb7=Qt+hw7(q,1);
lb7=Qt-hw7(q,1);
hw8(q,1)=1.645*Phi8*Psi/sqrt(n);
hb8=Qt+hw8(q,1);
lb8=Qt-hw8(q,1);
hw9(q,1)=1.645*Phi9*Psi/sqrt(n);
hb9=Qt+hw9(q,1);
lb9=Qt-hw9(q,1);
hw10(q,1)=1.645*Phi10*Psi/sqrt(n);
hb10=Qt+hw10(q,1);
lb10=Qt-hw10(q,1);

if tq>lb6 && tq<hb6
    U6(q,1)=1;
else
    U6(q,1)=0;
end
if tq>lb7 && tq<hb7
    U7(q,1)=1;
else
    U7(q,1)=0;
end
if tq>lb8 && tq<hb8
    U8(q,1)=1;
else
    U8(q,1)=0;
end
if tq>lb9 && tq<hb9
    U9(q,1)=1;
else
    U9(q,1)=0;
end
if tq>lb10 && tq<hb10
    U10(q,1)=1;
```

```
else
    U10(q,1)=0;
end

%Batching
Qe2=zeros(b,1);
Bth=batch(b,n,M(:,1));
Lth=batch(b,n,LR(:,1));
for o=1:b
    Qe2(o,1)=GetQt(Bth(:,o),Lth(:,o),p,n/b);
end
avg=sum(single(Qe2))/b;
svar=var(Qe2);
hw4(q,1)=1.833*sqrt(svar)/sqrt(b);
hb4(q,1)=avg+hw4(q,1);
lb4(q,1)=avg-hw4(q,1);
if tq>lb4(q,1) && tq<hb4(q,1)
    U4(q,1)=1;
else
    U4(q,1)=0;
end

%Mean estimation
mavg=sum(M(:,1).*LR(:,1))/n;
mvar=var(M(:,1).*LR(:,1));
hw5(q,1)=1.645*sqrt(mvar)/sqrt(n);
hb5(q,1)=mavg+hw5(q,1);
lb5(q,1)=mavg-hw5(q,1);
if mean<hb5(q,1) && mean>lb5(q,1)
    U5(q,1)=1;
else
    U5(q,1)=0;
end
end

L=H/r;
Q=I/r;
D=[sum(U4)/r sum(U5)/r sum(U6)/r sum(U7)/r sum(U8)/r
sum(U9)/r sum(U10)/r];
E=[sum(hw4)/r sum(hw5)/r sum(hw6)/r sum(hw7)/r sum(hw8)/r
sum(hw9)/r sum(hw10)/r];
result=[L Q D E];
}
```

Compare.m

```
{
function Flag= Compare(C,w,n)
% set up flag for stratification
Flag=0;
for i=1:w
    if C(i,1)<n
       Flag=1;
       break;
    end
end

}
```

G1.m

```
{
function g=G1(theta,d,e,f)
% density function of the mixture of exponential tilting
% distributions
eta=1-theta;
syms t;
a=1-exp(-eta(1,1)*t)*(1+eta(1,1)/(1-eta(1,1))
+eta(1,1)/(1-eta(1,1))^2)
+exp(-t)*(eta(1,1)/(1-eta(1,1)))*(1+t+1/(1-eta(1,1)));
b=1-exp(-eta(2,1)*t)
-eta(2,1)*t*exp(-eta(2,1)*t)
-(eta(2,1)*t)^2*exp(-eta(2,1)*t)/2;
c=1-exp(-eta(3,1)*t)*(1+eta(3,1)/(1-eta(3,1))
+eta(3,1)/(1-eta(3,1))^2)
+exp(-t)*(eta(3,1)/(1-eta(3,1)))*(1+t+1/(1-eta(3,1)));
g=d*a+e*b+f*c;
}
```

GetPsi1.m

```
{
function Psi=GetPsi1(M,LR,w,n,Qt)
%compute psi
Psi2=0;
Xi2=zeros(w,1);
lambda=1/w;
```

```
gamma=1/w;

for j=1:w
S=0;
T=0;
for i=1:n
    if M(i+n*(j-1),1)>Qt
    S=S+LR(i+n*(j-1),1)^2;
    T=T+LR(i+n*(j-1),1);
    end
end
Xi2(j,1)=S/n-(T/n)^2;

Psi2=Psi2+Xi2(j,1)*(lambda^2)/gamma;

end

Psi=sqrt(Psi2);

}
```

GetQt.m

```
{
function Qt=GetQt(M,LR,p,w,n)
%compute quantile estimator
A=[M LR/(n*w)];
B=sortrows(A);
v=n*w;
S=B(v,2);

while v>1 && S<1-p
    v=v-1;
    S=S+B(v,2);
end

Qt=B(v,1);

}
```

GetVar.m

```
{
function svar=GetVar(M,LR,w,n)
%compute the variance
Svar2=0;
Xi2=zeros(w,1);
lambda=1/w;
gamma=1/w;

for j=1:w
S=0;
T=0;
for i=1:n
    S=S+LR(i+n*(j-1),1)^2*M(i+n*(j-1),1)^2;
    T=T+LR(i+n*(j-1),1)*M(i+n*(j-1),1);
end
Xi2(j,1)=S/n-(T/n)^2;
Svar2=Svar2+Xi2(j,1)*(lambda^2)/gamma;
end
svar=sqrt(Svar2);

}
```

STR.m

```
{
function result=STR(p,j,w,n,r,b)
% IS+SS for SAN
tq=6.6645;%theoretical value
mean=3.4583;%theoretical value
result=zeros(1,78);
U1=zeros(r,1);
U2=zeros(r,1);
U3=zeros(r,1);
U4=zeros(r,1);
U5=zeros(r,1);
counter=zeros(r,1);
H=zeros(1,32);
I=zeros(1,32);
Phi1=zeros(8,1);
% precalculated 5 strata
EX1=[
     3.6943
```

```
    5.8424
    8.3344
   12.0055];
for q=1:r
m=[2;3;2];
%theta=GetTheta(m,p,j)
theta=[0.7399;0.6819;0.7399];
v=GetXi(m,theta,j);
MaxXi=trimax(v(1,1),v(2,1),v(3,1));
K=GetK(theta,MaxXi,m,j);
alpha=GetAlpha(K,j);
str=EX1;%w=10
%str=EX2;%w=20
%str=strata(theta,alpha,w);

% sampling
C=zeros(w,1);
M=zeros(n*w,1);
LR=zeros(n*w,1);
flag=1;
while flag==1
    T=zeros(3,1);
    A=zeros(5,1);
    UN=rand();
    if UN<alpha(1,1)
        A(1,1)=exprnd(1/(1-theta(1,1)));
        A(2,1)=exprnd(1/(1-theta(1,1)));
        A(3,1)=exprnd(1);
        A(4,1)=exprnd(1);
        A(5,1)=exprnd(1);
    elseif alpha(1,1)<UN && UN<(alpha(1,1)+alpha(2,1))
        A(1,1)=exprnd(1/(1-theta(2,1)));
        A(3,1)=exprnd(1/(1-theta(2,1)));
        A(5,1)=exprnd(1/(1-theta(2,1)));
        A(2,1)=exprnd(1);
        A(4,1)=exprnd(1);
    elseif (alpha(1,1)+alpha(2,1))<UN && UN<1
        A(4,1)=exprnd(1/(1-theta(3,1)));
        A(5,1)=exprnd(1/(1-theta(3,1)));
        A(1,1)=exprnd(1);
        A(2,1)=exprnd(1);
        A(3,1)=exprnd(1);
    end
    T(2,1)=A(1,1)+A(3,1)+A(5,1);
    T(3,1)=A(4,1)+A(5,1);
```

```
    T(1,1)=A(1,1)+A(2,1);
    counter(q,1)=counter(q,1)+1;
    a=ST(str,w,T(2,1));
    if C(a,1)<n
        C(a,1)=C(a,1)+1;
        M(C(a,1)+(a-1)*n,1)=trimax(T(1,1),T(2,1),T(3,1));
        LR(C(a,1)+(a-1)*n,1)=GetL(theta,alpha,j,m,T);
    end
    flag= Compare(C,w,n);
end

%L=sum(single(LR))/n
%A=[M LR/(n*w)];
%sortrows(A)
% compute forward, backward
% and central phi estimators
o=0;
Y=[0.05 0.1 0.25 0.5 0.75 1 1.5 2];
for c2=1:8
    c1=Y(1,c2);
    o=o+1;
fd=p+c1/sqrt(w*n);
bd=p-c1/sqrt(w*n);
Qt=GetQt(M,LR,p,w,n);
%Qt(q,1)=6.6645;

%estimate phi
q1=GetQt(M,LR,fd,w,n);
q2=GetQt(M,LR,bd,w,n);
Phi1(o,1)=sqrt(w*n)*(q1-q2)/(2*c1);
Phi=26.5396;
Phi2=sqrt(w*n)*(q1-Qt)/c1;
Phi3=sqrt(w*n)*(Qt-q2)/c1;

%estimate psi
Psi=GetPsi1(M,LR,w,n,Qt);
%Psi(q,1)=0.0823;
%Construct Confidence Interval
hw(q,o)=1.645*Phi*Psi/sqrt(n*w);
hb=Qt+hw(q,o);
lb=Qt-hw(q,o);

% confidence intervals
hw1(q,o)=1.645*Phi1(o,1)*Psi/sqrt(n*w);
hb1=Qt+hw1(q,o);
```

```
lb1=Qt-hw1(q,o);
hw2(q,o)=1.645*Phi2*Psi/sqrt(n*w);
hb2=Qt+hw2(q,o);
lb2=Qt-hw2(q,o);
hw3(q,o)=1.645*Phi3*Psi/sqrt(n*w);
hb3=Qt+hw3(q,o);
lb3=Qt-hw3(q,o);

if tq>lb && tq<hb
    U(q,o)=1;
else
    U(q,o)=0;
end

if tq>lb1 && tq<hb1
    U1(q,o)=1;
else
    U1(q,o)=0;
end
if tq>lb2 && tq<hb2
    U2(q,o)=1;
else
    U2(q,o)=0;
end
if tq>lb3 && tq<hb3
    U3(q,o)=1;
else
    U3(q,o)=0;
end

end

H=H+[U(q,:) U1(q,:) U2(q,:) U3(q,:)];
I=I+[hw(q,:) hw1(q,:) hw2(q,:) hw3(q,:)];

%Combined
Phi6=Phi1(1,1)*4/3-Phi1(2,1)/3;
Phi7=Phi1(3,1)*4/3-Phi1(4,1)/3;
Phi8=Phi1(5,1)*4/3-Phi1(7,1)/3;
Phi9=Phi1(6,1)*4/3-Phi1(8,1)/3;
Phi10=(Phi6+Phi7+Phi8+Phi9)/4;

hw6(q,1)=1.645*Phi6*Psi/sqrt(n*w);
hb6=Qt+hw6(q,1);
lb6=Qt-hw6(q,1);
```

```
hw7(q,1)=1.645*Phi7*Psi/sqrt(n*w);
hb7=Qt+hw7(q,1);
lb7=Qt-hw7(q,1);
hw8(q,1)=1.645*Phi8*Psi/sqrt(n*w);
hb8=Qt+hw8(q,1);
lb8=Qt-hw8(q,1);
hw9(q,1)=1.645*Phi9*Psi/sqrt(n*w);
hb9=Qt+hw9(q,1);
lb9=Qt-hw9(q,1);
hw10(q,1)=1.645*Phi10*Psi/sqrt(n*w);
hb10=Qt+hw10(q,1);
lb10=Qt-hw10(q,1);

if tq>lb6 && tq<hb6
    U6(q,1)=1;
else
    U6(q,1)=0;
end
if tq>lb7 && tq<hb7
    U7(q,1)=1;
else
    U7(q,1)=0;
end

if tq>lb8 && tq<hb8
    U8(q,1)=1;
else
    U8(q,1)=0;
end
if tq>lb9 && tq<hb9
    U9(q,1)=1;
else
    U9(q,1)=0;
end
if tq>lb10 && tq<hb10
    U10(q,1)=1;
else
    U10(q,1)=0;
end

%Batching
Qe2=zeros(b,1);
Bth=batch1(M,n,w,b);
Lth=batch1(LR,n,w,b);
for t=1:b
```

```
    Qe2(t,1)=GetQt(Bth(:,t),Lth(:,t),p,w,n/b);
end
avg=sum(single(Qe2))/b;
svar=var(Qe2);
hw4(q,1)=1.833*sqrt(svar)/sqrt(b);
hb4(q,1)=avg+hw4(q,1);
lb4(q,1)=avg-hw4(q,1);
if tq>lb4(q,1) && tq<hb4(q,1)
    U4(q,1)=1;
else
    U4(q,1)=0;
end

%Mean estimation
mavg=sum(M(:,1).*LR(:,1))/(n*w);
mvar=GetVar(M,LR,w,n);
hw5(q,1)=1.645*mvar/sqrt(n*w);
hb5(q,1)=mavg+hw5(q,1);
lb5(q,1)=mavg-hw5(q,1);
if mean<hb5(q,1) && mean>lb5(q,1)
    U5(q,1)=1;
else
    U5(q,1)=0;
end
end

L=H/r;
Q=I/r;
D=[sum(U4)/r sum(U5)/r sum(U6)/r sum(U7)/r
sum(U8)/r sum(U9)/r sum(U10)/r];
E=[sum(hw4)/r sum(hw5)/r sum(hw6)/r sum(hw7)/r
sum(hw8)/r sum(hw9)/r sum(hw10)/r];
result=[L Q D E];
}
```

strata.m

```
{

function s=strata(theta,alpha,w)
% stratification
% create equiprobable strata
g=G1(theta,alpha(1,1),alpha(2,1),alpha(3,1));
```

```
if w==1
    s=0;
else
s=zeros(w-1,1);
for j=1:w-1
    s(j,1)=solve(g-j/w);
end
end
}
```

GetStr.m

```
{
function str=GetStr(w,mu,sigma)
% Compute the strata
if w==1
    str=0;
else
str=zeros(w-1,1);

for i=1:(w-1)
    str(i,1)=norminv(i/w,mu,sigma);
end
end

}
```

STR2.m

```
{
function result=STR2(p,mu1,mu2,sigma1,sigma2,sigma3,w,n,r,b)
% IS+SS for normal/bivariate normal distribution
tq=norminv(0.8,0,1);
%tq=norminv(0.6,mu1,sigma1);%0.6 quantile
%tq=norminv(0.9,mu1,sigma1); %0.9 quantile
mean=0;%theoretical value
U1=zeros(r,1);
U2=zeros(r,1);
U3=zeros(r,1);
U4=zeros(r,1);
U5=zeros(r,1);
H=zeros(1,32);
```

```
I=zeros(1,32);
result=zeros(1,78);
counter=zeros(r,1);
theta=1.7941;%0.8 quantile
%theta=1.3537;%0.6 quantile
%theta=2.1460;% 0.9 quantile
str=GetStr(w,mu1+theta*sigma1^2,sigma1);
MU=[mu1+theta*sigma1^2 mu2+theta*sigma2^2];
SIGMA=[sigma1 sigma3;sigma3 sigma2];

for q=1:r

C=zeros(w,1);
M=zeros(n*w,1);
LR=zeros(n*w,1);
flag=1;
while flag==1
    X=mvnrnd(MU,SIGMA);
    counter(q,1)=counter(q,1)+1;
    a=ST(str,w,X(1,1));
    if C(a,1)<n
        C(a,1)=C(a,1)+1;
        M(C(a,1)+(a-1)*n,1)=X(1,2);
        LR(C(a,1)+(a-1)*n,1)=GetL2(theta,X(1,2),mu2,sigma2);
    end
    flag= Compare(C,w,n);
end

%L=sum(single(LR))/n
%A=[M LR/(n*w)];
%sortrows(A)
Qt=GetQt(M,LR,p,w,n);
Y=[0.05 0.1 0.25 0.5 1 2 4 8];
for o=1:8
    c1=Y(1,o);
fd=p+c1/sqrt(w*n);
bd=p-c1/sqrt(w*n);

%estimate phi
q1=GetQt(M,LR,fd,w,n);
q2=GetQt(M,LR,bd,w,n);
Phi1(o,1)=sqrt(w*n)*(q1-q2)/(2*c1);
Phi=3.5719;%1/f(\xi_p)for p=0.8
Phi2=sqrt(w*n)*(q1-Qt)/c1;
Phi3=sqrt(w*n)*(Qt-q2)/c1;
```

```
%estimate psi
Psi=GetPsi1(M,LR,w,n,Qt);
%Construct Confidence Interval

hw(q,o)=1.645*Phi*Psi/sqrt(n*w);
hb=Qt+hw(q,o);
lb=Qt-hw(q,o);
hw1(q,o)=1.645*Phi1(o,1)*Psi/sqrt(n*w);
hb1=Qt+hw1(q,o);
lb1=Qt-hw1(q,o);
hw2(q,o)=1.645*Phi2*Psi/sqrt(n*w);
hb2=Qt+hw2(q,o);
lb2=Qt-hw2(q,o);
hw3(q,o)=1.645*Phi3*Psi/sqrt(n*w);
hb3=Qt+hw3(q,o);
lb3=Qt-hw3(q,o);
if tq>lb && tq<hb
    U(q,o)=1;
else
    U(q,o)=0;
end
if tq>lb1 && tq<hb1
    U1(q,o)=1;
else
    U1(q,o)=0;
end
if tq>lb2 && tq<hb2
    U2(q,o)=1;
else
    U2(q,o)=0;
end
if tq>lb3 && tq<hb3
    U3(q,o)=1;
else
    U3(q,o)=0;
end

end

H=H+[U(q,:) U1(q,:) U2(q,:) U3(q,:)];
I=I+[hw(q,:) hw1(q,:) hw2(q,:) hw3(q,:)];

%Combined
Phi6=Phi1(1,1)*4/3-Phi1(2,1)/3;
```

```
Phi7=Phi1(3,1)*4/3-Phi1(4,1)/3;
Phi8=Phi1(5,1)*4/3-Phi1(7,1)/3;
Phi9=Phi1(6,1)*4/3-Phi1(8,1)/3;
Phi10=(Phi6+Phi7+Phi8+Phi9)/4;

hw6(q,1)=1.645*Phi6*Psi/sqrt(n*w);
hb6=Qt+hw6(q,1);
lb6=Qt-hw6(q,1);
hw7(q,1)=1.645*Phi7*Psi/sqrt(n*w);
hb7=Qt+hw7(q,1);
lb7=Qt-hw7(q,1);
hw8(q,1)=1.645*Phi8*Psi/sqrt(n*w);
hb8=Qt+hw8(q,1);
lb8=Qt-hw8(q,1);
hw9(q,1)=1.645*Phi9*Psi/sqrt(n*w);
hb9=Qt+hw9(q,1);
lb9=Qt-hw9(q,1);
hw10(q,1)=1.645*Phi10*Psi/sqrt(n*w);
hb10=Qt+hw10(q,1);
lb10=Qt-hw10(q,1);
if tq>lb6 && tq<hb6
    U6(q,1)=1;
else
    U6(q,1)=0;
end
if tq>lb7 && tq<hb7
    U7(q,1)=1;
else
    U7(q,1)=0;
end
if tq>lb8 && tq<hb8
    U8(q,1)=1;
else
    U8(q,1)=0;
end
if tq>lb9 && tq<hb9
    U9(q,1)=1;
else
    U9(q,1)=0;
end
if tq>lb10 && tq<hb10
    U10(q,1)=1;
else
    U10(q,1)=0;
end
```

```
%Batching
Qe2=zeros(b,1);
Bth=batch1(M,n,w,b);
Lth=batch1(LR,n,w,b);
for t=1:b
    Qe2(t,1)=GetQt(Bth(:,t),Lth(:,t),p,w,n/b);
end
avg=sum(single(Qe2))/b;
svar=var(Qe2);
hw4(q,1)=1.833*sqrt(svar)/sqrt(b);
hb4(q,1)=avg+hw4(q,1);
lb4(q,1)=avg-hw4(q,1);
if tq>lb4(q,1) && tq<hb4(q,1)
    U4(q,1)=1;
else
    U4(q,1)=0;
end

%Mean estimation
mavg=sum(M(:,1).*(LR(:,1)/(n*w)));
mvar=GetVar(M,LR,w,n);
hw5(q,1)=1.645*mvar/sqrt(n*w);
hb5(q,1)=mavg+hw5(q,1);
lb5(q,1)=mavg-hw5(q,1);
if mean<hb5(q,1) && mean>lb5(q,1)
    U5(q,1)=1;
else
    U5(q,1)=0;
end

end
L=H/r;
Q=I/r;
D=[sum(U4)/r sum(U5)/r sum(U6)/r sum(U7)/r
sum(U8)/r sum(U9)/r sum(U10)/r];
E=[sum(hw4)/r sum(hw5)/r sum(hw6)/r sum(hw7)/r
sum(hw8)/r sum(hw9)/r sum(hw10)/r];
result=[L Q D E];
}
```

CHAPTER 9

CONCLUDING REMARKS

The main contribution is that we have developed a general framework for producing an asymptotically valid confidence interval for a quantile ξ_p estimated using a VRT. Previous research on crude Monte Carlo also produced methods of contructing confidence intervals, but their resulting confidence intervals may be large especially for extreme quantiles and their methods cannot be generalized for VRTs. Our method, on the other hand, addresses this issue by providing a theoretical framework for multiple VRTs. Most previous work on estimating ξ_p using different VRTs did not produce asymptotically valid confidence intervals for ξ_p. Our theoretical framework can be used directly to accomodate different VRTs to construct confidence intervals. Specifically, our framework, which requires the CDF estimator obtained by applying a VRT to satisfy Assumptions A1–A3 in Chapter 4, encompasses IS+SS, AV and CV. Moreover, we have also presented explicit algorithms to construct confidence intervals for quantiles obtained using IS, IS+SS, AV and CV.

Another main contribution is that we have proved the consistency of the estimators for $1/f(\xi_p)$ (i.e., the forward finite-difference, the backward finite-difference and the central finite-difference estimators) for VRTs. According to Glynn (1996), the "major challenge" to constructing confidence intervals for quantiles when using VRTs is estimating $\phi_p = 1/f(\xi_p)$, which Glasserman et al. (2000b) called a "difficult" problem. We derived a consistent estimator of ϕ_p by first establishing that the quantile estimator satisfies a Bahadur-Ghosh representation, and then exploiting this to estimate ϕ_p. The quantity ϕ_p can be expressed as $\frac{d}{dp}F^{-1}(p)$, and we estimated it via a finite difference of the inverse of the estimated CDF at points that are $cn^{-1/2}$ apart, where $c \neq 0$ is a user-specified smoothing parameter and n is the computational budget. In the case of crude Monte Carlo, previous analysis suggested that one should choose c as large as possible to minimize MSE and the coverage error, and our experimental results when applying VRTs seem to confirm these suggestions. But how to choose c is still an ongoing topic for further research.

Other contributions are reviewed as follows. In our theoretical framework, we have established a weaker form of the Bahadur-Ghosh representation for VRTs, which implies CLTs and methods of developing consistent estimators of ϕ_p. In IS+SS, we have established a weaker moment condition than that in Glasserman et al. (2000b).

The contributions of the main results in this research are subtle albeit significant to the financial industry. So far, the most commonly used measure is still the point estimator of the VaR, which can be replaced by another measure –the confidence interval of the VaR– with more accuracy. Our framework provides explicit algorithms for constructing confidence intervals for the VaRs when using VRTs. This can be very useful in estimating extreme quantiles and/or in the high-dimensional setting, which is an interesting finance research topic.

The limitation of this research is that we have only considered the normal distribution and the SAN to study the finite-sample behavior of the confidence intervals constructed using the theoretical framework. We may want to explore a few more stochastic models; e.g., those of interest to the financial industry.

It may be interesting to propose new IS+SS methods for estimating the VaR of a portfolio. Glasserman et al. (2000b) develop IS+SS techniques when the price changes in the porfolio's "risk factors" (e.g., interest rates, currency exchange rates, stock prices, etc.) follow a multivariate normal distribution. However, a normal/multivariate normal distribution does not agree with the market data with higher peaks and heavier tails. Therefore, different stochastic models for the risk factors to follow appeal to the financial industry. Some possible extensions to investigate include mixtures of multivariate normals, a Markov regime-switching process, (Goldfeld and Quandt (1973)) or a jump-diffusion process (Merton (1976)).

REFERENCES

Avramidis, A. N. and Wilson, J. R. (1998), "Correlation-induction techniques for estimating quantiles in simulation," *Operations Research*, 46, 574–591.

Babu, G. J. (1986), "Efficient estimation of the reciprocal of the density quantile function at a point," *Statistics and Probability Letters*, 4, 133–139.

Bahadur, R. R. (1966), "A note on quantiles in large samples," *Annals of Mathematical Statistics*, 37, 577–580.

Billingsley, P. (1995), *Probability and Measure*, New York: John Wiley & Sons, 3rd ed.

— (1999), *Convergence of Probability Measures*, New York: John Wiley & Sons, 2nd ed.

Black, F. and Scholes, M. (1973), "The pricing of options and corporate liabilities," *Journal of Political Economy*, 81, 637–654.

Bloch, D. A. and Gastwirth, J. L. (1968), "On a simple estimate of the reciprocal of the density function," *Annals of Mathematical Statistics*, 39, 1083–1085.

Bofinger, E. (1975), "Estimation of a density function using order statistics," *Australian Journal of Statistics*, 17, 1–7.

Boyle, P., Broadie, M., and P.Glasserman (1997), "Monte Carlo methods for security pricing," *Journal of Economic Dynamics and COntrol*, 21, 1267–1321.

David, H. A. and Nagaraja, H. (2003), *Order Statistics*, Hoboken, NJ: Wiley, 3rd ed.

Duffie, D. and Pan, J. (1997), "An overview of value at risk," *Journal of Derivatives*, 4, 7–49.

Fu, M. C., Hong, L. J., and Hu, J.-Q. (2009), "Conditional Monte Carlo Estimation of Quantile Sensitivities," *Management Science*, 55, 2019–2027.

Ghosh, J. K. (1971), "A new proof of the Bahadur representation of quantiles and an application," *Annals of Mathematical Statistics*, 42, 1957–1961.

Glasserman, P. (2004), *Monte Carlo Methods in Financial Engineering*, New York: Springer.

Glasserman, P., Heidelberger, P., and Shahabuddin, P. (2000a), "Importance Sampling and Stratification for Value-at-Risk," in *Computational Finance 1999, Proceedings of the Sixth International Conference on Computational Finance*, eds. Abu-Mostafa, Y. S., Baron, B. L., Lo, A. W., and Weigend, A. S., pp. 7–24.

— (2000b), "Variance reduction techniques for estimating value-at-risk," *Management Science*, 46, 1349–1364.

Glynn, P. W. (1996), "Importance sampling for Monte Carlo estimation of quantiles," in *Mathematical Methods in Stochastic Simulation and Experimental Design: Proceedings of the 2nd St. Petersburg Workshop on Simulation*, Publishing House of St. Petersburg University, St. Petersburg, Russia, pp. 180–185.

Glynn, P. W. and Iglehart, D. L. (1989), "Importance sampling for stochastic systems," *Management Science*, 35, 1367–1393.

Glynn, P. W. and Whitt, W. (1992), "The asymptotic efficiency of simulation estimators," *Operations Research*, 40, 505–520.

Goldenberg, D. K., Qiu, L., Xie, H., Yang, Y. R., and Zhang, Y. (2004), "Optimizing cost and performance for multihoming," in *SIGCOMM '04 Proceedings*, pp. 79–92.

Goldfeld, S. M. and Quandt, R. E. (1973), "A Markov Model for Switching Regressions," *Journal of Econometrics*, 1, 3–16.

Hall, P. and Sheather, S. J. (1988), "On the Distribution of a Studentized Quantile," *Journal of the Royal Statistical Society B*, 50, 381–391.

Hardy, G. H. (1952), *A Course in Pure Mathematics*, New York: Cambridge University Press, 10th ed.

Heidelberger, P. (1995), "Fast simulation of rare events in queueing and reliability models," *ACM Transactions on Modeling and Computer Simulation*, 5, 43–85.

Hesterberg, T. C. and Nelson, B. L. (1998), "Control variates for probability and quantile estimation," *Management Science*, 44, 1295–1312.

Hogg, R. V., McKean, J. W., and Craig, A. (2004), *Introduction to Mathematical Statistics*, Englewood Cliffs, New Jersey: Prentice Hall, 6th ed.

Hong, L. J. (2009), "Estimating quantile sensitivities," *Operations Research*, 57, 118–130.

Hsu, J. C. and Nelson, B. L. (1990), "Control variates for quantile estimation," *Management Science*, 36, 835–851.

Hull, J. C. (2003), *Options, Futures and Other Derivatives*, New Jersey: Pearson-Hall, Inc, 5th ed.

Jin, X., Fu, M. C., and Xiong, X. (2003), "Probabilistic Error Bounds for Simulation Quantile Estimation," *Management Science*, 49, 230–246.

Juneja, S., Karandikar, R., and Shahabuddin, P. (2007), "Asymptotics and Fast Simulation for Tail Probabilities of Maximum of Sums of Few Random Variables," *ACM Transactions on Modeling and Computer Simulation*, 17, article 2, 35 pages.

Kemna, A. G. Z. and Vorst, A. C. F. (1990), "A pricing method for options based on average asset values," *Journal of Banking and Finance*, 14, 113–129.

Kiefer, J. (1967), "On Bahadur's representation of sample quantiles," *Annals of Mathemetical Statistics*, 38, 1350–1353.

Kuncicky, D. C. (2004), *Matlab Programming*, Upper Saddle River: Pearson Education.

Law, A. M. (2006), *Simulation Modeling and Analysis*, New York: McGraw-Hill, 4th ed.

Lehmann, E. L. (1999), *Elements of Large-Sample Theory*, New York: Springer.

Liu, G. and Hong, L. J. (2009), "Kernel estimation of quantile sensitivities," *Naval Research Logistics*, 56, 511–525.

Merton, R. C. (1976), "Option pricing when underlying stock returns are discontinuous," *Journal of Financial Economics*, 3, 125–144.

Parzen, E. (1979), "Density Quantile Estimation Approach to Statistical Data Modelling," in *Smoothing Techniques for Curve Estimation*, Berlin: Springer.

Ross, S. (1997), *Simulation*, San Diego, CA: Academic Press, 2nd ed.

— (2007), *Introduction to Probability Models*, San Diego, CA: Academic Press, nine ed.

Serfling, R. J. (1980), *Approximation Theorems of Mathematical Statistics*, New York: John Wiley & Sons.

Sommerville, I. (2004), *Software Engineering, Seventh Edition*, Reading, Massachusetts: Addison-Wesley.

Sun, L. and Hong, L. J. (2010), "Asymptotic Representations for Importance-Sampling Estimators of Value-at-Risk and Conditional Value-at-Risk," *Operations Research Letters*, to appear.

Tukey, J. W. (1965), "Which Part of the Sample Contains the Information?" *Proc Natl Acad Sci USA*, 53, 127–134.

Whitt, W. (2002), *Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and Their Application to Queues*, New York: Springer-Verlag.