An Importance Sampling Scheme For Multi-Credit-State Portfolios

by

Adam Sturge

A research paper submitted in conformity with the requirements
for the degree of Masters of Science
Graduate Department of Computer Science
University of Toronto

Abstract

An Importance Sampling Scheme For Multi-Credit-State Portfolios

Adam Sturge
Masters of Science
Graduate Department of Computer Science
University of Toronto
2018

Portfolio credit risk based on the Gaussian copula factor model is generally evaluated through Monte Carlo Integration. Glasserman and Li purposed a 2 level importance sampling scheme to improve the accuracy of this numerical approximation. However their result is limited to the binary credit state model. Zhe Wang developed a novel importance sampling technique for the outer level based on the work of Meng Han. While this method shows significant variance reduction, it introduces a bias away from the true answer. In this research paper we remove the bias from the outer level of Wang's method and extend the inner level of Glasserman and Li's method to the multi-credit-state model, combining them into a single powerful technique.

Acknowledgements

First I would like to thank my supervisor Dr. Ken Jackson for his guidance, which was invaluable whenever I became mired in some misunderstanding or just needed someone to bounce ideas off of. Similarly Meng Han and Zhe Wang were gracious enough to donate their time on more than one occasion to review my work and offer suggestions, for which I am thankful. Lastly Dr. Alex Kreinin has my gratitude for contributing their time as a second reviewer of this research paper.

Contents

1 Introduction **1**

1.1 Credit Risk . . . 1

1.1.1 Credit States . . . 1

1.1.2 Gaussian Copula Factor Model . . . 1

1.2 An Overview of the Thesis . . . 3

2 Naive Two Level Monte Carlo **4**

2.1 Monte Carlo Integration . . . 4

2.2 Bernoulli Indicators . . . 4

3 Glasserman and Li Importance Sampling **7**

3.1 Overview of Importance Sampling . . . 7

3.2 Exponential Twisting . . . 7

3.3 Glasserman and Li Importance Sampling . . . 8

3.3.1 Constrained Optimization for Bi-level Optimization Problems . . . 9

3.3.2 Gradients in Bi-level Optimization Problems . . . 9

4 Outer Level Importance Sampling **11**

4.1 Normal Approximation . . . 11

4.2 $\mathcal{Z}$ Importance Sampling . . . 12

4.2.1 Zero Variance Importance Sampling . . . 12

4.2.2 Approximating the Zero Variance Importance Sampler . . . 13

5 Inner Level Importance Sampling **15**

5.1 Discrete Importance Sampling . . . 15

5.2 Inner Level Exponential Twisting . . . 16

5.2.1 Final Algorithm . . . 17

6 Numerical Results **19**

6.1 Binary Credit State Experiments . . . 19

6.1.1 Missing Glasserman and Li Data . . . 19

6.1.2 Model Parameter Settings . . . 20

6.1.3 Experiment Setup . . . 20

6.1.4 S,ℓ Dependency . . . 20

6.1.5 π^* Training . . . 39

6.1.6 Central Limit Theorem Bias 44
6.2 Full Credit State Experiments 46

7 Conclusion **48**

8 Future Work **49**
8.1 Fix Glasserman and Li 49
8.2 Improve $\mathcal{E}$ Importance Sampling 49

Appendices **51**

A Approximate Importance Sampler for Unbiased Estimator **52**

B Likelihood Ratio for Exponential Twisting **55**

C Convex Bound on Second Moment for Inner Level Importance Sampler **56**

Bibliography **57**

List of Figures

6.1 $S = 5\ \ell = 0.3$ . . . 21
6.2 $S = 5\ \ell = 0.3$ . . . 22
6.3 $S = 5\ \ell = 0.2$ . . . 23
6.4 $S = 5\ \ell = 0.2$ . . . 24
6.5 $S = 5\ \ell = 0.1$ . . . 25
6.6 $S = 5\ \ell = 0.1$ . . . 26
6.7 $S = 10\ \ell = 0.3$. We did not include 2LvlMC or a line indicating the true answer for the reasons explained in the Analysis subsection on page 39. . . . 27
6.8 $S = 10\ \ell = 0.3$ . . . 28
6.9 $S = 10\ \ell = 0.2$ . . . 29
6.10 $S = 10\ \ell = 0.2$ . . . 30
6.11 $S = 10\ \ell = 0.1$ . . . 31
6.12 $S = 10\ \ell = 0.1$ . . . 32
6.13 $S = 20\ \ell = 0.3$. We did not include 2LvlMC or a line indicating the true answer for the reasons explained in the Analysis subsection on page 39. . . . 33
6.14 $S = 20\ \ell = 0.3$ . . . 34
6.15 $S = 20\ \ell = 0.2$ . . . 35
6.16 $S = 20\ \ell = 0.2$ . . . 36
6.17 $S = 20\ \ell = 0.1$ . . . 37
6.18 $S = 20\ \ell = 0.1$ . . . 38
6.19 Variance as a function of S . . . 40
6.20 $S = 20\ \ell = 0.1$ . . . 41
6.21 $S = 20\ \ell = 0.2$ . . . 42
6.22 $S = 5\ \ell = 0.2$ . . . 43
6.23 $S = 10\ \ell = 0.2$ . . . 45
6.24 $S = 10\ \ell = 0.2$ . . . 46

List of Tables

1.1 Credit Migration Matrix 1

6.1 Error and variance for $S = 5$, $\ell = 0.3$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 7.02e−4 21
6.2 Variance reduction for $S = 5$, $\ell = 0.3$ 22
6.3 Error and variance for $S = 5$, $\ell = 0.2$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 8.74e−3 23
6.4 Variance reduction for $S = 5$, $\ell = 0.2$ 24
6.5 Error and variance for $S = 5$, $\ell = 0.1$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 1.15e−1 25
6.6 Variance reduction for $S = 5$, $\ell = 0.1$ 26
6.7 Error and variance for $S = 10$, $\ell = 0.3$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is not known in this case as our overnight run of 2LvlMC did not converge. See the discussion in the Analysis subsection on page 39. 27
6.8 Variance reduction for $S = 10$, $\ell = 0.3$ 28
6.9 Error and variance for $S = 10$, $\ell = 0.2$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 1.20e−3 29
6.10 Variance reduction for $S = 10$, $\ell = 0.2$ 30
6.11 Error and variance for $S = 10$, $\ell = 0.1$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 8.17e−2 31
6.12 Variance reduction for $S = 10$, $\ell = 0.1$ 32
6.13 Error and variance for $S = 20$, $\ell = 0.3$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is not known in this case as our overnight run of 2LvlMC did not converge. See the discussion in the Analysis subsection on page 39. 33
6.14 Variance reduction for $S = 20$, $\ell = 0.3$ 34
6.15 Error and variance for $S = 20$, $\ell = 0.2$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 5.41e−5 35
6.16 Variance reduction for $S = 20$, $\ell = 0.2$ 36
6.17 Error and variance for $S = 20$, $\ell = 0.1$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 5.05e−2 37
6.18 Variance reduction for $S = 20$, $\ell = 0.1$ 38
6.19 Variance reduction for $S = 20$, $\ell = 0.1$ as a function of number of samples used to train π^* 41
6.20 Variance reduction for $S = 20$, $\ell = 0.2$ as a function of number of samples used to train π^* 42
6.21 Variance reduction for $S = 5$, $\ell = 0.2$ as a function of number of samples used to train π^* 43

6.22 Error and variance for $S = 10$, $\ell = 0.2$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 1.08e−3. The CLT approximation is approximately 1.04e−3 45
6.23 Error and variance for $S = 10$, $\ell = 0.2$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 1.08e−3. The CLT approximation is approximately 1.04e−3 46

List of Algorithms

2.1 2 level naive Monte Carlo integration 4
2.2 2 level naive Monte Carlo integration with Bernoulli indicator sampling 5
4.1 1 level Monte Carlo integration 12
4.2 1 level Monte Carlo integration with $\mathcal{Z}$ importance sampling 13
4.3 2 level Monte Carlo integration with $\mathcal{Z}$ importance sampling 14
5.1 2 level Monte Carlo integration with $\mathcal{E}$ importance sampling 17
5.2 2 level Monte Carlo integration with 2 level importance sampling 17

Chapter 1

Introduction

1.1 Credit Risk

Credit risk refers to the risk that a given creditor may fail to repay their loan, either its principal or the accrued interest. When this happens the entity that owns the loan suffers a loss. It is therefore a natural question to ask what is the probability that a given portfolio of creditors will result in a loss greater than a specified amount.

1.1.1 Credit States

A credit rating is a measure of how likely someone is to repay their loans. Therefore in practice portfolio loss due to credit risk is measured by drops in the credit rating of one or more creditors.

A credit migration matrix defines the baseline probability that any creditor will transition from one credit state to another, without taking into account any of the specifics about the individual. Table 1.1 shows an example of such a matrix. Each cell contains the probability that a creditor in the credit state associated with the row will transition to the credit state associated with the column. In the provided example once someone defaults (credit state D) they will remain in credit state D with 100% probability.

1.1.2 Gaussian Copula Factor Model

A few models have been developed to measure credit risk; the most widely used is the Gaussian copula factor model. In this framework creditors' risks are modeled by a set of underlying risk factors known as systemic and idiosyncratic risk factors. Throughout this paper we follow the notation introduced by Han [7] with a few additions.

1. diagonal usually largest: most of time rating class does not change
2. usually transition to neighboring states is more likely

	D(1)	C(2)	B(3)	A(4)
D(1)	1.0000	0.0000	0.0000	0.0000
C(2)	0.2550	0.6801	0.0649	0.0000
B(3)	0.0270	0.0125	0.9397	0.0208
A(4)	0.0002	0.0000	0.0202	0.9796

Table 1.1: Credit Migration Matrix

credit states for 1 creditor

in reality, ~20 credit state
+ most secure: US gov bond
+ less secure: ….
+ lowest category: default

bond becomes more valuable when it goes up the credit state, in this case, negative loss…

interested in: how change in credit state affect loss

probability of jumping to state that is far from current is small motivation for using gaussian

N	=	Number of creditors
C	=	Number of credit states 1,...,C
S	=	Dimension of systematic risk factor
$\mathcal{L}_N$	=	The percentage loss of the portfolio
$\mathbb{1}_n^c$	=	Indicator function. 1 if creditor n is in credit state c, 0 otherwise
EAD_n	=	Exposure at default. The value lost if creditor n defaults
LGC_n^c	=	Percentage loss/gain if creditor n moves to credit state c
$c(n)$	=	Current credit state of creditor n
$\mathbb{E}_p[X]$	=	Expected value of a random variable X distributed according to a pdf p
$d(x \sim p)$	=	Used in integrals when the integration variable is distributed according p. Analytically equivalent to writing $p(x)dx$

how much money at default
how much money will be lost at default
0~1
c(1) credit 1 at current time

In Merton's default model [8] a credit transition is modeled as

= probability of credit y inside the interval, representing transition to credit state c

motivation for threshold H
want to have probability of y_n within 2 consecutive threshold
the probability of transition from c(n) to c

$$\mathbb{1}_n^c = \mathbb{1}_{\{H_{c(n)}^{c-1} \leq y_n \leq H_{c(n)}^c\}} \tag{1.1}$$

state is discrete, but the gaussian samples are continuous, need those thresholds

$$H_{c(n)}^c = \Phi^{-1}\left(\sum_{\gamma \leq c} P_{c(n)}^{\gamma}\right) \tag{1.2}$$

cdf, area under the curve

In the above equation y_n is the creditworthness index for the n^{th} creditor. $H_{c(n)}^c$ can be thought of as the threshold for the n^{th} creditor to migrate from credit state $c(n)$ to credit state c. It is defined as the inverse cumulative normal distribution function applied to the sum of the probability values $P_{c(n)}^{\gamma}$. These probabilities are read off from the credit migration matrix as the probability of creditor n migrating from credit state $c(n)$ to credit state γ

one factor modelcov

As alluded to above we decompose y_n into a systematic risk factor $\mathcal{Z} \sim \mathcal{N}(0, I_S)$ and an idiosyncratic risk factor $\varepsilon \sim \mathcal{N}(0, I_N)$. Here I_m refers to the $m \times m$ identity matrix.

individual

Z: modeling different sector of economy
correlation between bonds modeled with betas
beta increase -> more correlated

cov(y_i, y_j) != 0 by presence of Z

$$y_n = \beta_n^T \mathcal{Z} + \sqrt{1 - \beta_n^T \beta_n}\epsilon_n$$

cov(Z, epsilon) = 0

E(y_n) = 0
var(y_n) = 1
y_n ~ N(0,1)
therefore use cdf of standard normal phi

$\beta_n \in \mathbb{R}^S$ represents the n^{th} creditor's sensitivity to the systematic credit risk $\mathcal{Z}$

Under this decomposition (1.1) can be rewritten

cov(y1,y_2) = beta_1 * beta_2

indicator: obligor n is in credit state c

$$\mathbb{1}_n^c = \mathbb{1}_{\left\{\frac{H_{c(n)}^{c-1} - \beta_n^T \mathcal{Z}}{\sqrt{1-\beta_n^T\beta_n}} \leq \epsilon_n \leq \frac{H_{c(n)}^{c} - \beta_n^T \mathcal{Z}}{\sqrt{1-\beta_n^T\beta_n}}\right\}} \tag{1.3}$$

2 level
1. sample Z in outer level, so H is constant in inner level
2. sample epsilon in inner level

With this in hand we can write down the model equation for the percentage loss $\mathcal{L}_N$:

randomness in the model comes from 2
gaussian/copula distribution in expression for y

$$\begin{aligned}
\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) &= \frac{\sum_{n=1}^{N}\left(EAD_n \sum_{c=1}^{C} LGC_n^c \mathbb{1}_n^c\right)}{\sum_{n=1}^{N} EAD_n} \\
&= \sum_{n=1}^{N}\left(\omega_n \sum_{c=1}^{C} LGC_n^c \mathbb{1}_n^c\right) \\
&= \sum_{n=1}^{N}\sum_{c=1}^{C} \omega_n^c \mathbb{1}_{\left\{\frac{H_{c(n)}^{c-1} - \beta_n^T \mathcal{Z}}{\sqrt{1-\beta_n^T\beta_n}} \leq \epsilon_n \leq \frac{H_{c(n)}^{c} - \beta_n^T \mathcal{Z}}{\sqrt{1-\beta_n^T\beta_n}}\right\}}
\end{aligned} \tag{1.4}$$

Where

$$\omega_n = \frac{EAD_n}{\sum_{n=1}^{N} EAD_n} \tag{1.5}$$

$$\omega_n^c = \omega_n LGC_n^c \tag{1.6}$$

The probability that the percentage loss is greater than some tail value ℓ is

sample Z, then sample epsilon, then average the result

$$\begin{aligned} P(\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell) &= \mathbb{E}_{\phi_S}[P(\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell | \mathcal{Z} = z)] \\ &= \int_{\mathbb{R}^S} P(\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell | \mathcal{Z} = z) d(z \sim \phi_S) \\ &= \int_{\mathbb{R}^S} \mathbb{E}_{\phi_N}[\mathbb{1}_{\{\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell | \mathcal{Z}=z, \mathcal{E}=\varepsilon\}}] d(z \sim \phi_S) \\ &= \int_{\mathbb{R}^S} \int_{\mathbb{R}^N} \mathbb{1}_{\{\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell | \mathcal{Z}=z, \mathcal{E}=\varepsilon\}} d(\varepsilon \sim \phi_N) d(z \sim \phi_S) \end{aligned} \tag{1.7}$$

z ~ N(0,I_S)

epsilon ~ N(0,I_N)

ϕ_S and ϕ_N being the standard multivariate normal probability density function for $\mathbb{R}^S$ and $\mathbb{R}^N$ respectively.

Written in this manner it is clear that one way to numerically approximate $P(\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell)$ is to sample $\mathcal{Z}$ and $\mathcal{E}$ from their corresponding normal distributions and perform Monte Carlo Integration.

1.2 An Overview of the Thesis

Chapter 2 is dedicated to explaining a simple Monte Carlo Integration scheme that serves as our baseline method.

Chapter 3 briefly reviews the techniques of importance sampling and exponential twisting. We then elaborate on how they are used by Glasserman and Li [5] to evalutate (1.7). A limition of their method, as it is described in [5], is that it is can only be applied to the binary credit state model. This limitation is removed by us in chapter 5. A discussion follows of how to get around some of the convergence issues faced by Wang [10] in their implementation of Glasserman and Li.

Chapter 4 discusses a normal approximation proposed by Han [7] in which the double integral in (1.7) is replaced by a single integral. This approximation was used by Wang [10] to build an importance sampling technique but it unfortunately suffered from a bias due to the underlying normal approximation. After introducing these results we proceed to offer a simple modification to Wang's technique that removes the bias.

Chapter 5 takes the exponential twisting technique used by Glassermand and Li [5] for the binary credit state model and extends it to the multi-credit-state model.

Chapter 6 contains numerical simulations showing the effectiveness of our new algorithms.

Chapter 7 concludes and summarizes our work.

Chapter 8 discusses possibilities of future work that might further improve our new methods.

Chapter 2

Naive Two Level Monte Carlo

2.1 Monte Carlo Integration

In chapter 1 we said that (1.7) begged for a Monte Carlo (MC) approach [1]. What we meant was the following algorithm.

Algorithm 2.1: 2 level naive Monte Carlo integration

Inputs: NZ, NE, $\{EAD_n\}_{n=1}^{N}$, $\{LGC_n^c\}_{n=1,c=1}^{N,C}$, β, H, ℓ
  Sample $\{z_i\}_{i=1}^{NZ}$ from $\mathcal{N}(0, I_S)$
  FOR $i = 1 : NZ$
    Sample $\{\varepsilon_j\}_{j=1}^{NE}$ from $\mathcal{N}(0, I_N)$
    FOR $j = 1 : NE$
      Compute $\{\mathbb{1}_n^c\}_{n=1,c=1}^{N,C}$ from z_i and ε_j
      Compute $\mathcal{L}_{ij} = \sum_{n=1}^{N}\sum_{c=1}^{C} \omega_n^c \mathbb{1}_n^c$
      Store $a_{ij} = \mathbb{1}_{\{\mathcal{L}_{ij} > \ell\}}$
  Return mean(a_{ij})

EAD, LGC, H random, but use the saved version for comparison sake

We wish to call attention to the icon in the algorithm title. This is a link to a github repo containing a Matlab implementation of the algorithm. All algorithms presented in this paper have such a link.

This algorithm produces an estimate for $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$. We refer to this method as "naive" because it does not contain any techniques to improve the estimate. In later chapters we develop better algorithms. Although this is a perfectly viable way to formulate the MC integration we have found there is another way that lends itself better to theoretical discussion and subsequent improvement. In order to get to that point first we have to do some analysis of the Gaussian copula factor model.

2.2 Bernoulli Indicators

change 1 <= to a < makes more sense

Conditional on $\mathcal{Z} = z$

$$\mathbb{1}_n^c = \mathbb{1}_{\left\{ \frac{H_{c(n)}^{c-1} - \beta_n^T z}{\sqrt{1-\beta_n^T \beta_n}} \leq \epsilon_n \leq \frac{H_{c(n)}^{c} - \beta_n^T z}{\sqrt{1-\beta_n^T \beta_n}} \right\}}$$

previously: sample z first then sample e
now: sample z first then sample from a bernoulli according to p(z)

why any given credit state, bernoulli RV = 1 for exactly 1 creditor?
try to answer: since bernoulli is discrete, would make sense to transition to 1 state only

In this form $\mathbb{1}_n^c$ can be thought of as a Bernoulli random variable

$$\mathbb{1}_n^c = \begin{cases} 1 & p_n^c(z) \\ 0 & 1 - p_n^c(z) \end{cases} \tag{2.1}$$

where

condition-ed on z

$$p_n^c(z) = \Phi\left(\frac{H_n^c - \beta_n^T z}{\sqrt{1-\beta_n^T\beta_n}}\right) - \Phi\left(\frac{H_n^{c-1} - \beta_n^T z}{\sqrt{1-\beta_n^T\beta_n}}\right) \tag{2.2}$$

because $\epsilon_n \sim \mathbb{N}(0,1)$

difference in area under the curve

Hence another way to perform MC integration under this model is sample all $\{\mathbb{1}_n^c(z)\}_{n=1,c=1}^{N,C}$ according to $\{p_n^c(z)\}_{n=1,c=1}^{N,C}$ with the constraint that for any given m $\{\mathbb{1}_m^c(z)\}_{c=1}^{C}$ contains exactly one 1 (meaning the rest of the indicators would be 0). Sampling $\{\mathbb{1}_n^c(z)\}_{n=1,c=1}^{N,C}$ would be tricky in general but for the fact that the correlation matrix for $\mathcal{E}$ is the identity matrix. This implies

intuitively, idiosyncratic risk is independent cross creditors, conditioned on Z=z, when systematic risk is given

$$p(\mathbb{1}_1^1, \cdots, \mathbb{1}_1^C, \mathbb{1}_2^1, \cdots, \mathbb{1}_2^C, \cdots, \mathbb{1}_N^1, \cdots, \mathbb{1}_N^C) = \prod_{n=1}^{N} p_n(\mathbb{1}_n^1, \cdots, \mathbb{1}_n^C) \tag{2.3}$$

Meaning we can sample the indicator functions for each creditor separately. However further decomposition is impossible since there is a clear negative correlation between $\mathbb{1}_m^c$ and $\mathbb{1}_m^d$, as only one can be active at a time.

Put another way we need to sample discrete random variables $\{W_n(z)\}_{n=1}^{N}$

W is a discrete probability distribution

w represent precentage loss
w = f(LGC, EAD) -> can be computed

$$W_n(z) = \begin{cases} \omega_n^1, & p_n^1(z) \\ \omega_n^2, & p_n^2(z) \\ \vdots & \\ \omega_n^C, & p_n^C(z) \end{cases} \tag{2.4}$$

and the percentage loss can be written

idea is we can sample z~N(0,I), separate the x-axis into domains

$$\mathcal{L}_N(z) = \sum_{n=1}^{N} W_n(z) \tag{2.5}$$

This discrete distribution can be sampled by first computing the cdf $P_n^c(z)$ from the discrete pdf:

$$P_n^c(z) = \sum_{k=1}^{c} p_n^k(z)$$

with cumsum

From there sampling a standard uniform random variable u and selecting the first ω_n^c for which $P_n^c(z) > u$ gives one the sampled value for $W_n(z)$.

This way of looking at the problem is fruitful later on. Utilizing these insights we can recast Algorithm 2.1 in the following manner.

TwoLvlMC Algorithm 2.2: 2 level naive Monte Carlo integration with Bernoulli indicator sampling

1 Inputs: NZ, NE, $\{EAD_n\}_{n=1}^{N}$, $\{LGC_n^c\}_{n=1,c=1}^{N,C}$, β, H, ℓ
2 Sample $\{z_i\}_{i=1}^{NZ}$ from $\mathcal{N}(0, I_S)$

3 FOR $i = 1 : NZ$

4 Compute $\{p_n^c(z_i)\}_{n=1,c=1}^{N,C}$

5 FOR $j = 1 : NE$ sampling from discrete distributions

6 Sample $\{\mathbb{1}_n^c\}_{n=1,c=1}^{N,C}$ according to $\{p_n^c(z_i)\}_{n=1,c=1}^{N,C}$

7 Compute $\mathcal{L}_{ij} = \sum_{n=1}^{N} \sum_{c=1}^{C} \omega_n^c \mathbb{1}_n^c$

8 Store $a_{ij} = \mathbb{1}_{\{\mathcal{L}_{ij} > \ell\}}$

9 Return mean(a_{ij})

This algorithm serves as the baseline method against which we compare the more sophisticated methods that are developed in later chapters.

Chapter 3

Glasserman and Li Importance Sampling

3.1 Overview of Importance Sampling

The goal of importance sampling [3] is, given a function of a random variable $f(z)$, where $\mathcal{Z}$ has a pdf $p(z)$, to find a pdf $\pi(z)$ such that $\mathbb{V}_\pi[\frac{f(\mathcal{Z})p(\mathcal{Z})}{\pi(\mathcal{Z})}] < \mathbb{V}_p[f(\mathcal{Z})]$. This switch maintains the expectation as can been seen below.

likelihood = p(z) / pi(z)

$$\begin{aligned}
\mathbb{E}_p[f(\mathcal{Z})] &= \int_{\mathbb{R}} f(z) d(z \sim p) \\
&= \int_{\mathbb{R}} f(z) p(z) dz \\
&= \int_{\mathbb{R}} f(z) \frac{p(z)}{\pi(z)} \pi(z) dz \\
&= \int_{\mathbb{R}} f(z) \frac{p(z)}{\pi(z)} d(z \sim \pi) \\
&= \mathbb{E}_\pi \left[\frac{f(\mathcal{Z})p(\mathcal{Z})}{\pi(\mathcal{Z})} \right]
\end{aligned}$$

So if our goal is to estimate the expected value of $f(z)$ we can instead estimate the expected value of this new random variable $\frac{f(z)p(z)}{\pi(z)}$ and take advantage of its lower variance to get smaller confidence intervals. Or alternatively one can use fewer samples to achieve the same confidence intervals.

requires less sampling, faster for an equivalent CI

3.2 Exponential Twisting

definition: a way of generating a class of distributions from p

When performing an exponential twist on a (possibly discrete) probability distribution $p(x)$ we define a new probability distribution

$$q(x, \theta) = \frac{e^{\theta x} p(x)}{E_p[e^{\theta X}]} \tag{3.1}$$

check q is a probability distribution, i.e.
- >= 0
- sum to 1

where $\theta \in \mathbb{R}$ is known as the twisting parameter. Note that when $\theta = 0$ we recover $p(x)$ (i.e. no twisting is performed).

3.3 Glasserman and Li Importance Sampling

In this section we give a very brief overview of the algorithm described in Glasserman and Li as well as some details regarding how to implement it in practice. For more details regarding their method see either Glasserman and Li [5] or Wang [10].

For the outer $\mathcal{Z}$ level they attempt to find a vector $\vec{\mu}$ to shift the mean of the $\mathcal{Z}$ distribution from $\mathcal{N}(\vec{0}, I)$ to $\mathcal{N}(\vec{\mu}, I)$. Their goal is to perform importance sampling by finding a $\vec{\mu}$ that reduces variance. $\vec{\mu}$ is selected by solving an optimization problem:

minimize the upper bound on the variance instead of minimizing the variance

$$\vec{\mu} = \arg\max_{z}\{\min_{\theta}\{-\theta(z)\ell + \psi(z, \theta(z))\} - \frac{1}{2}z^T z\} \tag{3.2}$$

where would want to compute partial derivatives

$$\psi(z, \theta) = \sum_{n=1}^{N} \ln\left(\sum_{c=1}^{C} p_n^c(z) e^{\theta \omega_n^c}\right)$$ inner summation equal E_p(e^{theta X}) ?

Note that for the purposes of generality of the results in sections 3.3.1 and 3.3.2 we present $\psi(z, \theta)$ in it's multi-credit-state form. The expression used by Glasserman and Li can be recovered by setting $C = 2$, $\omega_n^2 = 0$, and noting that $p_n^2(z) = 1 - p_n^1(z)$. note w_n^2 corresponds to c_k in Glasserman&Li

For the inner $\mathcal{E}$ level they use exponential twisting to find a $\theta(z)$ value that minimizes variance. This involves solving

$$\theta(z) = \operatorname*{argmin}_{\theta}\{-\theta\ell + \psi(z, \theta)\}$$ basically the second moment of c.g.f.

e^x increasing to minimizing x equiv to minimizing e^x

for each z you have sampled. Once found the default indicators $\{\mathbb{1}_n^1(z)\}_{n=1}^N$ are sampled from the twisted probability distribution

state=1 is default state…

$$q_n^1(\theta) = \frac{p_n^1(z) e^{\theta \omega_n^1}}{1 + p_n^1(z)(e^{\theta \omega_n^1} - 1)}$$ m.g.f. of bernoulli

proposal distribution from the exponentially tiled family of distributions

For more details on how this exponential twisting is performed see chapter 5 where we extend it to the multi-credit-state model.

Wang [10] found that the numerical method they used to solve (3.2) frequently failed to converge. We also initially encountered this issue and believe it is likely due to using divided differences in the optimization method used to solve (3.2) numerically. The rest of this chapter is dedicated to finding analytical expressions for the derivatives associated with (3.2). This allows us to stabilize and speed up solving this nested optimization problem.

3.3.1 Constrained Optimization for Bi-level Optimization Problems

First we introduce some notation

$$f^U(z,\theta) = -\theta\ell + \psi(z,\theta) - \frac{1}{2}z^T z$$
$$f^L(z,\theta) = -\theta\ell + \psi(z,\theta)$$

Then (3.2) can be written as

$$\vec{\mu} = \arg\max_z f^U(z,\theta) \tag{3.3}$$
$$\text{subject to } \theta = \operatorname*{argmin}_{\theta'} f^L(z,\theta') \tag{3.4}$$

This in turn can be written as a constrained optimization problem

$$\vec{\mu} = \arg\max_z f^U(z,\theta) \tag{3.5}$$
$$\text{subject to } \frac{\partial f^L(z,\theta)}{\partial\theta} = 0 \tag{3.6}$$

We can use a standard constrained optimization method, such as fmincon in matlab, to solve (3.5)-(3.6).

3.3.2 Gradients in Bi-level Optimization Problems

While casting (3.2) as a constrained optimization problem provides an effective approach to solve (3.2), speed became a concern. To provide a speed up we provided analytic gradients of both the objective function and constraint.

$$
\begin{aligned}
\nabla_z f^U &= \nabla_z \psi(z,\theta) - z \\
\frac{\partial f^U}{\partial \theta} &= \ell - \frac{\partial \psi}{\partial \theta} \\
\nabla_z \frac{\partial f^L}{\partial \theta} &= \nabla_z \frac{\partial \psi}{\partial \theta} \\
\frac{\partial^2 f^L}{\partial \theta^2} &= \frac{\partial^2 \psi}{\partial \theta^2} \\
\nabla_z \psi(z,\theta) &= \sum_{n=1}^{N} \frac{\sum_{c=1}^{C} \nabla_z p_n^c(z) e^{\theta \omega_n^c}}{\sum_{c=1}^{C} p_n^c(z) e^{\theta \omega_n^c}} \\
\frac{\partial \psi}{\partial \theta} &= \sum_{n=1}^{N} \frac{\sum_{c=1}^{C} p_n^c(z) \omega_n^c e^{\theta \omega_n^c}}{\sum_{c=1}^{C} p_n^c(z) e^{\theta \omega_n^c}} \\
\nabla_z \frac{\partial \psi}{\partial \theta} &= \sum_{n=1}^{N} \frac{\left(\sum_{c=1}^{C} \nabla_z p_n^c(z) \omega_n^c e^{\theta \omega_n^c}\right)\left(\sum_{c=1}^{C} p_n^c(z) e^{\theta \omega_n^c}\right)}{\left(\sum_{c=1}^{C} p_n^c(z) e^{\theta \omega_n^c}\right)^2} - \sum_{n=1}^{N} \frac{\left(\sum_{c=1}^{C} \nabla_z p_n^c(z) e^{\theta \omega_n^c}\right)\left(\sum_{c=1}^{C} p_n^c(z) \omega_n^c e^{\theta \omega_n^c}\right)}{\left(\sum_{c=1}^{C} p_n^c(z) e^{\theta \omega_n^c}\right)^2} \\
\frac{\partial^2 \psi}{\partial \theta^2} &= \sum_{n=1}^{N} \frac{\left(\sum_{c=1}^{C} p_n^c(z) (\omega_n^c)^2 e^{\theta \omega_n^c}\right)\left(\sum_{c=1}^{C} p_n^c(z) e^{\theta \omega_n^c}\right)}{\left(\sum_{c=1}^{C} p_n^c(z) e^{\theta \omega_n^c}\right)^2} - \sum_{n=1}^{N} \frac{\left(\sum_{c=1}^{C} p_n^c(z) \omega_n^c e^{\theta \omega_n^c}\right)^2}{\left(\sum_{c=1}^{C} p_n^c(z) e^{\theta \omega_n^c}\right)^2} \\
\nabla_z p_n^c(z) &= \frac{-1}{\sqrt{2\pi}} \left(\exp\left(\frac{-1}{2} \left(\frac{H_{c(n)}^c - \beta_n^T z}{\sqrt{1 - \beta_n^T \beta_n}} \right)^2 \right) - \exp\left(\frac{-1}{2} \left(\frac{H_{c(n)}^{c-1} - \beta_n^T z}{\sqrt{1 - \beta_n^T \beta_n}} \right)^2 \right) \right) \frac{\vec{\beta}_n}{\sqrt{1 - \vec{\beta}_n^T \vec{\beta}_n}}
\end{aligned}
$$

Gould et al. [6] explain how to solve these kinds of bi-level optimization problems using gradient ascent. Using the above gradients the update rule is

$$
z \leftarrow z + \eta \left(\nabla_z f^U + \frac{\partial f^U}{\partial \theta} \nabla_z \theta \right)
$$

where

$$
\nabla_z \theta = - \frac{\nabla_z \frac{\partial f^L}{\partial \theta}}{\frac{\partial^2 f^L}{\partial \theta^2}}
$$

and η is a user specified step size parameter. We found that computing $\vec{\mu}$ in this manner was a useful way to check our constrained optimization approach.

Chapter 4

Outer Level Importance Sampling

4.1 Normal Approximation

Based on the work on Han [7] we have the following convergence result.

Theorem 1 *Conditional on $\mathcal{Z} = z$ if the following conditions hold*

1. *$\exists \delta > 0$ such that $sup\{|\omega_n|\} = O(N^{-\frac{1}{2}-\delta})$*
2. *$\exists M \in [0, \infty)$ such that $z \in D^S = [-M, M] \times \cdots [-M, M]$*

then the normalized conditional portfolio loss converges in distribution to a standard normal random variable

$$\frac{\mathcal{L}(z, \mathcal{E}) - \mu(z)}{\sigma(z)} \xrightarrow{d} \mathcal{N}(0, 1)$$

as $N \to \infty$, where $\mu(z) = \mathbb{E}_{\phi_N}[\mathcal{L}(z, \mathcal{E})]$ and $\sigma^2(z) = \mathbb{V}_{\phi_N}[\mathcal{L}(z, \mathcal{E})]$

Both these assumptions are easily satisfied for our model. By taking the expectation and variance of (1.4) Han [7] arrives at closed form expressions for $\mu(z)$ and $\sigma(z)$

$$\begin{aligned}
\mu(z) &= \mathbb{E}_{\phi_N}[\mathcal{L}(z, \mathcal{E})] \\
&= \sum_{n=1}^{N} \sum_{c=1}^{C} \omega_n^c \mathbb{E}_{\phi_N} \left[\mathbb{1}_{\left\{ \frac{H_{c(n)}^{c-1} - \beta_n^T \mathcal{Z}}{\sqrt{1-\beta_n^T \beta_n}} \leq \epsilon_n \leq \frac{H_{c(n)}^{c-1} - \beta_n^T \mathcal{Z}}{\sqrt{1-\beta_n^T \beta_n}} \right\}} \right] \\
&= \sum_{n=1}^{N} \sum_{c=1}^{C} \omega_n^c \left(\Phi \left(\frac{H_{c(n)}^{c} - \beta_n^T \mathcal{Z}}{\sqrt{1-\beta_n^T \beta_n}} \right) - \Phi \left(\frac{H_{c(n)}^{c-1} - \beta_n^T \mathcal{Z}}{\sqrt{1-\beta_n^T \beta_n}} \right) \right) \\
&= \sum_{n=1}^{N} \sum_{c=1}^{C} \omega_n^c p_n^c(z)
\end{aligned} \tag{4.1}$$

Similarly

$$\begin{aligned}\sigma^2(z) &= \mathbb{V}_{\phi_N}[\mathcal{L}(z,\mathcal{E})] \\ &= \sum_{n=1}^{N}\omega_n^2\left(\frac{1}{2}\sum_{a=1}^{C}\sum_{b=1}^{C}(LGC_n^a - LGC_n^b)^2 p_n^a(z)p_n^b(z)\right) \\ &= \sum_{n=1}^{N}\omega_n^2\left(\sum_{a>b}^{C}(LGC_n^a - LGC_n^b)^2 p_n^a(z)p_n^b(z)\right)\end{aligned} \tag{4.2}$$

Under this approximation

$$P(\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell | \mathcal{Z} = z) \approx 1 - \Phi\left(\frac{\ell - \mu(z)}{\sigma(z)}\right) \tag{4.3}$$

so

$$P(\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell) \approx \int_{\mathbb{R}^S}\left(1 - \Phi\left(\frac{\ell - \mu(z)}{\sigma(z)}\right)\right) d(z \sim \phi_S) \tag{4.4}$$

This leads the the following MC integration scheme

Algorithm 4.1: 1 level Monte Carlo integration

Inputs: NZ, $\{EAD_n\}_{n=1}^{N}$, $\{LGC_n^c\}_{n=1,c=1}^{N,C}$, β, H, ℓ
  Sample $\{z_i\}_{i=1}^{NZ}$ from $\mathcal{N}(0, I_S)$
  FOR $i = 1 : NZ$
    Compute $\mu(z_i)$ and $\sigma(z_i)$
    Store $a_i = 1 - \Phi\left(\frac{\ell - \mu(z_i)}{\sigma(z_i)}\right)$
  Return mean(a_i)

This method has the advantage of completely removing any need to sample $\mathcal{E}$. However unlike the other estimators we have introduced a bias towards this standard normal random variable, which is only an approximation of $\mathcal{L}_N$ for finite N. Regardless, in practice we have found for large N ($N \approx 2500$) this result is close to the correct answer.

In the coming sections we build on this result to improve Algorithm 2.2, thereby removing the bias.

4.2 $\mathcal{Z}$ Importance Sampling

4.2.1 Zero Variance Importance Sampling

Continuing with the notation from section 3.1, assuming $\forall z,\ f(z) \geq 0$ the optimal importance sampler would be $\pi_o(z) = cf(z)p(z)$. This would reduce the variance to 0 so we naturally refer to it as the zero variance importance sampler.

$$V_{\pi_o}\left[\frac{f(\mathcal{Z})p(\mathcal{Z})}{\pi_o(\mathcal{Z})}\right] = V_{\pi_o}\left[\frac{1}{c}\right] = 0$$

In our case $\pi_o(z) = cP(\mathcal{L}_N(z,\mathcal{E}) \geq \ell)\phi_S(z)$.

However finding the appropriate normalization constant c is often as difficult as solving the problem we are tasked with. Instead we build a different importance sampler $\pi^*(z) \approx \pi_o(z)$.

4.2.2 Approximating the Zero Variance Importance Sampler

Recall that under the normal approximation of section 4.1 the probability the normalized loss is greater than ℓ is

$$\begin{aligned} P(\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell) &\approx \int_{\mathbb{R}^S} \left(1 - \Phi\left(\frac{\ell - \mu(z)}{\sigma(z)}\right)\right) d(z \sim \phi_S) \\ &= \int_{\mathbb{R}^S} \left(1 - \Phi\left(\frac{\ell - \mu(z)}{\sigma(z)}\right)\right) \phi_S(z) dz \\ &= E_{\phi_S}\left[1 - \Phi\left(\frac{\ell - \mu(\mathcal{Z})}{\sigma(\mathcal{Z})}\right)\right] \end{aligned}$$

Therefore the zero variance importance sampler for the biased estimator is

$$\pi(z) = c\left(1 - \Phi\left(\frac{\ell - \mu(z)}{\sigma(z)}\right)\right)\phi_S(z) \tag{4.5}$$

Following the work of Wang[10] we make an important observation. Although we do not know the constant c that makes $\pi(z)$ a normalized pdf that does not prevent us from sampling from $\pi(z)$. Markov Chain Monte Carlo (MCMC) methods [2] often only require the un-normalized pdf $\hat{\pi}(z) = \left(1 - \Phi\left(\frac{\ell-\mu(z)}{\sigma(z)}\right)\right)\phi_S(z)$ to draw samples from. So after selecting ones preferred MCMC technique one can sample the zero variance important sampler.

Our goal now is to train an approximator for $\pi(z)$. Again, as in Wang[10] we use a Gaussian Mixture Model (also sometimes refereed to as a Mixture of Gaussians).

$$\pi^*(z) = \sum_{k=1}^{K} r_k \frac{1}{\sqrt{|2\pi\Sigma_k|}} \exp(-\frac{1}{2}(z-\mu_k)^T\Sigma_k^{-1}(z-\mu_k)) \tag{4.6}$$

A Gaussian Mixture Model (GMM) is a weighted sum of (multivariate) Gaussian pdfs. Each one has its own mean μ_k and covariance matrix Σ_k. The weights r_k ensure that the resulting sum is a normalized pdf. That is $r_k \geq 0$ and $\sum_{k=1}^{K} r_k = 1$. The number of Gaussians K is a hyperparemeter for the model that must be hand selected.

With this definition of $\pi^*(z)$ and our samples from $\pi(z)$ we can train $\pi^*(z)$ to approximate $\pi(z)$ using Expectation Maximization [10].

This suggests the following algorithm

Algorithm 4.2: 1 level Monte Carlo integration with $\mathcal{Z}$ importance sampling

Inputs: NZ, NZ^* $\{EAD_n\}_{n=1}^{N}$,$\{LGC_n^c\}_{n=1,c=1}^{N,C}$, β, H, ℓ, K
  Define $\hat{\pi}(z) = \left(1 - \Phi\left(\frac{\ell-\mu(z)}{\sigma(z)}\right)\right)\phi_S(z)$
  Sample $\{z_i\}_{i=1}^{NZ}$ from $\hat{\pi}(z)$ using MCMC
  Define $\pi^*(z) = \sum_{k=1}^{K} r_k \frac{1}{\sqrt{|2\pi\Sigma_k|}} \exp(-\frac{1}{2}(z-\mu_k)^T\Sigma_k^{-1}(z-\mu_k))$
  Train $\pi^*(z)$ on $\{z_i\}_{i=1}^{NZ}$ using Expectation Maximization
  Sample $\{z_i^*\}_{i=1}^{NZ^*}$ from π^*
  FOR $i = 1 : NZ^*$

8 Compute $\mu(z_i^*)$ and $\sigma(z_i^*)$
9 Store $a_i = \left(1 - \Phi\left(\frac{\ell - \mu(z_i^*)}{\sigma(z_i^*)}\right)\right) \frac{\phi_S(z_i^*)}{\pi^*(z_i^*)}$
10 Return mean(a_i)

Algorithm 4.2 is the $\mathcal{Z}$ importance sampler from Wang [10]. It is an improvement over Algorithm 4.1 however it still suffers from the bias introduced by the normal approximation. For more details on this bias see [10].

Given theorem 4.1 it should not be very surprising that for large N if $\pi^*(z)$ is a good approximation to $\pi(z)$ then it is also a good approximation to $\pi_o(z)$. Motivation for this statement is contained in Appendix A. Thus we can merge Algorithm 4.2 with Algorithm 2.2 to produce a new algorithm that uses the normal approximation to perform importance sampling, but converges to the correct answer.

Algorithm 4.3: 2 level Monte Carlo integration with $\mathcal{Z}$ importance sampling

1 Inputs: NZ, NZ^*, NE, $\{EAD_n\}_{n=1}^N$, $\{LGC_n^c\}_{n=1,c=1}^{N,C}$, β, H, ℓ, K
2 Define $\hat{\pi}(z) = \left(1 - \Phi\left(\frac{\ell - \mu(z)}{\sigma(z)}\right)\right) \phi_S(z)$
3 Sample $\{z_i\}_{i=1}^{NZ}$ from $\hat{\pi}(z)$ using MCMC
4 Define $\pi^*(z) = \sum_{k=1}^{K} r_k \frac{1}{\sqrt{|2\pi\Sigma_k|}} \exp(-\frac{1}{2}(z-\mu_k)^T \Sigma_k^{-1}(z-\mu_k))$
5 Train $\pi^*(z)$ on $\{z_i\}_{i=1}^{NZ^*}$ using Expectation Maximization
6 Sample $\{z_i^*\}_{i=1}^{NZ^*}$ from π^*
7 FOR $i = 1 : NZ^*$
8 Compute $\{p_n^c(z_i^*)\}_{n=1,c=1}^{N,C}$
9 FOR $j = 1 : NE$
10 Sample $\{\mathbb{1}_n^c\}_{n=1,c=1}^{N,C}$ according to $\{p_n^c(z_i^*)\}_{n=1,c=1}^{N,C}$
11 Compute $\mathcal{L}_{ij} = \sum_{n=1}^{N} \sum_{c=1}^{C} \omega_n^c \mathbb{1}_n^c$
12 Store $a_{ij} = \mathbb{1}_{\{\mathcal{L}_{ij} > \ell\}} \frac{\phi_S(z_i^*)}{\pi^*(z_i^*)}$
13 Return mean(a_{ij})

In summary we've introduced the following new functions

$\pi_o(z)$	True zero variance importance sampler
$\pi(z)$	Approximation to $\pi_o(z)$ under the normal approximation of theorem 1
$\hat{\pi}(z)$	Unnormalized version of $\pi(z)$
$\pi^*(z)$	Gaussian Mixture Model trained on samples from $\hat{\pi}(z)$ to approximate $\pi(z)$

Chapter 5

Inner Level Importance Sampling

5.1 Discrete Importance Sampling

In the inner layer, conditional on $\mathcal{Z} = z$, we want to sample $\{\mathbb{1}_n^c\}_{n=1,c=1}^{N,C}$ from $p(\mathbb{1}_1^1, \cdots, \mathbb{1}_1^C, \mathbb{1}_2^1, \cdots, \mathbb{1}_2^C, \cdots, \mathbb{1}_N^1, \cdots, \mathbb{1}_N^C)$. Recalling (2.3) we know that we can break this large pdf into a product of smaller pdfs:

$$p(\mathbb{1}_1^1, \cdots, \mathbb{1}_1^C, \mathbb{1}_2^1, \cdots, \mathbb{1}_2^C, \cdots, \mathbb{1}_N^1, \cdots, \mathbb{1}_N^C) = \prod_{n=1}^{N} p_n(\mathbb{1}_n^1, \cdots, \mathbb{1}_n^C)$$

In order to perform importance sampling we seek a series of probability distributions $q_n(\mathbb{1}_n^1, \cdots, \mathbb{1}_n^C)$ to sample $\{\mathbb{1}_n^c\}_{n=1,c=1}^{N,C}$ from that reduce the variance of our samples. Given these q_n we compute

likelihood!

$$E_p[\mathbb{1}_{\{\mathcal{L}_N > l\}}] = E_q\left[\prod_{n=1}^{N} \frac{p_n(\mathbb{1}_n^1, \cdots, \mathbb{1}_n^C)}{q_n(\mathbb{1}_n^1, \cdots, \mathbb{1}_n^C)} \mathbb{1}_{\{\mathcal{L}_N > l\}}\right]$$

Since we are dealing with discrete random variables, $p_n(\mathbb{1}_n^1, \cdots, \mathbb{1}_n^C)$ is really shorthand for the set of real numbers

$$p_n(1, 0, 0, \cdots, 0) = p_n^1(z) \tag{5.1}$$
$$p_n(0, 1, 0, \cdots, 0) = p_n^2(z) \tag{5.2}$$
$$\vdots \tag{5.3}$$
$$p_n(0, 0, 0, \cdots, 1) = p_n^C(z) \tag{5.4}$$

where $p_n^c(z)$ is defined in (2.2).

Similarly $q_n(\mathbb{1}_n^1, \cdots, \mathbb{1}_n^C)$ is really shorthand for a set of real numbers

$$q_n(1,0,0,\cdots,0) = q_n^1(z) \tag{5.5}$$
$$q_n(0,1,0,\cdots,0) = q_n^2(z) \tag{5.6}$$
$$\vdots \tag{5.7}$$
$$q_n(0,0,0,\cdots,1) = q_n^C(z) \tag{5.8}$$

In the context of Monte Carlo Integration this means that if we sample such that $\mathbb{1}_n^k = 1$ (and therefore $\mathbb{1}_n^{c\neq k} = 0$) then

$$\frac{p_n(\mathbb{1}_n^1, \cdots, \mathbb{1}_n^C)}{q_n(\mathbb{1}_n^1, \cdots, \mathbb{1}_n^C)} = \frac{p_n^k}{q_n^k}$$

. This can be compactly written as

$$\prod_{n=1}^{N} \frac{p_n(\mathbb{1}_n^1, \cdots, \mathbb{1}_n^C)}{q_n(\mathbb{1}_n^1, \cdots, \mathbb{1}_n^C)} = \prod_{n=1}^{N}\prod_{c=1}^{C} \left(\frac{p_n^c}{q_n^c}\right)^{\mathbb{1}_n^c}$$

5.2 Inner Level Exponential Twisting

Based on the work of Glasserman and Li [5] we consider an approach based on Exponential Twisting [9]. Recall from section 3.2 that when performing an exponential twist one is generating a new pdf $q(x)$ from an existing pdf $p(x)$ like so

$$q(x,\theta) = \frac{e^{\theta x}p(x)}{E_p[e^{\theta X}]}$$

In our case this becomes

$$q_n^c(z,\theta) = \frac{p_n^c(z)e^{\theta\omega_n^c}}{\sum_{k=1}^{C} p_n^k(z)e^{\theta\omega_n^k}} \tag{5.9}$$

where our random variables are the W_n from (2.4). The likelihood ratio becomes

$$\prod_{n=1}^{N}\prod_{c=1}^{C} \left(\frac{p_n^c(z)}{q_n^c(z,\theta)}\right)^{\mathbb{1}_n^c} = e^{-\theta\mathcal{L}+\psi(z,\theta)} \tag{5.10}$$

in appendix B

$$\psi(z,\theta) = \sum_{n=1}^{N} \ln\left(\sum_{c=1}^{C} p_n^c(z)e^{\theta\omega_n^c}\right) \tag{5.11}$$

when theta=0, phi=0, phi pass through origin, and the upper bound is minimized

For more details on this calculation see Appendix B. Our new unbiased random variable is therefore

$$\mathbb{1}_{\{\mathcal{L}>\ell\}}e^{-\theta\mathcal{L}+\psi(z,\theta)}$$

The second moment of this new random variable is

$$E_q[\mathbb{1}_{\{\mathcal{L}>\ell\}}e^{-2\theta\mathcal{L}+2\psi(z,\theta)}] \leq e^{-2\theta\ell+2\psi(z,\theta)}$$

Where the bound holds for $\theta \geq 0$. The left hand side is hard to minimize directly, although recent work has been done along those lines [4]. The upper bound however is easy to minimize. Note, for fixed z $-\theta\ell + \psi(z,\theta)$ is a convex function of θ that passes through the origin. In fact, for any reasonable selection of model parameters it is strictly convex. A proof of this exists in Appendix C. Hence our optimal θ value is

inner level proposal distribution can be solved optimally

$$\theta^* = \begin{cases} \text{the unique solution to} \frac{\partial\psi}{\partial\theta} = \ell & \ell > \left.\frac{\partial\psi}{\partial\theta}\right|_{\theta=0} \\ 0 & \ell \leq \left.\frac{\partial\psi}{\partial\theta}\right|_{\theta=0} \end{cases}$$

dont understand this…

Using this technique we arrive at a new importance sampling algorithm

Algorithm 5.1: 2 level Monte Carlo integration with $\mathcal{E}$ importance sampling

Inputs: NZ, NE, $\{EAD_n\}_{n=1}^{N}$, $\{LGC_n^c\}_{n=1,c=1}^{N,C}$, β, H, ℓ
  Sample $\{z_i\}_{i=1}^{NZ}$ from $\mathcal{N}(0, I_S)$
  Compute $\{p_n^c(z_i)\}_{n=1,c=1}^{N,C}$
  Define $\psi(\theta, z) = \sum_{n=1}^{N} \ln\left(\sum_{c=1}^{C} p_n^c(z) e^{\theta\omega_n^c}\right)$
  FOR $i = 1 : NZ$:
    IF $\sum_{n=1}^{N}\sum_{c=1}^{C} \omega_n^c p_n^c(z_i) > \ell$
      Set $\theta = \underset{\theta}{\operatorname{argmin}}\{-\theta\ell + \psi(\theta, z_i)\}$ find optimal proposal distribution for inner sampling
    ELSE
      Set $\theta = 0$
    Compute $q_n^c(z_i, \theta) = \frac{p_n^c(z_i) e^{\theta\omega_n^c}}{\sum_{k=1}^{C} p_n^k(z_i) e^{\theta\omega_n^k}}$ for n=1,...,N and c=1,...,C
    FOR $j = 1 : NE$:
      Sample $\{\mathbb{1}_n^c\}_{n=1,c=1}^{N,C}$ according to $q_n^c(z_i, \theta)$
      Compute $\mathcal{L}_{ij} = \sum_{n=1}^{N}\sum_{c=1}^{C} \omega_n^c \mathbb{1}_n^c$
      Store $a_{ij} = \mathbb{1}_{\{\mathcal{L}_{ij} > \ell\}} e^{-\theta\mathcal{L}_{ij} + \psi(\theta, z_i)}$
  Return Mean(a_{ij})

5.2.1 Final Algorithm

Combining Algorithm 5.1 with Algorithm 4.3 we arrive at the final algorithm

Algorithm 5.2: 2 level Monte Carlo integration with 2 level importance sampling

Inputs: NZ, NZ^*, NE, $\{EAD_n\}_{n=1}^{N}$, $\{LGC_n^c\}_{n=1,c=1}^{N,C}$, β, H, ℓ, K
  Define $\hat{\pi}(z) = \left(1 - \Phi\left(\frac{l-\mu(z)}{\sigma(z)}\right)\right)\phi_S(z)$
  Sample $\{z_i\}_{i=1}^{NZ}$ from $\hat{\pi}(z)$ using MCMC
  Define $\pi^*(z) = \sum_{k=1}^{K} r_k \frac{1}{\sqrt{|2\pi\Sigma_k|}} \exp(-\frac{1}{2}(z-\mu_k)^T\Sigma_k^{-1}(z-\mu_k))$
  Train $\pi^*(z)$ on $\{z_i\}_{i=1}^{NZ}$ using Expectation Maximization
  Sample $\{z_i^*\}_{i=1}^{NZ^*}$ from π^*
  Compute $\{p_n^c(z_i^*)\}_{n=1,c=1}^{N,C}$

Define $\psi(\theta, z) = \sum_{n=1}^{N} \ln\left(\sum_{c=1}^{C} p_n^c(z) e^{\theta \omega_n^c}\right)$

FOR $i = 1 : NZ^*$:

  IF $\sum_{n=1}^{N} \sum_{c=1}^{C} \omega_n^c p_n^c(z_i^*) > \ell$

    Set $\theta = \underset{\theta}{\operatorname{argmin}}\{-\theta\ell + \psi(\theta, z_i^*)\}$

  ELSE

    Set $\theta = 0$

  Compute $q_n^c(z_i^*, \theta) = \frac{p_n^c(z_i^*) e^{\theta \omega_n^c}}{\sum_{k=1}^{C} p_n^k(z_i^*) e^{\theta \omega_n^k}}$ for n = 1 ,... ,N and c = 1 ,... ,C

  FOR $j = 1 : NE$:

    Sample $\{\mathbb{1}_n^c\}_{n=1,c=1}^{N,C}$ according to $q_n^c(z_i^*, \theta)$

    Compute $\mathcal{L}_{ij} = \sum_{n=1}^{N} \sum_{c=1}^{C} \omega_n^c \mathbb{1}_n^c$

    Store $a_{ij} = \mathbb{1}_{\{\mathcal{L}_{ij} > \ell\}} \frac{\phi_S(z_i^*)}{\pi^*(z_i^*)} e^{-\theta \mathcal{L}_{ij} + \psi(\theta, z_i^*)}$

Return Mean(a_{ij})

Chapter 6

Numerical Results

Here we put the various algorithms discussed so far to the test. The following notation is used throughout this chapter:

2LvlMC(nZ,nE)	=	Algorithm 2.2 with nZ $\mathcal{Z}$ samples and nE $\mathcal{E}$ samples per $\mathcal{Z}$ sample
1LvlISCLT(nZ)	=	Algorithm 4.2 with nZ $\mathcal{Z}$ samples
1LvlISZ(nZ,nE)	=	Algorithm 4.3 with nZ $\mathcal{Z}$ samples and nE $\mathcal{E}$ samples per $\mathcal{Z}$ sample
1LvlISE(nZ,nE)	=	Algorithm 5.1 with nZ $\mathcal{Z}$ samples and nE $\mathcal{E}$ samples per $\mathcal{Z}$ sample
2LvlIS(nZ,nE)	=	Algorithm 5.2 with nZ $\mathcal{Z}$ samples and nE $\mathcal{E}$ samples per $\mathcal{Z}$ sample

For 1LvlISZ(nZ,nE) and 2LvlIS(nZ,nE) 600 samples were used to train π^*

6.1 Binary Credit State Experiments

In order to compare our methods against established literature [10][5] we first run experiments in the binary credit state model. In this model there are 2 credit states, default and not-default. All creditors start in the not-default state and have an individual probability of defaulting. Note that this is a slight deviation from the multi credit state model where the transition probabilities between credit states does not vary between creditors, as seen in Table 1.1. If a creditor remains in the not-default state then no loss/gain is induced.

6.1.1 Missing Glasserman and Li Data

Unfortunately although we managed to improve on the implementation of the Glassermand and Li importance sampler used by Wang [10] we still suffer from the same issue as they did. Namely that the answer seems to be biased downward. We had hoped that the reason Wang saw this was because their implementation frequently failed to converge. However that does not appear to be the case as we still observe it after resolving the convergence issue. We do find it quite strange that this is continuing to happen, as the theory of importance sampling forbids a bias. Moreover numerical simulations report that the variance is being reduced as we would expect, so the method appears to be working. Since we could not determine the cause of this anomaly we have elected to exclude our version of the Glasserman and Li technique from our numerical report. Our implementation can be found online here.

6.1.2 Model Parameter Settings

The formulae used to generate the model parameters are the following

$$
\begin{aligned}
p_n &= 0.01\left(1+\sin\left(\frac{16\pi n}{N}\right)\right) && n=1,\cdots,N \\
\beta_{nj} &\sim \text{Unif}\left(-\frac{1}{\sqrt{S}},\frac{1}{\sqrt{S}}\right) && n=1,\cdots,N \qquad j=1,\cdots,S \\
LGC_n^D &= \left\lfloor \frac{5k}{N} \right\rfloor^2 && n=1,\cdots,N \\
EAD_n &\sim \text{Unif}(0.5,1.5) && n=1,\cdots,N
\end{aligned}
$$

The exact values drawn for β and EAD_k are included in the code repositories as param.m files.

6.1.3 Experiment Setup

In the following experiments the true answer was obtained by running Algorithm 2.2 with $10,000$ $\mathcal{E}$ samples for the inner layer, and the outer $\mathcal{Z}$ layer being allowed to run overnight for approximately 12 hours.

Each method being examined was run 5 times to help reduce outlier induced errors.

Except where otherwise noted all graphs use the same range for the vertical axis. To facilitate this the data is transformed by applying log base 10. Dot plots are values for individual runs and solid continuous lines are values averaged across all 5 runs. Dashed lines delineate the value being compared against, the meaning of which varies from section to section.

Tables contain values that are averaged over all 5 runs. The variance ratio column refers to the variance of the method in question divided by the variance from the corresponding runs of 2LvlMC(nZ,nE). So the variance ratio of 1LvlIS(300,300) is the average variance of 1LvlIS(300,300) divided by the average variance of 2LvlMC(300,300). Similarly for 1LvlIS(400,400) and so on.

6.1.4 S,ℓ Dependency

In this subsection we explore the dependency of our methods on S and ℓ. Here the dashed lines represent an approximation to the true answer obtained by running 2LvlMC overnight.

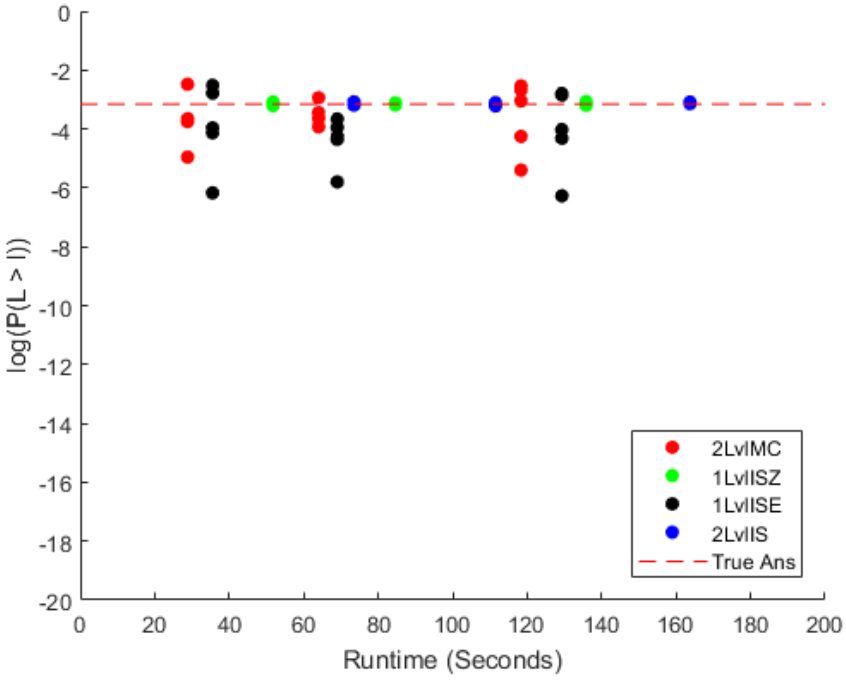

Figure 6.1: $S = 5$ $\ell = 0.3$

Method	Runtime(sec)	Absolute Error	Relative Error
2LvlMC(300,300)	28	1.00e-3	1.42
2LvlMC(400,400)	64	4.63e-4	6.60e-1
2LvlMC(500,500)	118	1.02e-3	1.45
1LvlISZ(300,300)	51	8.89e-5	1.26e-1
1LvlISZ(400,400)	84	4.45e-5	6.34e-2
1LvlISZ(500,500)	135	6.56e-5	9.34e-2
1LvlISE(300,300)	35	1.03e-3	1.47
1LvlISE(400,400)	69	6.15e-4	8.76e-1
1LvlISE(500,500)	129	7.16e-4	1.02
2LvlIS(300,300)	73	5.11e-5	7.28e-2
2LvlIS(400,400)	111	5.49e-5	7.82e-2
2LvlIS(500,500)	163	4.15e-5	5.92e-2

Table 6.1: Error and variance for $S = 5$, $\ell = 0.3$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 7.02e−4

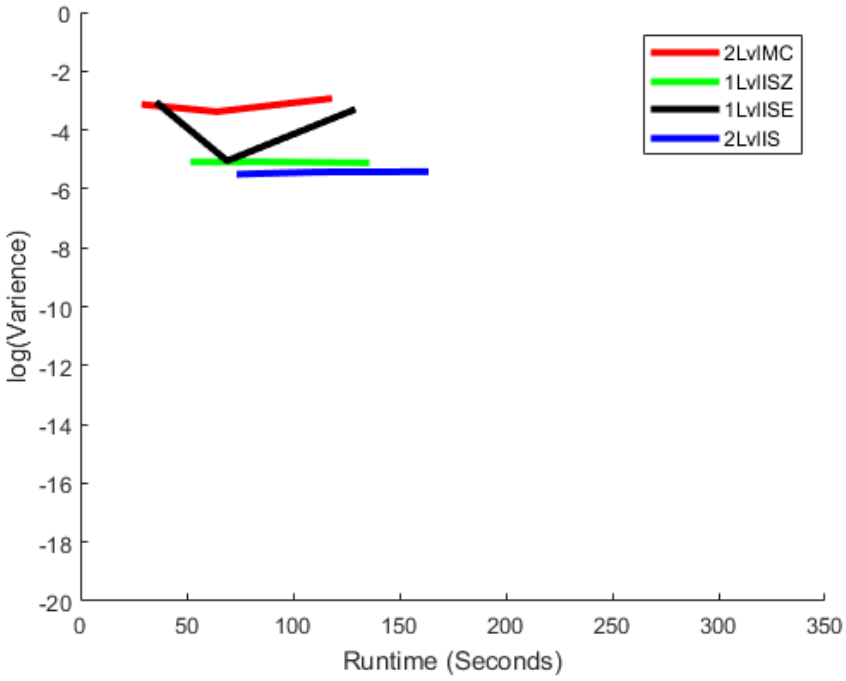

Figure 6.2: $S = 5$ $\ell = 0.3$

Method	Runtime(sec)	Variance	Variance Ratio	Standard Error
2LvlMC(300,300)	28	7.42e-4	100%	5.44e-5
2LvlMC(400,400)	64	4.17e-4	100%	4.08e-5
2LvlMC(500,500)	118	1.18e-3	100%	6.87e-5
1LvlISZ(300,300)	51	8.12e-6	1.09%	5.69e-6
1LvlISZ(400,400)	84	8.27e-6	1.98%	5.75e-6
1LvlISZ(500,500)	135	7.64e-6	0.64%	5.53e-6
1LvlISE(300,300)	35	8.94e-4	120.53%	5.98e-5
1LvlISE(400,400)	69	8.93e-6	2.14%	5.97e-6
1LvlISE(500,500)	129	5.06e-4	42.88%	4.5e-5
2LvlIS(300,300)	73	3.13e-6	0.42%	3.54e-6
2LvlIS(400,400)	111	3.73e-6	0.89%	3.86e-6
2LvlIS(500,500)	163	3.83e-6	0.32%	3.91e-6

Table 6.2: Variance reduction for $S = 5$, $\ell = 0.3$

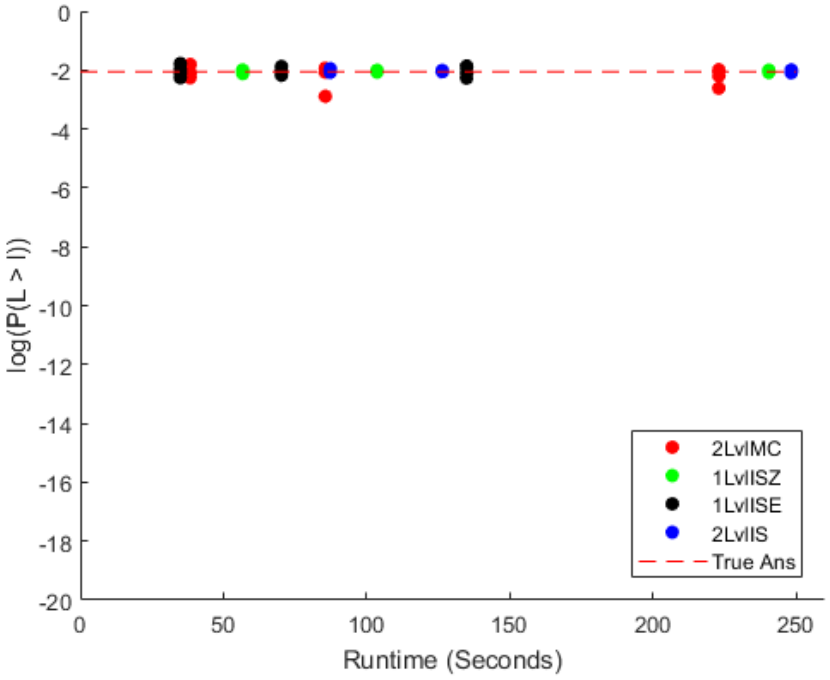

Figure 6.3: $S = 5$ $\ell = 0.2$

Method	Runtime(sec)	Absolute Error	Relative Error
2LvlMC(300,300)	38	3.94e-3	4.51e-1
2LvlMC(400,400)	85	2.39e-3	2.73e-1
2LvlMC(500,500)	222	2.31e-3	2.64e-1
1LvlISZ(300,300)	56	8.07e-4	9.23e-2
1LvlISZ(400,400)	70	5.27e-4	6.02e-2
1LvlISZ(500,500)	134	5.03e-4	5.75e-2
1LvlISE(300,300)	34	2.97e-3	3.40e-1
1LvlISE(400,400)	103	2.02e-3	2.31e-1
1LvlISE(500,500)	240	2.67e-3	3.05e-1
2LvlIS(300,300)	87	1.05e-3	1.20e-1
2LvlIS(400,400)	126	5.84e-4	6.68e-2
2LvlIS(500,500)	248	4.97e-4	5.68e-2

Table 6.3: Error and variance for $S = 5$, $\ell = 0.2$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 8.74e−3

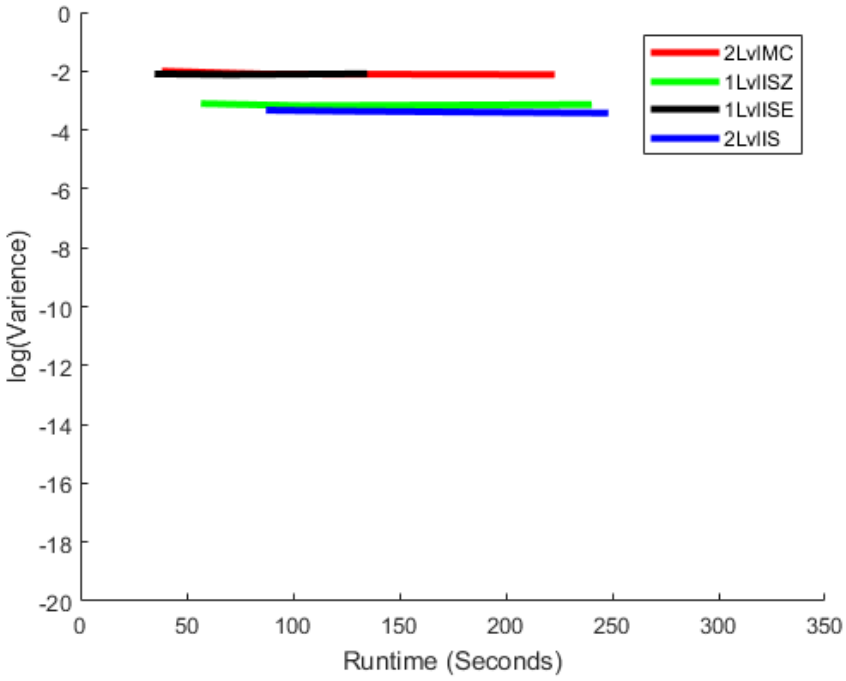

Figure 6.4: $S = 5$ $\ell = 0.2$

Method	Runtime(sec)	Variance	Variance Ratio	Standard Error
2LvlMC(300,300)	38	9.93e-3	100%	1.99e-4
2LvlMC(400,400)	85	7.98e-3	100%	1.78e-4
2LvlMC(500,500)	222	7.55e-3	100%	1.73e-4
1LvlISZ(300,300)	56	7.85e-4	7.90%	5.60e-5
1LvlISZ(400,400)	103	6.67e-4	8.36%	5.16e-5
1LvlISZ(500,500)	240	7.36e-4	9.74%	5.42e-5
1LvlISE(300,300)	34	8.02e-3	80.75%	1.79e-4
1LvlISE(400,400)	70	7.59e-3	95.09%	1.74e-4
1LvlISE(500,500)	134	8.11e-3	107.46%	1.80e-4
2LvlIS(300,300)	87	4.66e-4	4.69%	4.32e-5
2LvlIS(400,400)	126	4.34e-4	5.44%	4.16e-5
2LvlIS(500,500)	248	3.74e-4	4.96%	3.87e-5

Table 6.4: Variance reduction for $S = 5$, $\ell = 0.2$

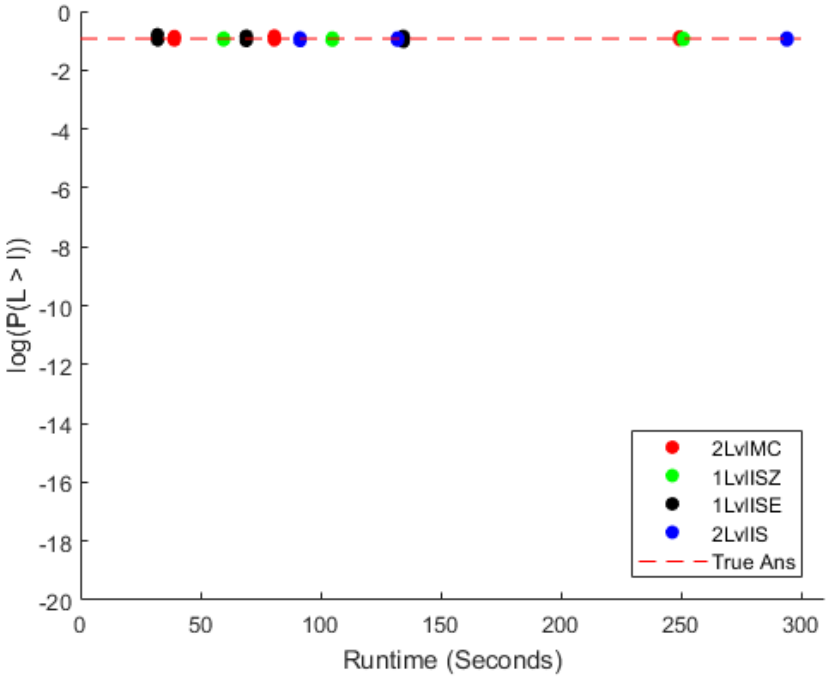

Figure 6.5: $S = 5$ $\ell = 0.1$

Method	Runtime(sec)	Absolute Error	Relative Error
2LvlMC(300,300)	39	1.23e-2	1.06e-1
2LvlMC(400,400)	80	1.18e-2	1.01e-1
2LvlMC(500,500)	249	9.33e-3	8.05e-2
1LvlISZ(300,300)	59	5.56e-3	4.80e-2
1LvlISZ(400,400)	104	5.88e-3	5.07e-2
1LvlISZ(500,500)	251	4.85e-3	4.18e-2
1LvlISE(300,300)	32	1.26e-2	1.09e-1
1LvlISE(400,400)	69	1.35e-2	1.16e-1
1LvlISE(500,500)	134	1.47e-2	1.26e-1
2LvlIS(300,300)	91	7.59e-3	6.54e-2
2LvlIS(400,400)	132	6.07e-3	5.23e-2
2LvlIS(500,500)	294	4.20e-3	3.62e-2

Table 6.5: Error and variance for $S = 5$, $\ell = 0.1$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 1.15e−1

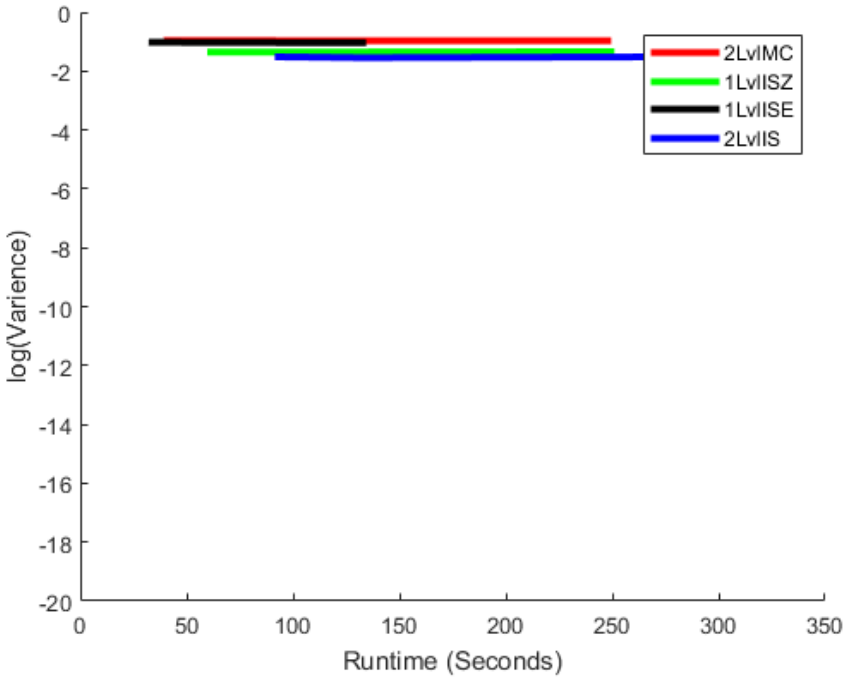

Figure 6.6: $S = 5$ $\ell = 0.1$

Method	Runtime(sec)	Variance	Variance Ratio	Standard Error
2LvlMC(300,300)	39	1.08e-1	100%	6.59e-4
2LvlMC(400,400)	80	1.08e-1	100%	6.58e-4
2LvlMC(500,500)	249	1.06e-1	100%	6.53e-4
1LvlISZ(300,300)	59	4.43e-2	40.75%	4.21e-4
1LvlISZ(400,400)	104	4.41e-2	40.73%	4.20e-4
1LvlISZ(500,500)	251	4.57e-2	42.84%	4.27e-4
1LvlISE(300,300)	32	9.55e-2	87.86%	6.18e-4
1LvlISE(400,400)	69	9.33e-2	86.14%	6.10e-4
1LvlISE(500,500)	134	9.25e-2	86.70%	6.08e-4
2LvlIS(300,300)	91	3.04e-2	27.98%	3.48e-4
2LvlIS(400,400)	132	2.85e-2	26.36%	3.38e-4
2LvlIS(500,500)	294	3.06e-2	28.75%	3.50e-4

Table 6.6: Variance reduction for $S = 5$, $\ell = 0.1$

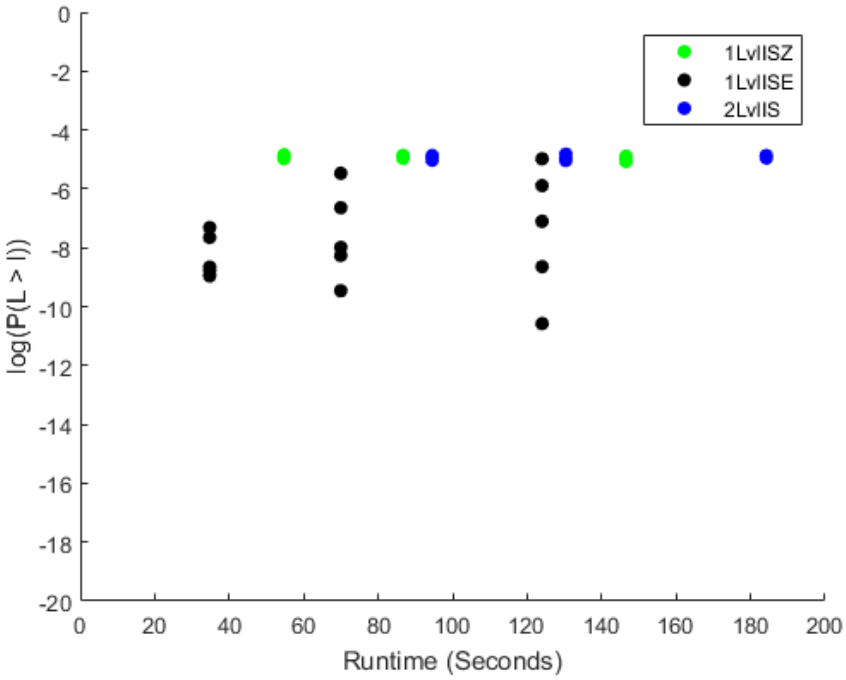

Figure 6.7: $S = 10$ $\ell = 0.3$. We did not include 2LvlMC or a line indicating the true answer for the reasons explained in the Analysis subsection on page 39.

Method	Runtime(sec)	Absolute Error	Relative Error
2LvlMC(300,300)	33	N/A	N/A
2LvlMC(400,400)	64	N/A	N/A
2LvlMC(500,500)	133	N/A	N/A
1LvlISZ(300,300)	54	N/A	N/A
1LvlISZ(400,400)	86	N/A	N/A
1LvlISZ(500,500)	146	N/A	N/A
1LvlISE(300,300)	34	N/A	N/A
1LvlISE(400,400)	69	N/A	N/A
1LvlISE(500,500)	124	N/A	N/A
2LvlIS(300,300)	94	N/A	N/A
2LvlIS(400,400)	130	N/A	N/A
2LvlIS(500,500)	184	N/A	N/A

Table 6.7: Error and variance for $S = 10$, $\ell = 0.3$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is not known in this case as our overnight run of 2LvlMC did not converge. See the discussion in the Analysis subsection on page 39.

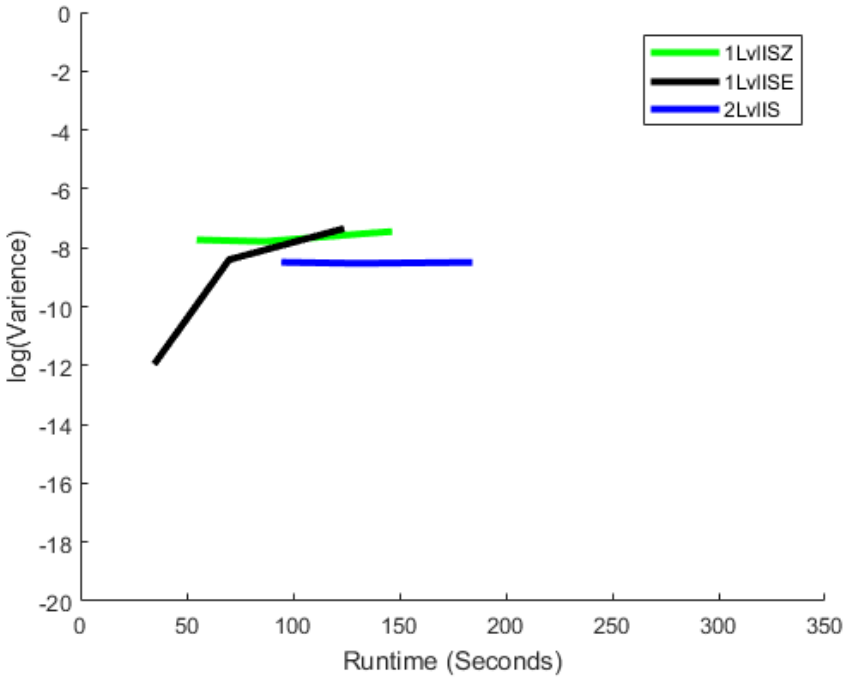

Figure 6.8: $S = 10$ $\ell = 0.3$

Method	Runtime(sec)	Variance	Variance Ratio	Standard Error
2LvlMC(300,300)	33	0.0	100%	0
2LvlMC(400,400)	64	0.0	100%	0
2LvlMC(500,500)	133	1.59e-6	100%	2.52e-6
1LvlISZ(300,300)	54	1.85e-8	N/A	2.72e-7
1LvlISZ(400,400)	86	1.62e-8	N/A	2.55e-7
1LvlISZ(500,500)	146	3.57e-8	N/A	3.78e-7
1LvlISE(300,300)	34	1.10e-12	N/A	2.09e-9
1LvlISE(400,400)	69	3.93e-9	N/A	1.25e-7
1LvlISE(500,500)	124	4.49e-8	N/A	4.24e-7
2LvlIS(300,300)	94	3.21e-9	N/A	1.13e-7
2LvlIS(400,400)	130	2.94e-9	N/A	1.08e-7
2LvlIS(500,500)	184	3.19e-9	N/A	1.13e-7

Table 6.8: Variance reduction for $S = 10$, $\ell = 0.3$

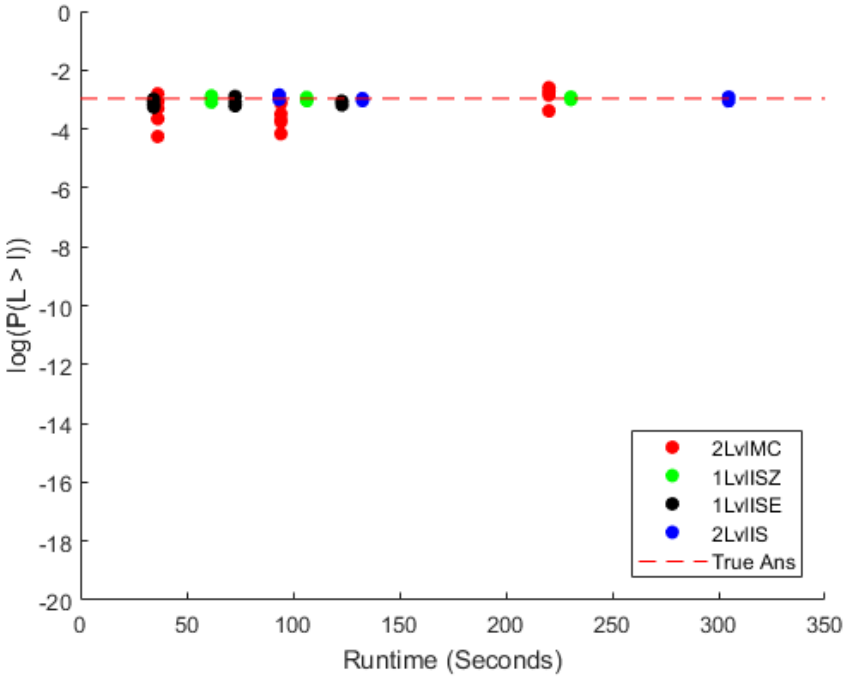

Figure 6.9: $S = 10$ $\ell = 0.2$

Method	Runtime(sec)	Absolute Error	Relative Error
2LvlMC(300,300)	36	6.32e-4	5.85e-1
2LvlMC(400,400)	94	7.77e-4	7.19-1
2LvlMC(500,500)	220	7.69e-4	7.12e-1
1LvlISZ(300,300)	61	1.29e-4	1.20e-1
1LvlISZ(400,400)	106	7.69e-5	7.12e-2
1LvlISZ(500,500)	230	1.02e-4	9.48e-2
1LvlISE(300,300)	34	6.43-3	7.87e-2
1LvlISE(400,400)	72	4.93e-3	6.03e-2
1LvlISE(500,500)	123	3.48e-3	4.25e-2
2LvlIS(300,300)	93	1.34e-4	1.24e-1
2LvlIS(400,400)	132	7.97e-5	7.38e-2
2LvlIS(500,500)	304	7.54e-5	6.98e-2

Table 6.9: Error and variance for $S = 10$, $\ell = 0.2$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 1.20e−3

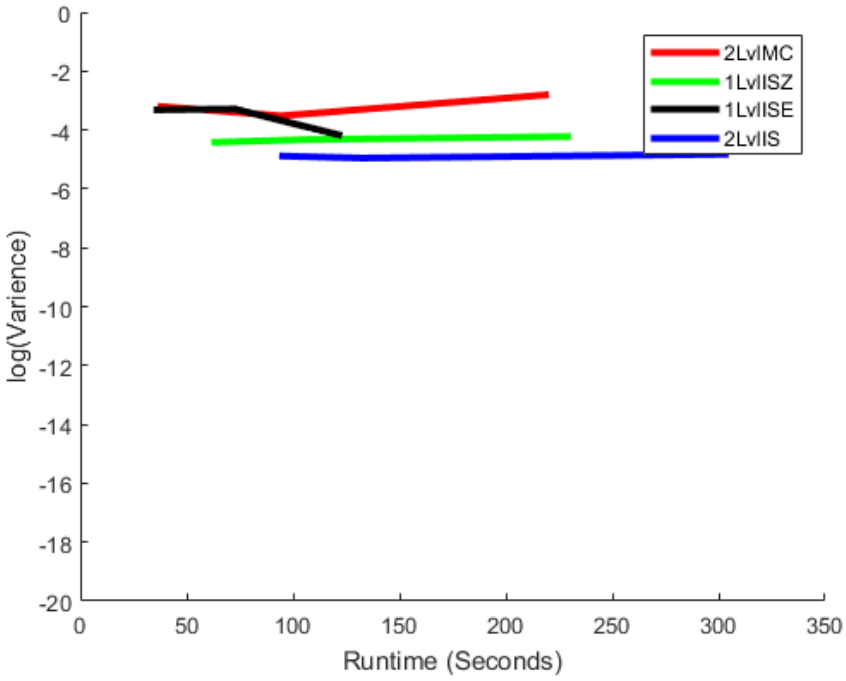

Figure 6.10: $S = 10$ $\ell = 0.2$

Method	Runtime(sec)	Variance	Variance Ratio	Standard Error
2LvlMC(300,300)	36	6.41e-4	100%	5.06e-5
2LvlMC(400,400)	94	3.02e-4	100%	3.47e-5
2LvlMC(500,500)	220	1.57e-3	100%	7.94e-5
1LvlISZ(300,300)	61	3.79e-5	5.91%	1.23e-5
1LvlISZ(400,400)	106	4.72e-5	15.62%	1.37e-5
1LvlISZ(500,500)	230	5.98e-5	3.79%	1.54e-5
1LvlISE(300,300)	34	4.83e-4	75.32%	4.39e-5
1LvlISE(400,400)	72	5.14e-4	170.05%	4.53e-5
1LvlISE(500,500)	123	6.41e-5	4.06%	1.60e-5
2LvlIS(300,300)	93	1.30e-5	2.03%	7.22e-6
2LvlIS(400,400)	132	1.10e-5	3.66%	6.65e-6
2LvlIS(500,500)	333	1.50e-5	0.95%	7.74e-6

Table 6.10: Variance reduction for $S = 10$, $\ell = 0.2$

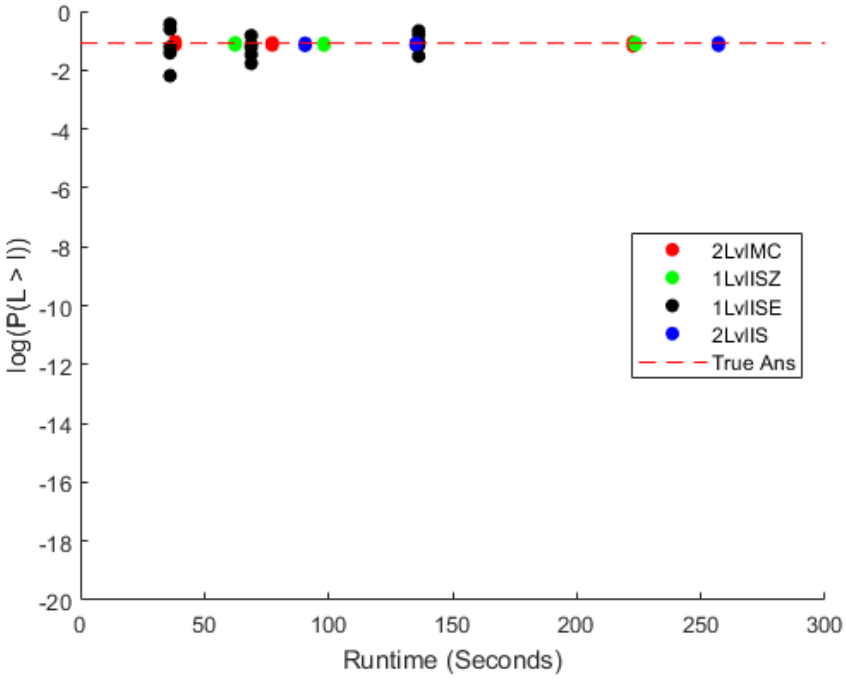

Figure 6.11: $S = 10$ $\ell = 0.1$

Method	Runtime(sec)	Absolute Error	Relative Error
2LvlMC(300,300)	38	7.43e-3	9.09e-2
2LvlMC(400,400)	77	3.62e-3	4.43e-2
2LvlMC(500,500)	222	5.21e-3	6.38e-2
1LvlISZ(300,300)	62	6.43e-3	7.87e-2
1LvlISZ(400,400)	98	4.93e-3	6.03e-2
1LvlISZ(500,500)	223	3.48e-3	4.25e-2
1LvlISE(300,300)	36	1.17e-2	1.44e-1
1LvlISE(400,400)	68	5.46e-3	6.68e-2
1LvlISE(500,500)	136	9.37e-3	1.14e-1
2LvlIS(300,300)	90	4.85e-3	5.93e-2
2LvlIS(400,400)	135	5.67e-3	6.93e-2
2LvlIS(500,500)	257	4.42e-3	5.41e-2

Table 6.11: Error and variance for $S = 10$, $\ell = 0.1$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 8.17e−2

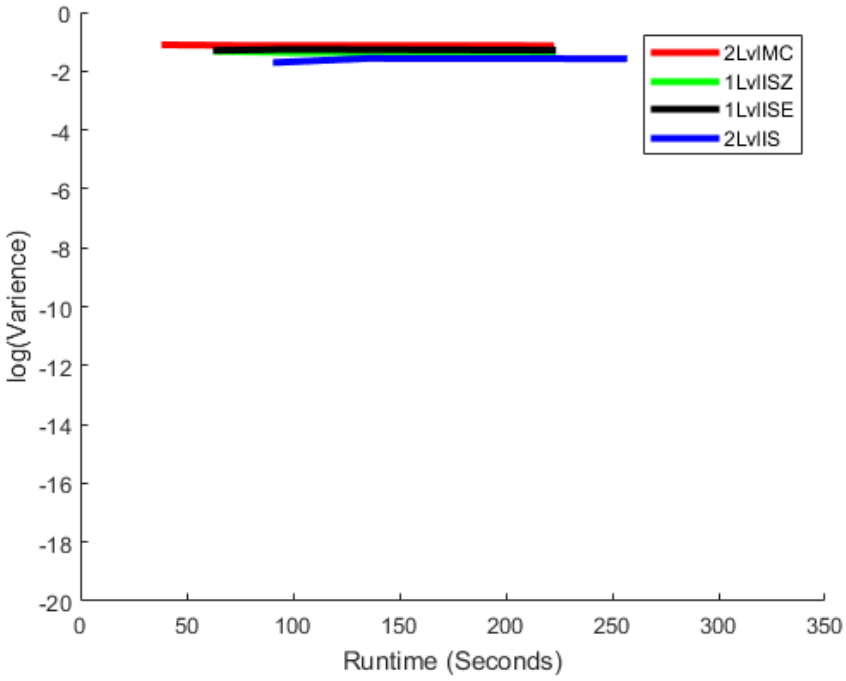

Figure 6.12: $S = 10$ $\ell = 0.1$

Method	Runtime(sec)	Variance	Variance Ratio	Standard Error
2LvlMC(300,300)	38	7.81e-2	100%	5.59e-4
2LvlMC(400,400)	77	7.42e-2	100%	5.44e-4
2LvlMC(500,500)	222	7.34e-2	100%	5.42e-4
1LvlISZ(300,300)	62	4.71e-2	60.29%	4.34e-4
1LvlISZ(400,400)	98	4.27e-2	57.62%	4.13e-4
1LvlISZ(500,500)	223	4.42e-2	60.25%	4.20e-4
1LvlISE(300,300)	36	5.03e-2	64.37%	4.48e-4
1LvlISE(400,400)	68	5.52e-2	74.41%	4.70e-4
1LvlISE(500,500)	136	5.13e-2	69.86%	4.53e-4
2LvlIS(300,300)	90	1.93e-2	24.80%	2.78e-4
2LvlIS(400,400)	135	2.69e-2	36.25%	3.28e-4
2LvlIS(500,500)	257	2.62e-2	35.74%	3.24e-4

Table 6.12: Variance reduction for $S = 10$, $\ell = 0.1$

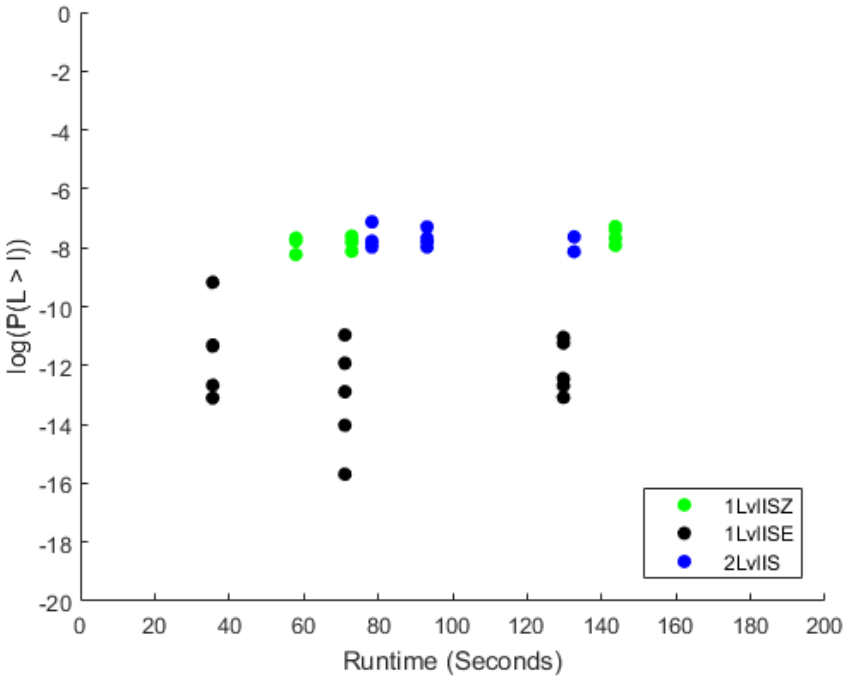

Figure 6.13: $S = 20$ $\ell = 0.3$. We did not include 2LvlMC or a line indicating the true answer for the reasons explained in the Analysis subsection on page 39.

Method	Runtime(sec)	Absolute Error	Relative Error
2LvlMC(300,300)	29	N/A	N/A
2LvlMC(400,400)	65	N/A	N/A
2LvlMC(500,500)	131	N/A	N/A
1LvlISZ(300,300)	57	N/A	N/A
1LvlISZ(400,400)	72	N/A	N/A
1LvlISZ(500,500)	143	N/A	N/A
1LvlISE(300,300)	35	N/A	N/A
1LvlISE(400,400)	71	N/A	N/A
1LvlISE(500,500)	129	N/A	N/A
2LvlIS(300,300)	93	N/A	N/A
2LvlIS(400,400)	78	N/A	N/A
2LvlIS(500,500)	132	N/A	N/A

Table 6.13: Error and variance for $S = 20$, $\ell = 0.3$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is not known in this case as our overnight run of 2LvlMC did not converge. See the discussion in the Analysis subsection on page 39.

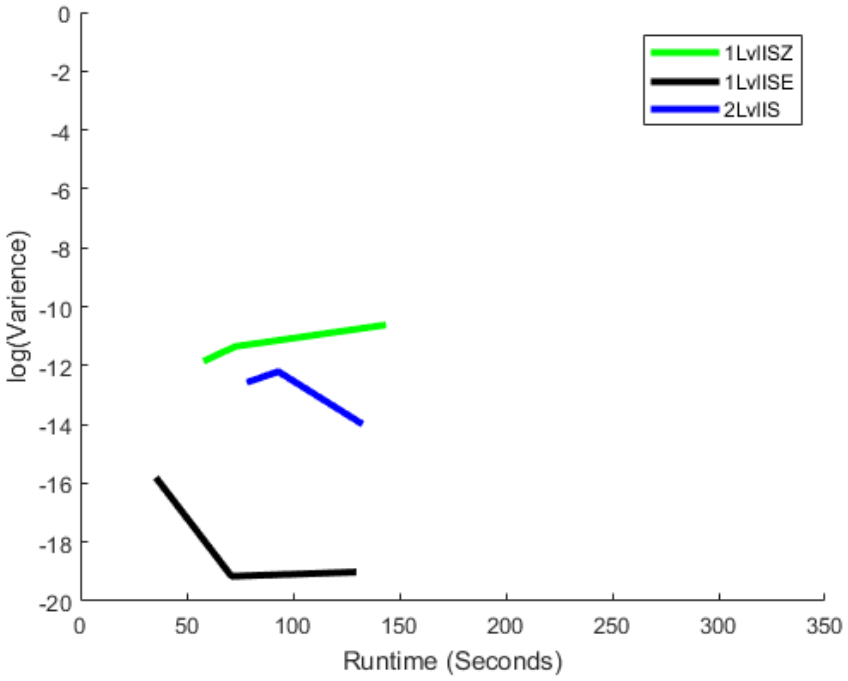

Figure 6.14: $S = 20$ $\ell = 0.3$

Method	Runtime(sec)	Variance	Variance Ratio	Standard Error
2LvlMC(300,300)	29	0.0	100%	0
2LvlMC(400,400)	65	0.0	100%	0
2LvlMC(500,500)	131	0.0	100%	0
1LvlISZ(300,300)	57	1.39e-12	N/A	2.36e-9
1LvlISZ(400,400)	72	4.42e-12	N/A	4.20e-9
1LvlISZ(500,500)	143	2.38e-11	N/A	9.77e-9
1LvlISE(300,300)	35	1.55e-16	N/A	2.49e-11
1LvlISE(400,400)	71	6.91e-20	N/A	5.26e-13
1LvlISE(500,500)	129	9.62e-20	N/A	6.20e-13
2LvlIS(300,300)	93	2.67e-13	N/A	1.03e-9
2LvlIS(400,400)	78	6.17e-13	N/A	1.57e-9
2LvlIS(500,500)	132	1.01e-14	N/A	2.01e-10

Table 6.14: Variance reduction for $S = 20$, $\ell = 0.3$

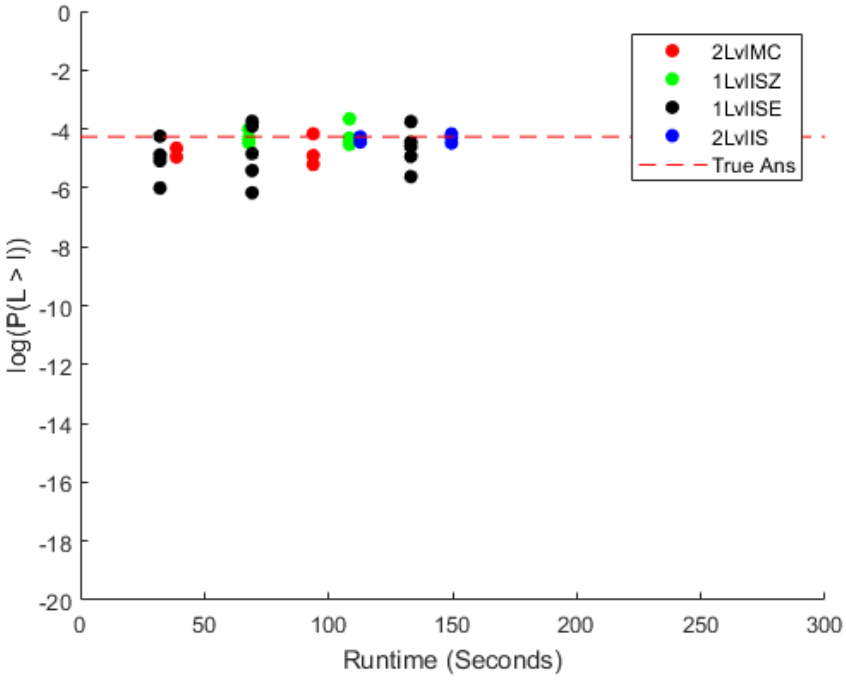

Figure 6.15: $S = 20$ $\ell = 0.2$

Method	Runtime(sec)	Absolute Error	Relative Error
2LvlMC(300,300)	38	4.52e-5	8.35e-1
2LvlMC(400,400)	93	3.99e-5	7.38e-1
2LvlMC(500,500)	234	1.11e-4	2.06
1LvlISZ(300,300)	67	2.61e-5	4.82e-1
1LvlISZ(400,400)	108	5.37e-5	9.93e-1
1LvlISZ(500,500)	133	2.42e-5	4.47e-1
1LvlISE(300,300)	32	3.72e-5	6.17
1LvlISE(400,400)	69	6.82e-5	21.57
1LvlISE(500,500)	245	5.29e-5	4.77
2LvlIS(300,300)	112	9.39e-6	1.73e-1
2LvlIS(400,400)	149	1.93e-5	3.57e-1
2LvlIS(500,500)	277	1.32e-5	2.45e-1

Table 6.15: Error and variance for $S = 20$, $\ell = 0.2$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 5.41e−5

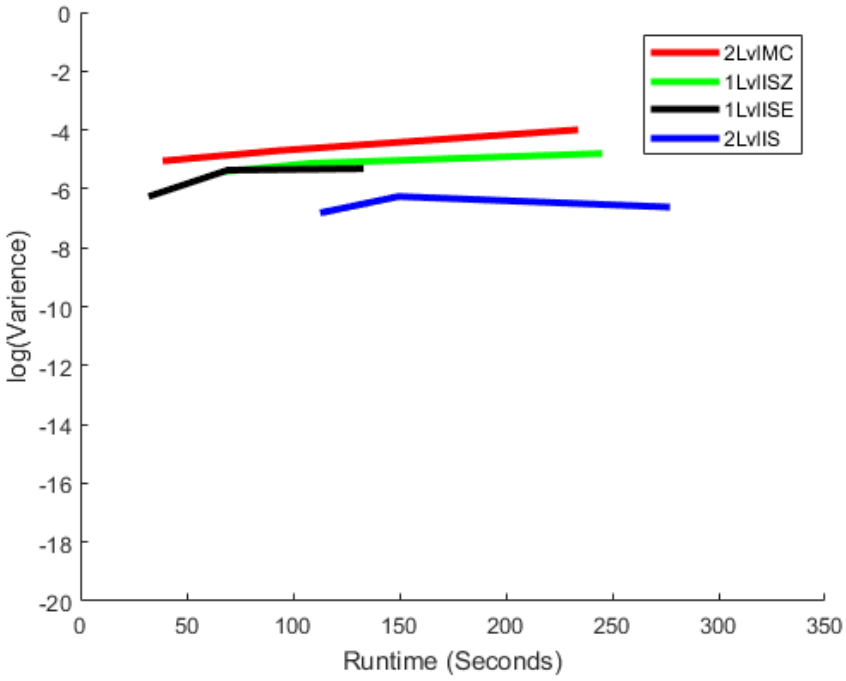

Figure 6.16: $S = 20$ $\ell = 0.2$

Method	Runtime(sec)	Variance	Variance Ratio	Standard Error
2LvlMC(300,300)	38	1.01e-4	100%	5.96e-6
2LvlMC(400,400)	93	1.99e-5	100%	8.94e-6
2LvlMC(500,500)	234	8.88e-6	100%	2.01e-5
1LvlISZ(300,300)	67	3.89e-6	43.77%	3.94e-6
1LvlISZ(400,400)	108	7.38e-6	36.94%	5.43e-6
1LvlISZ(500,500)	245	1.59e-5	15.67%	7.97e-6
1LvlISE(300,300)	67	5.48e-7	6.17%	1.48e-6
1LvlISE(400,400)	108	4.31e-6	21.57%	4.15e-6
1LvlISE(500,500)	245	4.84e-6	4.77%	4.40e-6
2LvlIS(300,300)	112	1.52e-7	1.71%	7.80e-7
2LvlIS(400,400)	149	5.50e-7	2.75%	1.48e-6
2LvlIS(500,500)	277	2.39e-7	0.23%	9.78e-7

Table 6.16: Variance reduction for $S = 20$, $\ell = 0.2$

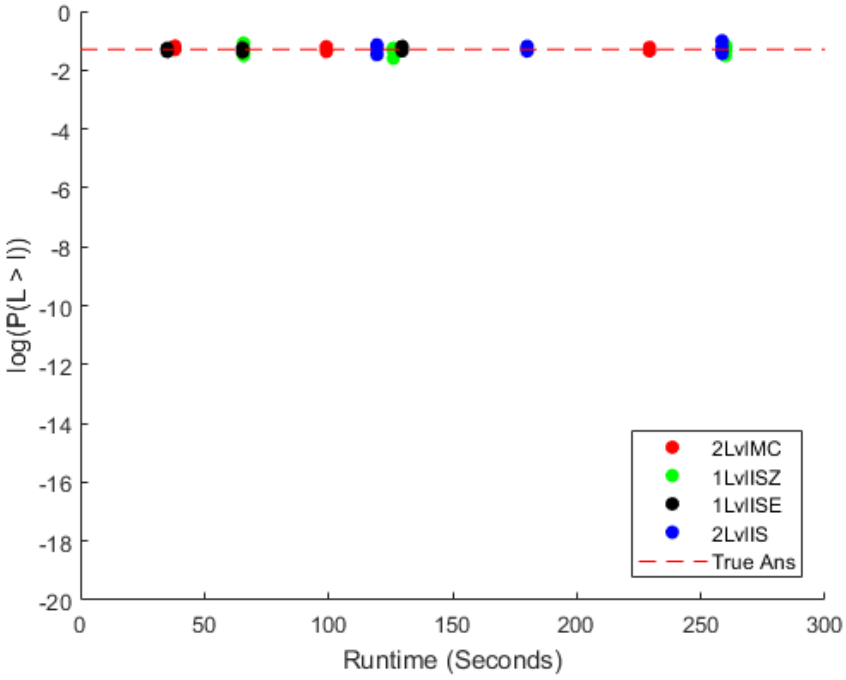

Figure 6.17: $S = 20$ $\ell = 0.1$

Method	Runtime(sec)	Absolute Error	Relative Error
2LvlMC(300,300)	38	5.17e-3	1.02e-1
2LvlMC(400,400)	99	5.32e-3	1.05e-1
2LvlMC(500,500)	229	3.92e-3	7.77e-2
1LvlISZ(300,300)	65	1.35e-2	2.67e-1
1LvlISZ(400,400)	126	8.66e-3	1.71e-1
1LvlISZ(500,500)	259	1.01e-2	2.00e-1
1LvlISE(300,300)	35	4.01e-3	7.94e-2
1LvlISE(400,400)	65	4.39e-3	8.69e-2
1LvlISE(500,500)	129	6.62e-3	1.31e-1
2LvlIS(300,300)	119	1.23e-2	2.43e-1
2LvlIS(400,400)	179	5.16e-3	1.02e-1
2LvlIS(500,500)	258	1.72e-2	3.42e-1

Table 6.17: Error and variance for $S = 20$, $\ell = 0.1$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 5.05e−2

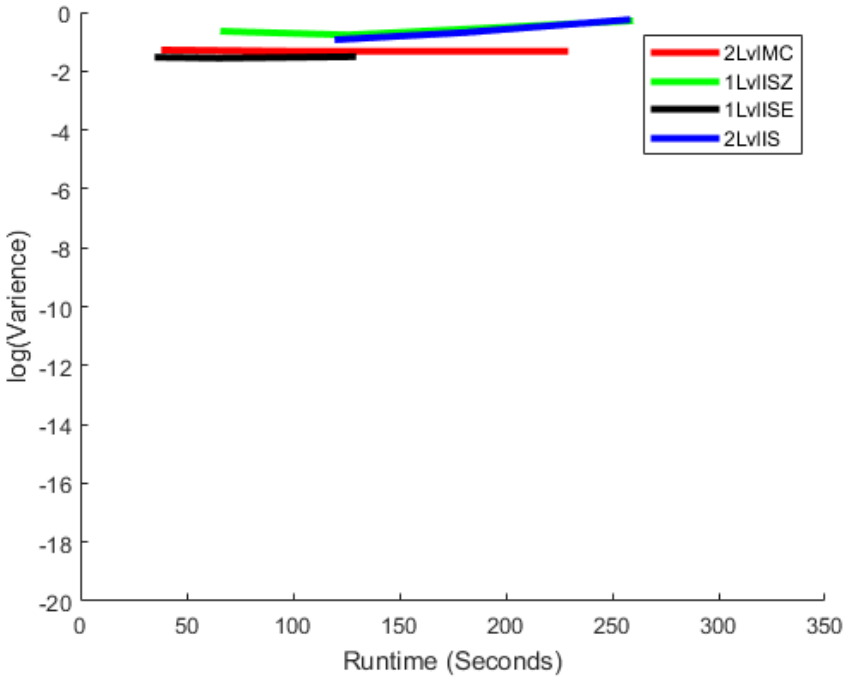

Figure 6.18: $S = 20$ $\ell = 0.1$

Method	Runtime(sec)	Variance	Variance Ratio	Standard Error
2LvlMC(300,300)	38	5.21e-2	100%	4.56e-4
2LvlMC(400,400)	99	4.73e-2	100%	4.35e-4
2LvlMC(500,500)	229	4.74e-2	100%	4.35e-4
1LvlISZ(300,300)	65	2.25e-1	432.06%	9.49e-4
1LvlISZ(400,400)	126	1.69e-1	357.11%	8.22e-4
1LvlISZ(500,500)	391	5.13e-1	1082.35%	1.43e-3
1LvlISE(300,300)	35	2.95e-2	56.75%	3.44e-4
1LvlISE(400,400)	65	2.77e-2	58.48%	3.32e-4
1LvlISE(500,500)	129	3.09e-2	65.21%	3.51e-4
2LvlIS(300,300)	119	1.15e-1	222.12%	6.80e-4
2LvlIS(400,400)	179	2.02e-1	427.06%	8.99e-4
2LvlIS(500,500)	258	5.61e-1	1183.24%	1.49e-3

Table 6.18: Variance reduction for $S = 20$, $\ell = 0.1$

Analysis

Analysis of the data from this run of experiments shows a number of interesting trends.

First is that as ℓ decreases the variance reduction of our new methods also decreases. This is not unexpected. Importance sampling is primarily used in rare event simulation, so it is not unreasonable to expect it to perform better as the events you are considering get more rare.

Second is that as S increases our new methods experience a similar decrease in performance as when ℓ increases. Our intuition for this is that higher dimensional spaces simply permit greater variance in the simulated data.

Third is that our methods actually under-perform naive 2 level MC integration in the case where $S = 20$ and $\ell = 0.1$. Our intuition here is that as S increases one must be careful to ensure that we adequately train $\pi^*(z) : \mathbb{R}^S \to \mathbb{R}$. If an inadequate number of samples is used to train $\pi^*(z)$ then it will only learn a small subspace of $\mathbb{R}^S$ represented by those samples. As a result, it may fail to generalize to an appropriate approximation of $\pi(z)$ on the whole space. This intuition is explored and verified in subsection 6.1.5.

Fourth is that the $\mathcal{E}$ importance sampler that Glasserman and Li developed [5], and which we generalized to the multi-credit-state model, is not as effective as the $\mathcal{Z}$ importance sampler. We believe that there are two reasons for this. The first is that we set $N = 2500$. Hence we are in a regime where the normal approximation from theorem 1 is actually quite good. Thus, if π^* is trained well, it does closely approximate the zero variance importance sampler for the true probability, not just the approximation. Second is that the importance sampler for $\mathcal{E}$ does not minimize the variance directly but instead minimizes a loose upper bound on the second moment. Chapter 8 briefly explores a possible avenue for obtaining a better minimization problem in order to fix the second issue. However it is worth noting that the usefulness of the $\mathcal{E}$ importance sampler increases as S increases. Looking at $\ell = 0.2$ we see that it barely helps in the $S = 5$ regime. However by the time we get up to $S = 20$ it reduces the variance by a factor of 10. Figure 6.19 highlights this trend. It is possible that this is due to a poorly trained π^* leaving behind more variance to be reduced by the exponential twisting. This hypothesis is also explored in the next section.

Fifth is that in the case of extremely rare events ($\ell = 0.3$) 2LvlMC completely failed to produce a non-zero answer for $S = 10$ and $S = 20$. In this regime one is unlikely to sample any $\mathcal{Z}$ or $\mathcal{E}$ that result in the indicator functions being 1 without making use of a technique like importance sampling. In such cases the variance reduction and error are reported as N/A as we do not have a valid baseline to compare against.

6.1.5 π^* Training

In this subsection we validate the hypothesis that as S increases a larger number of samples are needed to effectively train π^*. Here the dashed lines represent the average variance of 2LvlMC when ran with the same number of $\mathcal{Z}$ and $\mathcal{E}$ samples as the methods being studied.

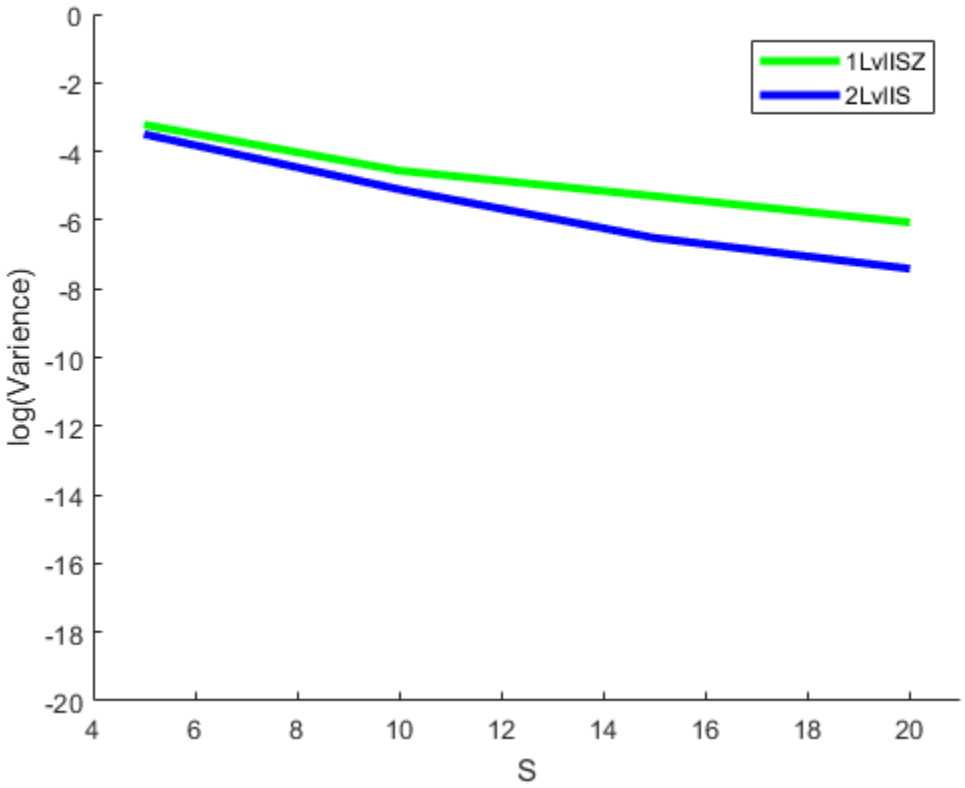

Figure 6.19: Variance as a function of S

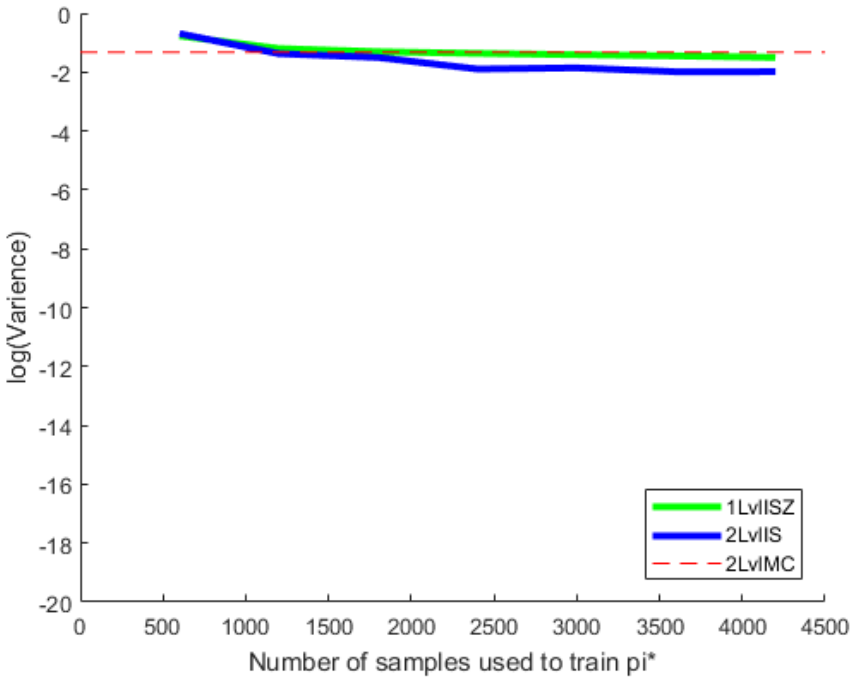

Figure 6.20: $S = 20$ $\ell = 0.1$

Method	Sample No.	Variance	Variance Ratio	Standard Error
1LvlISZ(400,400)	600	1.69e-1	357.11%	8.22e-4
1LvlISZ(400,400)	1200	6.29e-2	132.94%	5.01e-4
1LvlISZ(400,400)	1800	4.92e-2	104.00%	4.43e-4
1LvlISZ(400,400)	2400	4.32e-2	91.23%	4.15e-4
1LvlISZ(400,400)	3000	3.89e-2	82.18%	3.94e-4
1LvlISZ(400,400)	3600	3.54e-2	74.91%	3.76e-4
1LvlISZ(400,400)	4200	3.13e-2	66.23%	3.54e-4
2LvlIS(400,400)	600	2.02e-1	427.06%	8.99e-4
2LvlIS(400,400)	1200	4.22e-2	89.18%	4.11e-4
2LvlIS(400,400)	1800	3.13e-2	66.23%	3.54e-4
2LvlIS(400,400)	2400	1.26e-2	26.79%	2.25e-4
2LvlIS(400,400)	3000	1.38e-2	29.28%	2.35e-4
2LvlIS(400,400)	3600	1.02e-2	21.72%	2.028e-4
2LvlIS(400,400)	4200	1.03e-2	21.80%	2.03e-4

Table 6.19: Variance reduction for $S = 20$, $\ell = 0.1$ as a function of number of samples used to train π^*

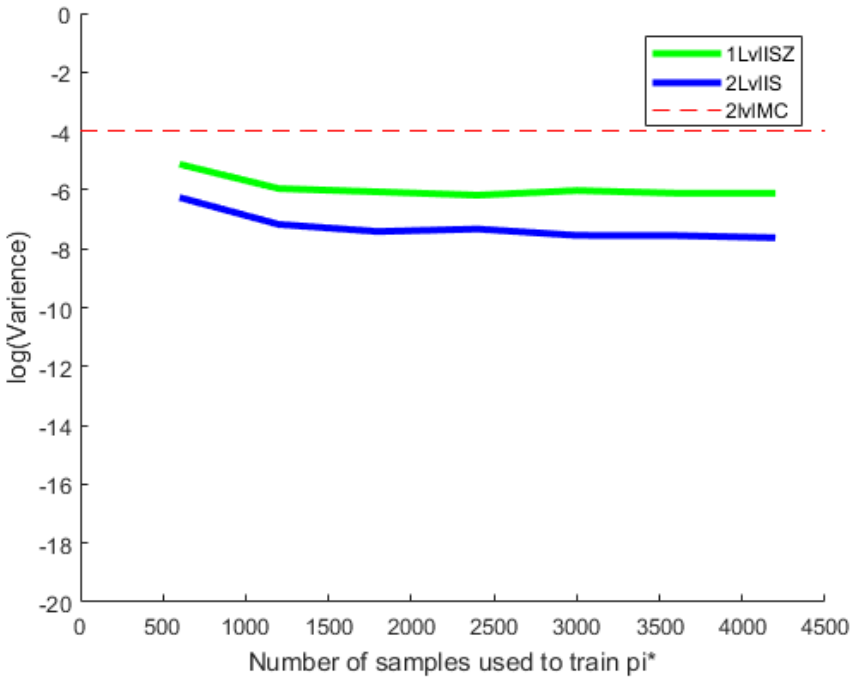

Figure 6.21: $S = 20$ $\ell = 0.2$

Method	Sample No.	Variance	Variance Ratio	Standard Error
1LvlISZ(400,400)	600	7.38e-6	7.27%	5.43e-6
1LvlISZ(400,400)	1200	1.10e-6	1.08%	2.10e-6
1LvlISZ(400,400)	1800	8.57e-7	0.84%	1.85e-6
1LvlISZ(400,400)	2400	6.57e-7	0.64%	1.62e-6
1LvlISZ(400,400)	3000	9.48e-7	0.93%	1.94e-6
1LvlISZ(400,400)	3600	7.68e-7	0.75%	1.75e-6
1LvlISZ(400,400)	4200	7.73e-7	0.76%	1.75e-6
2LvlIS(400,400)	600	5.50e-7	0.54%	1.48e-6
2LvlIS(400,400)	1200	6.68e-8	0.065%	5.17e-7
2LvlIS(400,400)	1800	3.83e-8	0.037%	3.91e-7
2LvlIS(400,400)	2400	4.67e-8	0.045%	4.32e-7
2LvlIS(400,400)	3000	2.83e-8	0.027%	3.37e-7
2LvlIS(400,400)	3600	2.79e-8	0.027%	3.34e-7
2LvlIS(400,400)	4200	2.36e-8	0.023%	3.07e-7

Table 6.20: Variance reduction for $S = 20$, $\ell = 0.2$ as a function of number of samples used to train π^*

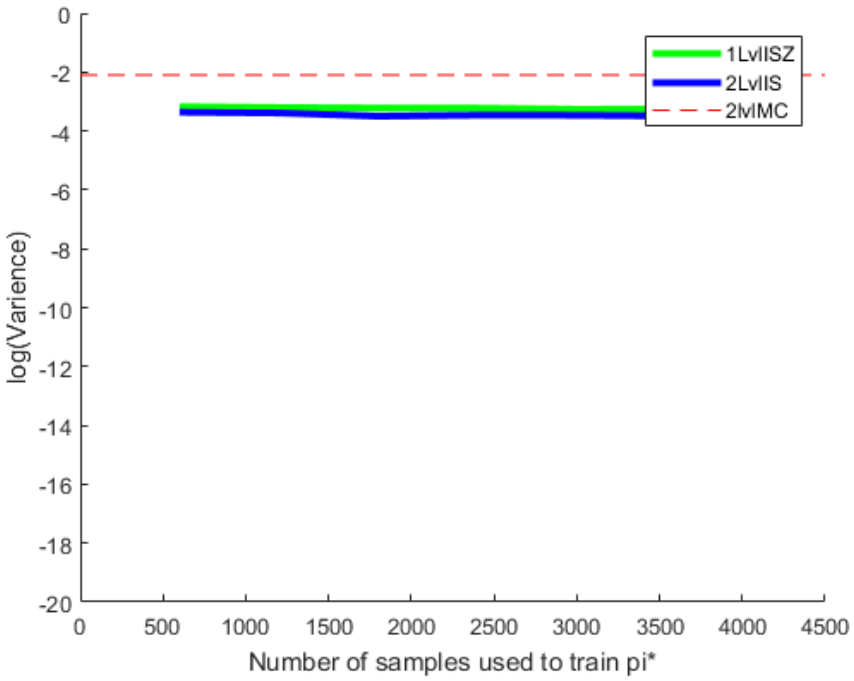

Figure 6.22: $S = 5$ $\ell = 0.2$

Method	Sample No.	Variance	Variance Ratio	Standard Error
1LvlISZ(400,400)	600	6.67e-4	8.36%	5.16e-5
1LvlISZ(400,400)	1200	6.25e-4	7.82%	5.00e-5
1LvlISZ(400,400)	1800	6.01e-4	7.53%	4.90e-5
1LvlISZ(400,400)	2400	5.95e-4	7.46%	4.88e-5
1LvlISZ(400,400)	3000	5.45e-4	6.83%	4.67e-5
1LvlISZ(400,400)	3600	5.52e-4	6.92%	4.70e-5
1LvlISZ(400,400)	4200	6.19e-4	7.75%	4.97e-5
2LvlIS(400,400)	600	4.34e-4	5.44%	4.16e-5
2LvlIS(400,400)	1200	4.10e-4	5.14%	4.05e-5
2LvlIS(400,400)	1800	3.16e-4	3.96%	3.56e-5
2LvlIS(400,400)	2400	3.44e-4	4.31%	3.71e-5
2LvlIS(400,400)	3000	3.41e-4	4.28%	3.69e-5
2LvlIS(400,400)	3600	3.23e-4	4.04%	3.59e-5
2LvlIS(400,400)	4200	3.38e-4	4.23%	3.67e-5

Table 6.21: Variance reduction for $S = 5$, $\ell = 0.2$ as a function of number of samples used to train π^*

Analysis

Figure 6.20 and Table 6.19 support our claim that better training π^* removes the anomaly where 2LvlMC outperforms 1LvlISZ and 2LvlIS.

The fact that 2LvlIS continues to show a noticeably better variance reduction over 1LvlISZ invalidates the notion that the increased performance of the $\mathcal{E}$ importance sampler came from the reduced effectiveness of the $\mathcal{Z}$ importance sampler due to a poorly trained π^*.

In order to see if a better trained π^* might further reduce the variance in our lower dimension experiments we created Figure 6.22 and Table 6.21. These showed only minor improvements and we generally consider them not worth the extra run time they incur.

6.1.6 Central Limit Theorem Bias

When we refer to the bias introduced by the normal approximation of Theorem 1 we mean the following: If you were to simply use Algorithm 4.2 the MC integration would converge quite rapidly. However since this CLT based result is exact only in the limit of $N = \infty$ the answer that it converges to is not the true answer, but one which is biased towards the normal approximation. The following experiments are meant to elucidate this matter. Here the dashed lines represent an approximation to the true answer for each algorithm obtained by running them overnight. The red line was obtained using 2LvlMC and the purple line was obtained using Algorithm 4.1

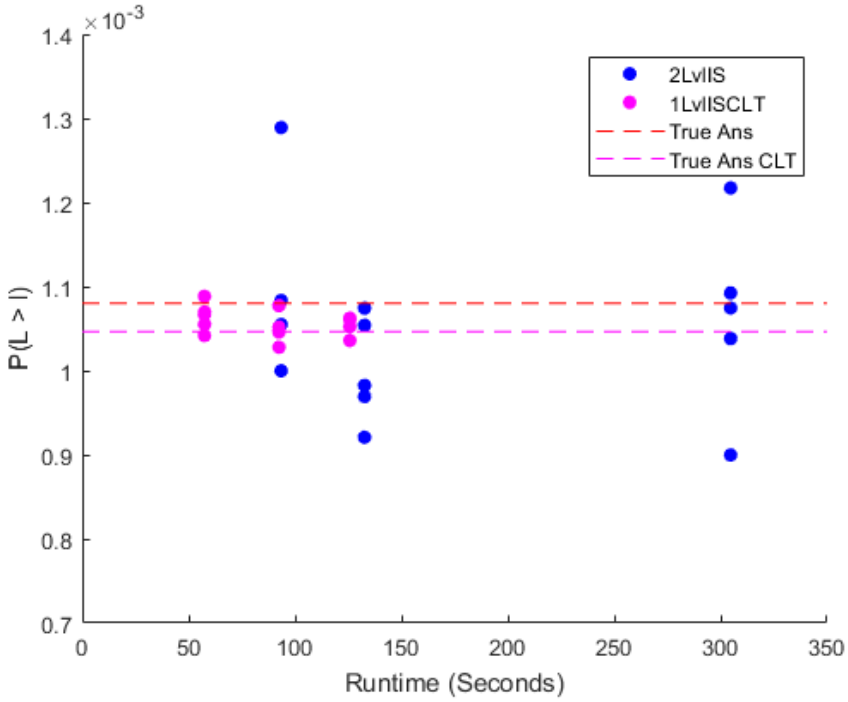

Figure 6.23: $S = 10$ $\ell = 0.2$

Method	Runtime(sec)	Absolute Error	Relative Error	Variance
1LvlISZCLT(20000)	57	1.89e-5	1.75e-2	7.90e-6
1LvlISZCLT(40000)	92	3.01e-5	2.78e-2	6.42e-6
1LvlISZCLT(60000)	125	2.71e-5	2.51e-2	5.20e-6
2LvlIS(300,300)	93	1.34e-4	1.24e-1	1.30e-5
2LvlIS(400,400)	132	7.97e-5	7.38e-2	1.10e-5
2LvlIS(500,500)	304	7.54e-5	6.98e-2	1.50e-5

Table 6.22: Error and variance for $S = 10$, $\ell = 0.2$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 1.08e−3. The CLT approximation is approximately 1.04e−3

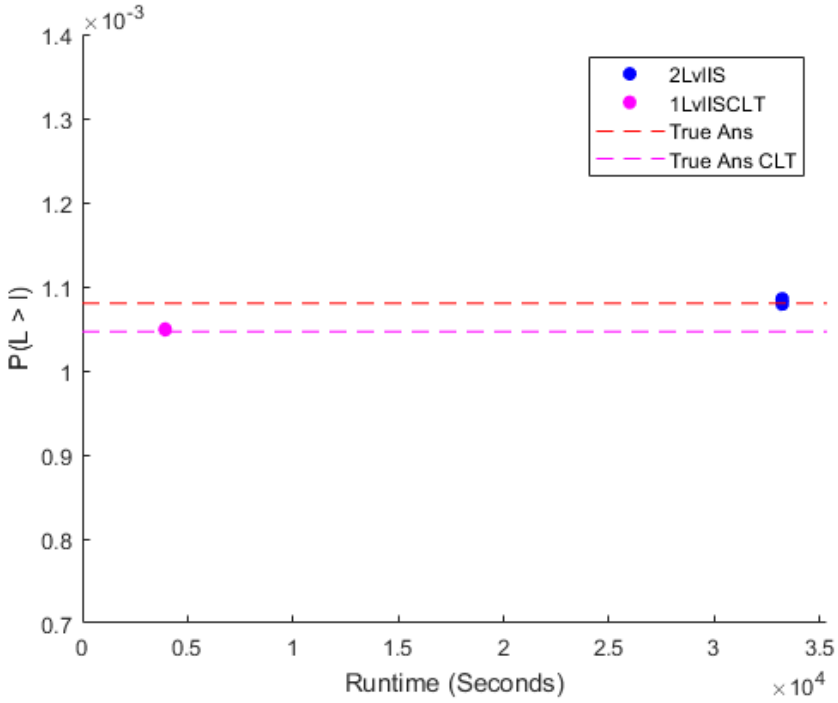

Figure 6.24: $S = 10$ $\ell = 0.2$

Method	Runtime(sec)	Absolute Error	Relative Error	Variance
1LvlISZCLT(1400000)	3935	3.08e-5	2.85e-2	7.90e-6
2LvlIS(10000,10000)	33222	3.08e-6	2.85e-3	1.30e-5

Table 6.23: Error and variance for $S = 10$, $\ell = 0.2$. True value of $P(\mathcal{L}_N(\mathcal{Z}, \mathcal{E}) \geq \ell)$ is approximately 1.08e−3. The CLT approximation is approximately 1.04e−3

Analysis

First note that the axis scales here have been changed compared to those in the previous sections to highlight the differences between the normal approximation and the true answer.

Making direct use of the CLT approximation removes $\mathcal{E}$ from the problem entirely. As such, this one level algorithm is much faster than its two level counterparts. Thus more samples can be taken per time interval, and convergence is therefore faster. However as mentioned in the introduction to this section the convergence is to a biased answer. Analyzing Figures 6.23 and 6.24 it becomes clear that in the short runtime regime the CLT approximation provides a more reliable estimate of the true answer, but if the runtime is allowed to be lengthened then our two level importance sampling scheme, 2LvlIS, outperforms it.

6.2 Full Credit State Experiments

One of the key achievements in this paper is extending Glasserman and Li's [5] inner importance sampler to from a binary-credit-state model to a multi-credit-state model. In order to test this method we used

the following model parameters.

$$
\begin{aligned}
p_n^c &= \frac{0.01}{3}\left(1+\sin\left(\frac{16\pi n}{N}\right)\right) && n=1,\cdots,N \qquad c=1,3,4\\
p_n^2 &= 1-(p_n^1+p_n^3+p_n^4) && n=1,\cdots,N\\
\beta_{nj} &\sim \text{Unif}\left(-\frac{1}{\sqrt{S}},\frac{1}{\sqrt{S}}\right) && n=1,\cdots,N \qquad j=1,\cdots,S\\
LGC_n^c &= \left\lfloor\frac{5k}{N}\right\rfloor^2 && n=1,\cdots,N \qquad c=1,3,4\\
EAD_n &\sim \text{Unif}(0.5,1.5) && n=1,\cdots,N
\end{aligned}
$$

By using these settings for p_n^c and LGC_n^c and copying the randomly generated parameters from the binary case we can recreate the binary-credit-state experiments in the multi-credit-state case. Having performed these calculations we have found the answers agree within reason. That being the case we have elected not to include these results here, but the data can be found online here for those who are interested.

Chapter 7

Conclusion

In this research paper we removed previously existing limitations from the works of Wang [10] and Glasserman and Li [5]. That is, we removed the bias from the $\mathcal{Z}$ importance sampler introduced by Wang, and we extended the $\mathcal{E}$ exponential twisting approach of Glasserman and Li to the multi-credit state model. We ran numerical experiments to probe the effectiveness of these improvements and discovered several interesting facts. First, for low dimensions ($S < 10$), the $\mathcal{E}$ importance sampling was not very useful. However in higher dimensions it becomes more important. In general, though, it is not as powerful as the $\mathcal{Z}$ importance sampler. A potential means of improving this technique is discussed in chapter 8. The importance of properly training π^* was also shown, as a poorly trained π^* may actually increase the variance. Also a fast but biased CLT based method was compared against our unbiased method showing the trade off between run time and accuracy in the large N regime.

Chapter 8

Future Work

8.1 Fix Glasserman and Li

The most obvious future work is to remove the downward bias that we have been experiencing from our implementation of the Glasserman and Li importance sampler. Then we can directly compare our methods against Glasserman and Li's as opposed to just the naive two level MC method.

8.2 Improve $\mathcal{E}$ Importance Sampling

Our $\mathcal{E}$ importance sampler works by minimizing an upper bound on the second moment.

$$\mathbb{E}_q[\mathbb{1}_{\{\mathcal{L}>\ell\}}e^{-2\theta\mathcal{L}+2\psi(z,\theta)}] \leq e^{-2\theta\ell+2\psi(z,\theta)}$$

for $\theta \geq 0$. This upper bound does not apply for all θ and where it does it is not particularly tight. One idea we consider to be a potentially fruitful avenue of approach is to apply the normal approximation that has served us so well for the outer layer to the inner layer as well. Conditional on $\mathcal{Z} = z$ we have

$$\begin{aligned}
\mathbb{V}_q[\mathbb{1}_{\{\mathcal{L}>\ell\}}e^{\theta\mathcal{L}+\psi(z,\theta)}] &= \mathbb{E}_q[\mathbb{1}_{\{\mathcal{L}>\ell\}}e^{-2\theta\mathcal{L}+2\psi(z,\theta)}] - \mathbb{E}_q[\mathbb{1}_{\{\mathcal{L}>\ell\}}e^{-\theta\mathcal{L}+\psi(z,\theta)}]^2 \\
&= \mathbb{E}_q[\mathbb{1}_{\{\mathcal{L}>\ell\}}e^{-2\theta\mathcal{L}}]e^{2\psi(z,\theta)} - \mathbb{E}_q[\mathbb{1}_{\{\mathcal{L}>\ell\}}e^{-\theta\mathcal{L}}]^2e^{2\psi(z,\theta)} \\
&= \left(\mathbb{E}_q[\mathbb{1}_{\{\mathcal{L}>\ell\}}]\mathbb{E}_q[e^{-2\theta\mathcal{L}}] + Cov[\mathbb{1}_{\{\mathcal{L}>\ell\}}, e^{-2\theta\mathcal{L}}]\right)e^{2\psi(z,\theta)} \\
&- \left(\mathbb{E}_q[\mathbb{1}_{\{\mathcal{L}>\ell\}}]\mathbb{E}_q[e^{-\theta\mathcal{L}}] + Cov[\mathbb{1}_{\{\mathcal{L}>\ell\}}, e^{-\theta\mathcal{L}}]\right)^2 e^{2\psi(z,\theta)} \\
&= \left(P(\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell|\mathcal{Z}=z)\mathbb{E}_q[e^{-2\theta\mathcal{L}}] + Cov[\mathbb{1}_{\{\mathcal{L}>\ell\}}, e^{-2\theta\mathcal{L}}]\right)e^{2\psi(z,\theta)} \\
&- \left(P(\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell|\mathcal{Z}=z)\mathbb{E}_q[e^{-\theta\mathcal{L}}] + Cov[\mathbb{1}_{\{\mathcal{L}>\ell\}}, e^{-\theta\mathcal{L}}]\right)^2 e^{2\psi(z,\theta)}
\end{aligned}$$

The moment generating function for a normal random variable X is

$$M(t) = \mathbb{E}[e^{tX}] = e^{\mu t+\frac{1}{2}\sigma^2t^2} \tag{8.1}$$

Therefore approximating $\mathcal{L}$ as a normal random variable we get

$$\mathbb{E}_q[e^{-\theta\mathcal{L}}] \approx e^{-\mu(z,\theta)\theta+\frac{1}{2}\sigma(z,\theta)^2\theta^2} \tag{8.2}$$

$$P(\mathcal{L}_N(\mathcal{Z},\mathcal{E}) \geq \ell|\mathcal{Z}=z) \approx 1-\Phi\left(\frac{\ell-\mu(z,\theta)}{\sigma(z,\theta)}\right) \tag{8.3}$$

Where, recalling that $\mathbb{1}_n^c(z) \sim q_n^c(z,\theta)$ under Exponential Twisting

$$\mu(z,\theta) = \mathbb{E}_q[\mathcal{L}(z,\mathcal{E})] = \sum_{n=1}^{N}\sum_{c=1}^{C}\omega_n^c\mathbb{E}_q\left[\mathbb{1}_n^c(z)\right] = \sum_{n=1}^{N}\sum_{c=1}^{C}\omega_n^c q_n^c(z,\theta) \tag{8.4}$$

and

$$\begin{aligned}\sigma(z,\theta)^2 &= \mathbb{V}_q[\mathcal{L}(z,\mathcal{E})]\\ &= \sum_{n=1}^{N}\omega_n^2\left(\frac{1}{2}\sum_{a=1}^{C}\sum_{b=1}^{C}(LGC_n^a - LGC_n^b)^2 q_n^a(z,\theta)q_n^b(z,\theta)\right)\\ &= \sum_{n=1}^{N}\omega_n^2\left(\sum_{a>b}^{C}(LGC_n^a - LGC_n^b)^2 q_n^a(z,\theta)q_n^b(z,\theta)\right)\end{aligned} \tag{8.5}$$

Plugging these results into our equation for $\mathbb{V}_q[\mathbb{1}_{\{\mathcal{L}>\ell\}}e^{\theta\mathcal{L}+\psi(z,\theta)}]$ we get

$$\begin{aligned}&\left(\left(1-\Phi\left(\frac{\ell-\mu(z,\theta)}{\sigma(z,\theta)}\right)\right)e^{-2\mu(z,\theta)\theta+2\sigma(z,\theta)^2\theta^2} + Cov\left[\mathbb{1}_{\{\mathcal{L}>\ell\}}, e^{-2\theta\mathcal{L}}\right]\right)e^{2\psi(z,\theta)} -\\ &\left(\left(1-\Phi\left(\frac{\ell-\mu(z,\theta)}{\sigma(z,\theta)}\right)\right)e^{-\mu(z,\theta)\theta+\frac{1}{2}\sigma(z,\theta)^2\theta^2} + Cov\left[\mathbb{1}_{\{\mathcal{L}>\ell\}}, e^{-\theta\mathcal{L}}\right]\right)^2 e^{2\psi(z,\theta)}\end{aligned}$$

The roadblock here is the covariance terms, which we have no easy way to evaluate or bound. One option is to ignore them altogether in the hopes that their influence is small. We found this option did not provide significant improvement over our existing approach, although we did not explore it thoroughly.

Appendices

Appendix A

Approximate Importance Sampler for Unbiased Estimator

Theorem 1 implies that $P(\mathcal{L}_N(z,\mathcal{E}) > \ell|\mathcal{Z} = z) \to 1 - \Phi\left(\frac{l-\mu(z)}{\sigma(z)}\right)$ as $N \to \infty$. The question is: given that π^* is a good importance sampler for $1 - \Phi\left(\frac{l-\mu(z)}{\sigma(z)}\right)$ is it also a good importance sampler for $P(\mathcal{L}_N(z,\mathcal{E}) > \ell|\mathcal{Z} = z)$? To answer this question, we introduce some notation

$$f_N(z) = P(\mathcal{L}_N(z,\mathcal{E}) > \ell|\mathcal{Z} = z)$$
$$f(z) = 1 - \Phi\left(\frac{l-\mu(z)}{\sigma(z)}\right)$$

Note that we have pointwise convergence of f_N to f. That is for any z

$$\lim_{N\to\infty} f_N(z) = f(z)$$

For fixed $z \in \mathbb{R}^S$, $\pi^*(z)$ and $\phi(z)$ are non-zero constants so we also have

$$\lim_{N\to\infty} \frac{f_N(z)\phi(z)}{\pi^*(z)} = \frac{f(z)\phi(z)}{\pi^*(z)}$$

$\mathbb{V}[]$ is an integral operation so we need something stronger than pointwise convergence for the next step of our argument. First we show that $\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)^2 \to \left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)^2$ pointwise.

Given any $\eta > 0$ we want to show $\exists M$ s.t. $\forall N \geq M$, $\left|\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)^2 - \left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)^2\right| \leq \eta$. A proof of this follows:

$$\left|\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)^2 - \left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)^2\right|$$
$$= \left|\left(\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right) - \left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)\right)\left(\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right) + \left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)\right)\right|$$

We know $\forall\delta > 0$ $\exists N_\delta$ s.t. $\left|\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right) - \left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)\right| \leq \delta$ for $N \geq N_\delta$. Under this assumption we

have

$$\begin{aligned}
&\left|\left(\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)-\left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)\right)\left(\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)+\left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)\right)\right| \\
&\leq \delta\left|\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)+\left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)\right| \\
&= \delta\left|\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)-\left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)+\left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)+\left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)\right| \\
&\leq \delta\left(\left|\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)-\left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)\right|+\left|\left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)+\left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)\right|\right) \\
&\leq \delta\left(\delta+2\left|\left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)\right|\right) \\
&= \delta\left(\delta+2f(z)\frac{\phi(z)}{\pi^*(z)}\right)
\end{aligned}$$

Now we merely need to show that $\eta = \delta\left(\delta + 2f(z)\frac{\phi(z)}{\pi^*(z)}\right)$ always has a (real positive) solution. This follows from the quadratic formula.

In fact through the dominated convergence theorem we see that the convergence is uniform because $f_N(z)\frac{\phi(z)}{\pi^*(z)} \leq \frac{\phi(z)}{\pi^*(z)}$. Since all the terms involved are positive this bound can be squared to arrive at a bound for f_N^2 as well. These facts together show that

$$\begin{aligned}
\lim_{N\to\infty}\mathbb{V}_{\pi^*}\left[\frac{f_N(z)\phi(z)}{\pi^*(z)}\right] &= \lim_{N\to\infty}\left[\int_{\mathbb{R}}\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)^2 dz - \left(\int_{\mathbb{R}} f_N(z)\frac{\phi(z)}{\pi^*(z)}dz\right)^2\right] \\
&= \lim_{N\to\infty}\int_{\mathbb{R}}\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)^2 dz - \lim_{N\to\infty}\left(\int_{\mathbb{R}} f_N(z)\frac{\phi(z)}{\pi^*(z)}dz\right)^2 \\
&= \lim_{N\to\infty}\int_{\mathbb{R}}\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)^2 dz - \left(\lim_{N\to\infty}\int_{\mathbb{R}} f_N(z)\frac{\phi(z)}{\pi^*(z)}dz\right)^2 \\
&= \int_{\mathbb{R}}\lim_{N\to\infty}\left(f_N(z)\frac{\phi(z)}{\pi^*(z)}\right)^2 dz - \left(\int_{\mathbb{R}}\lim_{N\to\infty} f_N(z)\frac{\phi(z)}{\pi^*(z)}dz\right)^2 \\
&= \int_{\mathbb{R}}\left(f(z)\frac{\phi(z)}{\pi^*(z)}\right)^2 dz - \left(\int_{\mathbb{R}} f(z)\frac{\phi(z)}{\pi^*(z)}dz\right)^2 \\
&= \mathbb{V}_{\pi^*}\left[\frac{f(z)\phi(z)}{\pi^*(z)}\right]
\end{aligned}$$

All of these arguments apply equally well to $\mathbb{V}_\phi[]$.

In summary we have

$$\lim_{N\to\infty} \mathbb{V}_{\phi}[f_N(z)] = \mathbb{V}_{\phi}[f(z)] \tag{A.1}$$

$$\lim_{N\to\infty} \mathbb{V}_{\pi^*}\left[f_N(z)\frac{\phi(z)}{\pi^*(z)}\right] = \mathbb{V}_{\pi^*}\left[f(z)\frac{\phi(z)}{\pi^*(z)}\right] \tag{A.2}$$

$$\mathbb{V}_{\pi^*}\left[f(z)\frac{\phi(z)}{\pi^*(z)}\right] < \mathbb{V}_{\phi}[f(z)] \tag{A.3}$$

Where the last equation comes from π^* being an importance sampler for the normal approximation.

We now want to show $\exists M(z) \in \mathbb{N}$ s.t. $\forall N > M(z)$ $\mathbb{V}_{\pi^*}\left[\frac{f_N(z)\phi(z)}{\pi^*(z)}\right] < \mathbb{V}_{\phi}[f_N(z)]$. This is simply saying that if $a_N \to a$, $b_N \to b$, and $b < a$ then there exists some cutoff point M such that $b_j < a_j$ for $j \geq M$. The proof of this statement follows: From the convergence of $a_N \to a$ we know there exists some M_a s.t $\forall N \geq M_a$ we have $|a_N - a| \leq \frac{a-b}{3} \iff -\frac{a-b}{3} \leq a_N - a \leq \frac{a-b}{3}$. This likewise holds true for b_N. Choosing $M = \max(M_a, M_b)$ we have for all $N \geq M$

$$\begin{aligned}
b_N &\leq b + \frac{a-b}{3} \\
&\leq b - a + a + \frac{a-b}{3} \\
&= a - (a-b) + \frac{a-b}{3} \\
&= a - \frac{2(a-b)}{3} \\
&< a - \frac{(a-b)}{3} \\
&\leq a_N
\end{aligned}$$

Therefore for N large enough $\pi^*(z)$ is a good importance sampler for $P(\mathcal{L}_N(z, \mathcal{E}) > \ell | \mathcal{Z} = z)$.

Appendix B

Likelihood Ratio for Exponential Twisting

$$
\begin{aligned}
\prod_{n=1}^{N}\prod_{c=1}^{C}\left(\frac{p_n^c}{q_n^c}\right)^{\mathbb{1}_n^c} &= \exp\left(\ln\left(\prod_{n=1}^{N}\prod_{c=1}^{C}\left(\frac{p_n^c}{q_n^c}\right)^{\mathbb{1}_n^c}\right)\right) \\
&= \exp\left(\sum_{n=1}^{N}\sum_{c=1}^{C}\ln\left(\left(\frac{p_n^c}{q_n^c}\right)^{\mathbb{1}_n^c}\right)\right) \\
&= \exp\left(\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbb{1}_n^c\ln\left(\frac{p_n^c}{q_n^c}\right)\right) \\
&= \exp\left(\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbb{1}_n^c\ln\left(\frac{\sum_{k=1}^{C}p_n^k e^{\theta\omega_n^k}}{e^{\theta\omega_n^c}}\right)\right) \\
&= \exp\left(\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbb{1}_n^c\left[\ln\left(\sum_{k=1}^{C}p_n^k e^{\theta\omega_n^k}\right)-\ln\left(e^{\theta\omega_n^c}\right)\right]\right) \\
&= \exp\left(\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbb{1}_n^c\ln\left(\sum_{k=1}^{C}p_n^k e^{\theta\omega_n^k}\right)-\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbb{1}_n^c\theta\omega_n^c\right) \\
&= \exp\left(\sum_{n=1}^{N}\ln\left(\sum_{k=1}^{C}p_n^k e^{\theta\omega_n^k}\right)\sum_{c=1}^{C}\mathbb{1}_n^c-\theta\mathcal{L}\right) \\
&= \exp\left(\sum_{n=1}^{N}\ln\left(\sum_{k=1}^{C}p_n^k e^{\theta\omega_n^k}\right)-\theta\mathcal{L}\right) \\
&= \exp\left(\psi(\theta)-\theta\mathcal{L}\right)
\end{aligned}
$$

sum to 1

Appendix C

Convex Bound on Second Moment for Inner Level Importance Sampler

summation of convex function is convex

$$\psi(\theta) = \sum_{n=1}^{N} \ln\left(\sum_{c=1}^{C} p_n^c e^{\theta\omega_n^c}\right) = \sum_{n=1}^{N} \psi_n(\theta)$$

To show that $\psi(\theta)$ is convex it is enough to show that each $\psi_n(\theta)$ is convex. In order to show this we will prove that $\psi_n''(\theta) \geq 0$

why the minus… since 2nd order derivative

$$\begin{aligned}
\psi_n''(\theta) =& \frac{\sum_{c=1}^{C} p_n^c (\omega_n^c)^2 e^{\theta\omega_n^c}}{\sum_{c=1}^{C} p_n^c e^{\theta\omega_n^c}} - \frac{\left(\sum_{c=1}^{C} p_n^c \omega_n^c e^{\theta\omega_n^c}\right)^2}{\left(\sum_{c=1}^{C} p_n^c e^{\theta\omega_n^c}\right)^2} \geq 0 \\
\iff & \left(\sum_{c=1}^{C} p_n^c e^{\theta\omega_n^c}\right)\left(\sum_{c=1}^{C} p_n^c (\omega_n^c)^2 e^{\theta\omega_n^c}\right) - \left(\sum_{c=1}^{C} p_n^c \omega_n^c e^{\theta\omega_n^c}\right)^2 \geq 0 \\
\iff & \left(\sum_{c=1}^{C} p_n^c \omega_n^c e^{\theta\omega_n^c}\right)^2 \leq \left(\sum_{c=1}^{C} p_n^c e^{\theta\omega_n^c}\right)\left(\sum_{c=1}^{C} p_n^c (\omega_n^c)^2 e^{\theta\omega_n^c}\right)
\end{aligned}$$

Let $\vec{x}_n, \vec{y}_n \in \mathbb{R}^C, x_n^c = \sqrt{p_n^c e^{\theta\omega_n^c}}, y_n^c = \omega_n^c x_n^c$. Then the inequality becomes

$$\left(\sum_{c=1}^{C} x_n^c y_n^c\right)^2 \leq \left(\sum_{c=1}^{C} (x_n^c)^2\right)\left(\sum_{c=1}^{C} (y_n^c)^2\right) \tag{C.1}$$

This is the Cauchy-Schwarz inequality. To prove strict convexity we note that Cauch-Schwarz is equality iff $\vec{y}_n = \lambda\vec{x}_n, \lambda \in \mathbb{R}$ or one of $\vec{x}_n$ or $\vec{y}_n$ is $\vec{0}$. While this is possible mathematically it is not likely in the credit model case. Moreover if at least one $\psi_n(\theta)$ is strictly convex then $\psi(\theta)$ is strictly convex. Therefore the inequality in equation (C.1) is strict in almost all cases.

The fact that $\psi(0) = 0$ follows immediately from $\sum_{c=1}^{C} p_n^c = 1$.

Bibliography

[1] Paolo Brandimarte. *Numerical Methods in Finance and Economics: A MATLAB-Based Introduction*. Wiley-Interscience, 2006.

[2] Steve Brooks, Andrew Gelman, Galin L. Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall, 2011.

[3] James Bucklew. *Introduction to Rare Event Simulation*. Springer, 2004.

[4] C.-D. Fuh and C.-J. Wang. Efficient Simulation for Portfolio Credit Risk in Normal Mixture Copula Models. *ArXiv e-prints*, November 2017.

[5] Paul Glasserman and Jingyi Li. Importance Sampling for Portfolio Credit Risk. *Management Science*, pages 1643–1656, 2005.

[6] S. Gould, B. Fernando, A. Cherian, P. Anderson, R. Santa Cruz, and E. Guo. On Differentiating Parameterized Argmin and Argmax Problems with Application to Bi-level Optimization. *ArXiv e-prints*, July 2016.

[7] Meng Han. Approximations and Optimization for Portfolio Credit Risk in the Gaussian Copula Factor Model. *Prelimmary PhD Thesis Draft. Computer Science Dept, University of Toronto*, 2017.

[8] Robert C. Merton. On the pricing of corporate debt: The risk structure of interest rates. *Journal of Finance*, (29):449–470, 1973.

[9] J.S. Sadowsky. On the optimality and stability of exponential twisting in Monte Carlo estimation. *IEEE Transactions on Information Theory*, pages 119–128, 1993.

[10] Zhe Wang. New Approaches to Importance Sampling for Portfolio Credit Risk Valuation. *MSc Research Paper. Computer Science Dept, University of Toronto*, 2015. http://www.cs.toronto.edu/pub/reports/na/Zhe.Wang.MSC.Thesis.pdf.