risks

MDPI

Article

Importance Sampling in the Presence of PD-LGD Correlation

Adam Metzler [1,*] and Alexandre Scott [2]

[1] Department of Mathematics, Wilfrid Laurier University, Waterloo, ON N2L 3C5, Canada
[2] Department of Applied Mathematics, University of Western Ontario, London, ON N6A 3K7, Canada; alexandre.scott202@gmail.com
* Correspondence: ametzler@wlu.ca

Received:20 January 2020; Accepted: 5 March 2020; Published: 10 March 2020

Abstract: This paper seeks to identify computationally efficient importance sampling (IS) algorithms for estimating large deviation probabilities for the loss on a portfolio of loans. Related literature typically assumes that realised losses on defaulted loans can be predicted with certainty, i.e., that loss given default (LGD) is non-random. In practice, however, LGD is impossible to predict and tends to be positively correlated with the default rate and the latter phenomenon is typically referred to as PD-LGD correlation (here PD refers to probability of default, which is often used synonymously with default rate). There is a large literature on modelling stochastic LGD and PD-LGD correlation, but there is a dearth of literature on using importance sampling to estimate large deviation probabilities in those models. Numerical evidence indicates that the proposed algorithms are extremely effective at reducing the computational burden associated with obtaining accurate estimates of large deviation probabilities across a wide variety of PD-LGD correlation models that have been proposed in the literature.

Keywords: importance sampling; acceptance-rejection sampling; portfolio credit risk; tail probabilities; large deviation probabilities; stochastic recovery; PD-LGD correlation; credit risk; loss probabilities

1. Introduction

This paper seeks to identify computationally efficient importance sampling (IS) algorithms for estimating large deviation probabilities for the loss on a portfolio of loans. Related literature assumes that realised losses on defaulted loans can be predicted with certainty, i.e., that loss given default (LGD) is non-random. In practice, however, LGD is impossible to predict and tends to be positively correlated with the default rate and the latter phenomenon is typically referred to as PD-LGD correlation (here PD refers to probability of default, which is often used synonymously with default rate). There is a large literature on modelling stochastic LGD and PD-LGD correlation, but there is a paucity of literature on using importance sampling to estimate large deviation probabilities in those models. This gap in the literature was brought to our attention by a risk management professional at a large Canadian financial institution, and filling that gap is the ultimate goal of this paper.

Problem Formulation and Related Literature

Consider a portfolio of N exposures of equal size. Let $L_1, L_2, \ldots, L_N$ denote the losses on the individual loans, expressed as a percentage of notional value. The percentage loss on the entire portfolio is:

$$\bar{L}_N := \frac{1}{N}\sum_{i=1}^{N} L_i \,. \tag{1}$$

We are interested in using IS to estimate large deviation probabilities of the form:

$$p_x := \mathbb{P}(\bar{L}_N \geq x) \,, \tag{2}$$

where $x >> \mathbb{E}[L_i] = \mathbb{E}[\bar{L}_N]$ is some large, user-defined, threshold.

In practice the number of exposures is large (e.g., in the thousands) and prudent risk management requires one to assume that the individual losses are correlated. In practice, then, $\bar{L}_N$ is the average of a large number of correlated variables. As such, its probability distribution is highly intractable and Monte Carlo is the method of choice for approximating p_x. As the probability of interest is typically small (e.g., on the order of 10^{-3} or 10^{-4}), the computational burden required to obtain an accurate estimate of p_x using Monte Carlo can be prohibitive. For instance if p_x is on the order of 10^{-3} and N is on the order of 1000 then, in the absence of any variance reduction techniques, the sample size required to reduce the estimator's relative error[1] to 10% is on the order of one hundred thousand. Since each realisation of $\bar{L}_N$ requires simulation of one thousand individual losses, a sample size of 100,000 requires one to generate one hundred million variables. If the desired degree of accuracy is reduced to 1%, the number of variables that must be generated increases to a staggering 10 billion.

Importance sampling (IS) is a variance reduction technique that has the potential to significantly reduce the computational burden associated with obtaining accurate estimates of large deviation probabilities. In the present context, effective IS algorithms have been identified for a variety of popular risk management models, but most are limited to the special case that loss given default (LGD) is non-random. The seminal paper in the area is (Glasserman and Li 2005), other papers include (Chan and Kroese 2010) and (Scott and Metzler 2015). It is well documented empirically, however, that portfolio-level LGD is not only stochastic, but positively correlated with the portfolio-level default rate as seen, for instance, in any of the studies listed in (Kupiec 2008) or (Frye and Jacobs 2012). This phenomenon is typically referred to as PD-LGD correlation. (Miu and Ozdemir 2006) show that ignoring PD-LGD correlation when it is in fact present can lead to material underestimates of portfolio risk measures.

There is a large literature on modelling PD-LGD correlation (Frye 2000); (Pykhtin 2003); (Miu and Ozdemir 2006); (Kupiec 2008); (Sen 2008); (Witzany 2011); (de Wit 2016); (Eckert et al. 2016); and others listed in (Frye and Jacobs 2012), but there is a much smaller literature on using IS to estimate large deviation probabilities in such models. To the best of our knowledge only (Deng et al. 2012) and (Jeon et al. 2017) have developed algorithms that allow for PD-LGD correlation (the former paper considers a dynamic intensity-based framework, the latter considers a static model with asymmetric and heavy-tailed risk factors). The present paper contributes to this nascent literature by developing algorithms that can be applied in a wide variety of PD-LGD correlation models that have been proposed in the literature, and are popular in practice.

The paper is structured as follows. Section 2 outlines important assumptions, notation, and terminology. Section 3 theoretically motivates the proposed algorithm in a general setting, and Section 4 discusses a few practical issues that arise when implementing the algorithm. Section 5 describes a general framework for PD-LGD correlation modelling that includes, as special cases, many of the models that have been developed in the literature and Section 6 describes how to implement the proposed algorithm in this general framework. Numerical results are presented and discussed in Section 7 and demonstrate that the proposed algorithms are extremely effective at reducing the computational burden required to obtain an accurate estimate of p_x.

---

[1] Relative error is the preferred measure of accuracy for large deviation probabilities. If $\hat{p}_x$ is an estimator of p_x, its relative error is defined as $\mathrm{SD}(\hat{p}_x)/p_x$, where SD denotes standard deviation.

2. Assumptions, Notation and Terminology

We assume that individual losses are of the form $L_i = \mathcal{L}(\mathbf{Z}, \mathbf{Y}_i)$, where $\mathcal{L}$ is some deterministic function, $\mathbf{Z} = (Z_1, \ldots, Z_d)$ is a d-dimensional vector of systematic risk factors that affect all exposures, and $\mathbf{Y}_i$ is a vector of idiosyncratic risk factors that only affect exposure i. We assume that $\mathbf{Z}, \mathbf{Y}_1, \mathbf{Y}_2, \ldots$ are independent, and that the $\mathbf{Y}_i$ are identically distributed. The primary role of the systematic risk factors is to induce correlation among the individual exposures, and it is common to interpret the realised values of the systematic risk factors as determining the overall macroeconomic environment. It is worth noting that the we do not require the components of $\mathbf{Z}$ to be independent of one another, etc. for the components of $\mathbf{Y}_i$.

2.1. Large Portfolios and the Region of Interest

In a large portfolio, the influence of the idiosyncratic risk factors is negligible. Indeed, since individual losses are conditionally independent, given the realised values of the systematic risk factors, we have the almost sure limit:

$$\lim_{N\to\infty} \bar{L}_N = \mu(\mathbf{Z}) \,, \tag{3}$$

where

$$\mu(\mathbf{z}) := \mathbb{E}[L_i|\mathbf{Z} = \mathbf{z}] = \mathbb{E}[\bar{L}_N|\mathbf{Z} = \mathbf{z}] \,. \tag{4}$$

Since $\mu(\mathbf{Z}) \approx \bar{L}_N$ for large N by Equation (3), the random variable $\mu(\mathbf{Z})$ is often called the large portfolio approximation (LPA) to $\bar{L}_N$. The LPA is often used to formalise the intuitive notion that, in a large portfolio, all risk is systematic (i.e., idiosyncratic is "diversified away"). We define the region of interest as the set:

$$\{\mathbf{z} \in \mathbb{R}^d : \ \mu(\mathbf{z}) \geq x\} \,. \tag{5}$$

The region of interest is "responsible" for large deviations in the sense that:

$$\lim_{N\to\infty} \mathbb{P}(\mu(\mathbf{Z}) \geq x|\bar{L}_N \geq x) = 1 \tag{6}$$

for most values[2] of x. Together, Equations (3) and (6) suggest that for large portfolios, it is relatively more important to identify an effective IS distribution for the systematic risk factors, as compared to the idiosyncratic risk factors.

2.2. Systematic Risk Factors

We assume that $\mathbf{Z}$ is continuous and let $f(\mathbf{z})$ denote its joint density. We assume that f is a member of an exponential family (see Bickel and Doksum 2001 for definitions and important properties) with natural sufficient statistic $S : \mathbb{R}^d \mapsto \mathbb{R}^p$. Any other member of the family can be put in the form:

$$f_\lambda(\mathbf{z}) := \exp(\lambda^T S(\mathbf{z}) - K(\lambda)) \cdot f(\mathbf{z}) \,, \tag{7}$$

where $K(\cdot)$ is the cumulant generating function (cgf) of $S(\mathbf{Z})$ and $\lambda \in \mathbb{R}^p$ is such that $K(\lambda)$ is well-defined. The parameter λ is called the natural parameter of the family in Equation (7). Appendix B embeds the Gaussian and multivariate t families into this general framework.

---

[2] In light of the almost sure limit in Equation (3), we have that $\bar{L}_N$ converges to $\mu(\mathbf{Z})$ in distribution, which implies that Equation (6) is valid for all values of x such that $\mathbb{P}(\mu(\mathbf{Z}) = x) = 0$. If $\mu(\mathbf{Z})$ is a continuous random variable (which it is in most cases of practical interest) then Equation (6) is satisfied for every value of x.

We will eventually be using densities of the form in Equation (7) as IS densities for the systematic risk factors. The associated IS weight is:

$$\frac{f(\mathbf{Z})}{f_\lambda(\mathbf{Z})} = \exp(-\lambda^T S(\mathbf{Z}) + K(\lambda)) \,, \tag{8}$$

and it will be important to know when the variance of the IS weight is finite. The following observation is readily verified.

Remark 1. *If $\mathbf{Z} \sim f_\lambda$, then Equation (8) has finite variance if and only if both $K(\lambda)$ and $K(-\lambda)$ are well defined.*

A standard result in the theory of exponential families is that:

$$\nabla K(\lambda) = \mathbb{E}_\lambda[S(\mathbf{Z})] \,, \tag{9}$$

where ∇ denotes gradient and $\mathbb{E}_\lambda$ denotes expectation with respect to the density f_λ.

2.3. Individual Losses

We assume that L_i takes values in the unit interval. In general L_i will have a point mass at zero (if it did not, the loan would not be prudent) and the conditional distribution of L_i, given that $L_i > 0$, is called the (account-level) LGD distribution. We allow the LGD distribution to be arbitrary in the sense that it could be either discrete or continuous, or a mixture of both. This contrasts with the case of non-random LGD, where the LGD distribution is degenerate at a single point. We let $\ell_{\max} \in (0, 1]$ denote the supremum of the support of L_i. Individual losses will therefore never exceed $\ell_{\max}$ but could take on values arbitrarily close (and possibly equal) to $\ell_{\max}$.

Remark 2. *Despite the fact that L_i is not a continuous variable, in what follows we will proceed as if it was and make repeated reference to its "density." This is done without loss of generality, and in an interest of simplifying the presentation and discussion. Nothing in the sequel requires L_i to be a continuous variable, and everything carries over to the case where it is either discrete or continuous, or has both a discrete and continuous component.*

For $\mathbf{z} \in \mathbb{R}^d$ we let $g(\ell|\mathbf{z})$ denote the conditional density of L_i, given that $\mathbf{Z} = \mathbf{z}$. We assume that the support of $g(\cdot|\mathbf{z})$ is identical to the unconditional support, in particular it does not depend on the value of $\mathbf{z}$. Note that $\mu(\mathbf{z})$ is the mean of $g(\cdot|\mathbf{z})$.

In practice (i.e., for all of the PD-LGD correlation models listed in the introduction) $g(\cdot|\mathbf{z})$ is not a member of an established parametric family, and direct simulation from $g(\cdot|\mathbf{z})$ using a standard technique such as inverse transform or rejection sampling is not straightforward. Simulation from $g(\cdot|\mathbf{z})$ is most easily accomplished by simulating the idiosyncratic risk factors, $\mathbf{Y}_i$, from their density, say $\eta(\mathbf{y})$, and then setting $L_i = \mathcal{L}(\mathbf{z}, \mathbf{Y}_i)$. In other words, in order to simulate from $g(\cdot|\mathbf{z})$ we make use of the fact that $L_i = \mathcal{L}(\mathbf{z}, \mathbf{Y}_i)$ is a drawing from $g(\cdot|\mathbf{z})$ whenever $\mathbf{Y}_i$ is a drawing from $\eta(\cdot)$.

For $\theta \in \mathbb{R}$ and $\mathbf{z} \in \mathbb{R}^d$ we let:

$$k(\theta, \mathbf{z}) := \log(\mathbb{E}[\exp(\theta L_i)|\mathbf{Z} = \mathbf{z}])$$

and

$$k'(\theta, \mathbf{z}) := \frac{\partial k}{\partial \theta}(\theta, \mathbf{z}) \,.$$

Then $k(\cdot, \mathbf{z})$ is the conditional cgf of L_i, given that $\mathbf{Z} = \mathbf{z}$, and $k'(\cdot, \mathbf{z})$ is its first derivative. In practice, neither $k(\cdot, \mathbf{z})$ nor $k'(\cdot, \mathbf{z})$ is available in closed form. In the examples we consider later in the paper each can be expressed as a one-dimensional integral, but the numerical values of those integrals must

be approximated using quadrature. This contrasts with the case of non-random LGD, where the conditional cgf can be computed in closed form[3].

For $x \in (0, \ell_{\max})$ and $\mathbf{z} \in \mathbb{R}^d$ we let $\hat{\theta}(x, \mathbf{z})$ denote the unique solution to the equation $k'(\theta, \mathbf{z}) = \max(x, \mu(\mathbf{z}))$. We often suppress dependence on x and $\mathbf{z}$, and simply write $\hat{\theta}$ instead of $\hat{\theta}(x, \mathbf{z})$. That $\hat{\theta}$ is well-defined follows immediately from the developments in Appendix A.1. Based on the discussion there we find that $\hat{\theta}$ is zero whenever $\mathbf{z}$ lies in the region of interest, and is strictly positive otherwise.

Remark 3. *In practice, the value of $\hat{\theta}$ cannot be computed in closed form and must be approximated using a numerical root-finding algorithm. Since each evaluation of the function $k'(\cdot, \mathbf{z})$ requires quadrature, computing $\hat{\theta}$ is straightforward but relatively time consuming. This contrasts with the case of non-random LGD, where $\hat{\theta}$ can be computed in closed form at essentially no cost.*

For $\mathbf{z} \in \mathbb{R}^d$ we let $q(\cdot, \mathbf{z})$ denote the Legendre transform of $k(\cdot, \mathbf{z})$ over $[0, \infty)$. That is,

$$q(x, \mathbf{z}) := \max_{\theta \geq 0}(\theta x - k(\theta, \mathbf{z})) = \hat{\theta} x - k(\hat{\theta}, \mathbf{z}) \, . \tag{10}$$

That $\hat{\theta}$ is the uniquely defined point at which the function $\theta \mapsto \theta x - k(\theta, \mathbf{z})$ attains its maximum on $[0, \infty)$ follows from the developments in Appendix A.2. Based on the discussion there, we find that both $\hat{\theta}$ and q are equal to zero whenever $\mathbf{z}$ lies in the region of interest, and that both are strictly positive otherwise.

2.4. Conditional Tail Probabilities

Given the realised values of the systematic risk factors, individual losses are independent. Large deviations theory can therefore provide useful insights into the large-N behaviour of the tail probability $\mathbb{P}(\bar{L}_N \geq x | \mathbf{Z} = \mathbf{z})$. For instance, Chernoff's bound yields the estimate:

$$\mathbb{P}(\bar{L}_N > x | \mathbf{Z} = \mathbf{z}) \leq \exp(-N q(x, \mathbf{z})) \, , \tag{11}$$

and Cramér's (large deviation) theorem yields the limit:

$$\lim_{N \to \infty} \frac{\log(\mathbb{P}(\bar{L}_N > x | \mathbf{Z} = \mathbf{z}))}{N} = -q(x, \mathbf{z}) \, . \tag{12}$$

Together these results are often used to justify the approximation:

$$\mathbb{P}(\bar{L}_N > x | \mathbf{Z} = \mathbf{z}) \approx \exp(-N q(x, \mathbf{z})) \, , \tag{13}$$

which will be used repeatedly throughout the paper. The approximation in Equation (13) is often called the large deviation approximation (LDA) to the tail probability $\mathbb{P}(\bar{L}_N > x | \mathbf{Z} = \mathbf{z})$. Note that since $q(x, \mathbf{z}) = 0$ whenever $\mu(\mathbf{z}) \geq x$, the LDA suggests that $\mathbb{P}(\bar{L}_N > x | \mathbf{Z} = \mathbf{z}) \approx 1$ whenever $\mathbf{z}$ lies in the region of interest.

2.5. Conditional Densities

Let $\mathbf{L} = (L_1, \ldots, L_N)$, noting that $\mathbf{L}$ takes values in $[0, \ell_{\max}]^N$. For $\mathbf{z} \in \mathbb{R}^d$ and $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_N) \in [0, \ell_{\max}]^N$, we let $h_x(\mathbf{z}, \boldsymbol{\ell})$ denote the conditional density of $(\mathbf{Z}, \mathbf{L})$, given that $\bar{L}_N > x$. Then h_x is given by:

$$h_x(\mathbf{z}, \boldsymbol{\ell}) = \frac{f(\mathbf{z}) \cdot \prod_{i=1}^N g(\ell_i | \mathbf{z})}{p_x} \cdot \mathbf{1}_{\{\boldsymbol{\ell} \in A_{N,x}\}} \, , \tag{14}$$

---

[3] In the case of non-random LGD we have $k(\theta, \mathbf{z}) = \log(1 + (e^{(1-R)\theta} - 1) \cdot \mathbb{P}(L_i > 0 | \mathbf{Z} = \mathbf{z}))$, where R is the known recovery rate on the exposure.

where $A_{N,x}$ is the set of points $\boldsymbol{\ell} \in [0, \ell_{\max}]^N$ for which $N^{-1}\sum_{i=1}^N \ell_i > x$.

We let $f_x(\mathbf{z})$ denote the conditional density of the systematic risk factors, given that $\bar{L}_N > x$, noting that:

$$f_x(\mathbf{z}) = \frac{\mathbb{P}(\bar{L}_N > x | \mathbf{Z} = \mathbf{z})}{\mathbb{P}(\bar{L}_N \geq x)} \cdot f(\mathbf{z}) \,. \tag{15}$$

In the examples we consider the mean of f_x tends to lie inside, but close to the boundary of, the region of interest. And relative to the unconditional density f, the conditional density f_x tends to be much more concentrated about its mean.

Finally, we let $g_x(\ell|\mathbf{z})$ denote the conditional density of an individual loss, given that $\mathbf{Z} = \mathbf{z}$ and $\bar{L}_N > x$, noting that:

$$g_x(\ell|\mathbf{z}) = \frac{\mathbb{P}(\bar{L}_{N-1} > x + \frac{x-\ell}{N-1} | \mathbf{Z} = \mathbf{z})}{\mathbb{P}(\bar{L}_N > x | \mathbf{Z} = \mathbf{z})} \cdot g(\ell|\mathbf{z}) \,. \tag{16}$$

If the realised value of $\mathbf{z}$ lies inside the region of interest, the conditional density $g_x(\cdot|\mathbf{z})$ tends to resemble the unconditional density $g(\cdot|\mathbf{z})$. Intuitively, for such values of $\mathbf{z}$ the LDA informs that the event $\{\bar{L}_N > x\}$ is very likely, and conditioning on its occurrence is not overly informative. If the realised value of $\mathbf{z}$ does not lie in the region of interest then $g_x(\cdot|\mathbf{z})$ tends to resemble the exponentially tilted version of $g(\cdot|\mathbf{z})$ whose mean is exactly x. See Appendix A.3 for more details.

Neither h_x, f_x, nor g_x are numerically tractable, but as we will soon see they do serve as useful benchmarks against which to compare candidate IS densities. In addition, it is worth noting here that the representations of Equations (15) and (16) lend themselves to numerical approximation via the LDA in Equation (13).

3. Proposed Algorithm

In practice, the most common approach to estimating p_x via Monte Carlo simulation in this framework is summarised in Algorithm 1 below.

Algorithm 1 Standard Monte Carlo Algorithm for Estimating p_x

1: Simulate M i.i.d. copies of the systematic risk factors. Think of these as different economic scenarios and denote the simulated values by $\mathbf{z}_1, \ldots, \mathbf{z}_M$.
2: For each scenario m:
   (a) Simulate the idiosyncratic risk factors for each exposure. Denote the simulated values $\mathbf{y}_{1,m}, \ldots, \mathbf{y}_{N,m}$.
   (b) Set $\ell_{i,m} = \mathcal{L}(\mathbf{z}_m, \mathbf{y}_{i,m})$ for each exposures i, and $\bar{\ell}_m = \frac{1}{N}\sum_{i=1}^N \ell_{i,m}$.
3: Return $\widehat{p}_x = \frac{1}{M}\sum_{m=1}^M \mathbf{1}_{\{\bar{\ell}_m > x\}}$.

Algorithm 1 consists of two stages. In the first stage one simulates the systematic risk factors, and in the second stage one simulates the idiosyncratic risk factors for each exposure. Mathematically, the first stage induces independence among the individual exposures, so that the second stage amounts to simulating a large number of i.i.d. variables. Intuitively, it is useful to think of the first stage as determining the prevailing macroeconomic environment, which fixes economy-wide quantities such as default and loss-given-default rates. The second stage of the algorithm overlays idiosyncratic noise on top of economy-wide rates, to arrive at the default and loss-given-default rates for a particular portfolio.

Relative error is the preferred measure of accuracy for estimators of rare event probabilities. The relative error of the estimator $\widehat{p}_x$ in Algorithm 1 is:

$$\frac{1}{\sqrt{M}}\sqrt{\frac{1-p_x}{p_x}} \,,$$

and the sample size required to ensure the relative error does not exceed some predetermined threshold ϵ is:

$$M(\epsilon) = \frac{1}{\epsilon^2}\frac{1-p_x}{p_x} \,. \tag{17}$$

The number of variables that must be generated in order to achieve the desired degree of accuracy ϵ is therefore $(N+d)\cdot M(\epsilon)$, which grows without bound as $p_x \to 0$. For instance if $p_x = 10^{-3}$, $N = 10^3$, $d = 2$, and $\epsilon = 5\cdot 10^{-2}$ then the number of variables that must be generated is approximately four hundred million, which is an enormous computational burden for a modest degree of accuracy. In the next section we discuss general principles for selecting an IS algorithm that can reduce the computational burden required to obtain an accurate estimate of p_x.

3.1. General Principles

For practical reasons, we insist that our IS procedure retains conditional independence of individual losses, given the realised value of the systematic risk factors. This is important because it allows us to reduce the problem of simulating a large number of dependent variables to the (much) more computationally efficient problem of simulating a large number of independent variables.

In the first stage we simulate the systematic risk factors from the IS density $f_{IS}(\mathbf{z})$. The IS weight associated with this first stage is therefore:

$$\Lambda_1(\mathbf{z}) := \frac{f(\mathbf{z})}{f_{IS}(\mathbf{z})} \,.$$

In the second stage we simulate the individual losses as i.i.d. drawings from the density $g_{IS}(\ell|\mathbf{z})$. The IS weight associated with this second stage is:

$$\Lambda_2(\mathbf{z},\boldsymbol{\ell}) = \prod_{i=1}^{N}\frac{g(\ell_i|\mathbf{z})}{g_{IS}(\ell_i|\mathbf{z})} \,,$$

and the IS density from which we sample $(\mathbf{Z},\mathbf{L})$ is therefore of the form:

$$h_{IS}(\mathbf{z},\boldsymbol{\ell}) = f_{IS}(\mathbf{z})\cdot\prod_{i=1}^{N} g_{IS}(\ell_i|\mathbf{z}) \,. \tag{18}$$

The so-described algorithm, with as-yet unspecified IS densities, is summarised in Algorithm 2.

Algorithm 2 IS Algorithm for Estimating p_x

1: Simulate M i.i.d. copies of the systematic risk factors from the density $f_{IS}(\mathbf{z})$. Think of these as different economic scenarios and denote the simulated values by $\mathbf{z}_1,\ldots,\mathbf{z}_M$.
2: For each scenario m:
   (a) Independently simulate $\ell_{1,m},\ell_{2,m},\ldots,\ell_{N,m}$ from the density $g_{IS}(\cdot|\mathbf{z}_m)$.
   (b) Set $\bar{\ell}_m = \frac{1}{N}\sum_{i=1}^{N}\ell_{i,m}$.
3: Return $\widehat{p}_x = \frac{1}{M}\sum_{m=1}^{M}\Lambda_1(\mathbf{z}_m)\cdot\Lambda_2(\mathbf{z}_m,\boldsymbol{\ell}_m)\cdot\mathbf{1}_{\{\bar{\ell}_m>x\}}$, where $\boldsymbol{\ell}_m = (\ell_{1,m},\ldots,\ell_{N,m})$.

It is important to note that in the second stage, we will not be simulating individual losses directly from the (conditional) IS density g_{IS}. Rather, we will simulate the idiosyncratic risk factors $\mathbf{Y}_i$ in such a way as to ensure that for a given value of $\mathbf{z}$, the variable $L_i = \mathcal{L}(\mathbf{z},\mathbf{Y}_i)$ has the desired density g_{IS}.

Focusing on the "indirect" IS density of L_i, as opposed to "direct" IS density of $\mathbf{Y}_i$, allows us to identify a much more effective second stage algorithm[4].

The estimator $\widehat{p}_x$ produced by Algorithm 2 is demonstrably unbiased and its variance is:

$$\mathbb{E}_{IS}[(\Lambda(\mathbf{Z},\mathbf{L})\cdot\mathbf{1}_{\{\bar{L}_N>x\}}-p_x)^2]=p_x^2\cdot\mathbb{E}_{IS}[(\Lambda_x(\mathbf{Z},\mathbf{L})\cdot\mathbf{1}_{\{\bar{L}_N>x\}}-1)^2]\,, \tag{19}$$

where $\mathbb{E}_{IS}$ denotes expectation under the IS distribution, $\Lambda(\mathbf{z},\boldsymbol{\ell}):=\Lambda_1(\mathbf{z})\cdot\Lambda_2(\mathbf{z},\boldsymbol{\ell})$ and

$$\Lambda_x(\mathbf{z},\boldsymbol{\ell}):=\frac{\Lambda(\mathbf{z},\boldsymbol{\ell})}{p_x}\,.$$

Note that Λ_x is the ratio of (i) the IS density in Equation (18) to (ii) the conditional density in Equation (14). The estimator's squared relative error can then be decomposed as:

$$\mathbb{E}_{IS}[(\Lambda_x(\mathbf{Z},\mathbf{L})-1)^2\cdot\mathbf{1}_{\{\bar{L}_N>x\}}]+[1-\mathbb{P}_{IS}(\bar{L}_N>x)]\,, \tag{20}$$

where $\mathbb{P}_{IS}$ denotes probability under the IS distribution.

Inspecting Equation (20) we see that an effective IS density should (i) assign a high probability to the event of interest and (ii) should resemble the conditional density in Equation (14) as closely as possible, in the sense that the ratio Λ_x should deviate as little as possible from unity. Clearly, an estimator that satisfies (ii) should also satisfy (i), since h_x assigns probability one to the event that $\bar{L}_N>x$. The task now is to identify a density of the form in Equation (18) that resembles the ideal density in Equation (14), in some sense.

3.2. *Identifying the Ideal IS Densities*

Our measure of similarity is Kullback–Leibler divergence (KLD), or divergence for short. See Chatterjee and Diaconis (2018) for a general discussion of the merits of minimum divergence as a criteria for identifying effective IS distributions. We begin by writing:

$$\frac{h_x(\mathbf{z},\boldsymbol{\ell})}{h_{IS}(\mathbf{z},\boldsymbol{\ell})}=\frac{f_x(\mathbf{z})}{f_{IS}(\mathbf{z})}\cdot\frac{\tilde{g}_x(\boldsymbol{\ell}|\mathbf{z})}{\tilde{g}_{IS}(\boldsymbol{\ell}|\mathbf{z})}\,, \tag{21}$$

where for fixed $\mathbf{z}$,

$$\tilde{g}_x(\boldsymbol{\ell}|\mathbf{z})=\frac{\prod_{i=1}^N g(\ell_i|\mathbf{z})}{\mathbb{P}(\bar{L}_N>x|\mathbf{Z}=\mathbf{z})}\cdot\mathbf{1}_{\{\boldsymbol{\ell}\in A_{N,x}\}}$$

is the joint density of N independent variables having marginal density $g(\cdot|\mathbf{z})$, conditioned on their average value exceeding the threshold x, and

$$\tilde{g}_{IS}(\boldsymbol{\ell}|\mathbf{z})=\prod_{i=1}^N g_{IS}(\ell_i|\mathbf{z})\,.$$

is the joint density of N independent variables having marginal density $g_{IS}(\cdot|\mathbf{z})$.

Using Equation (21) it is straightforward to decompose the divergence of h_{IS} from h_x as:

$$D(h_{IS}||h_x)=D(f_{IS}||f_x)+\mathbb{E}\left[D(\tilde{g}_{IS}(\cdot|\mathbf{Z})||\tilde{g}_x(\cdot|\mathbf{Z}))|\,\bar{L}_N>x\right]\,, \tag{22}$$

where $D(\xi||\eta)$ denotes the divergence of the density ξ from the density η. The first term in Equation (22) is the divergence of f_{IS} from f_x, and is therefore minimised by setting $f_{IS}=f_x$. In other words, the best

[4] In the earliest stages of this project we focused directly on an IS density for $\mathbf{Y}_i$ and had difficulties identifying effective candidates.

possible IS density for the systematic risk factors (according to the criteria of minimum divergence) is the conditional density f_x. The second term in Equation (22) is the average divergence of $\tilde{g}_{IS}(\cdot|\mathbf{z})$ from $\tilde{g}_x(\cdot|\mathbf{z})$, averaged over all possible realisations of the systematic risk factors and conditioned on portfolio loss exceeding the threshold. Based on the developments in Appendix A.5, for fixed $\mathbf{z} \in \mathbb{R}^d$ the divergence of $\tilde{g}_{IS}(\cdot|\mathbf{z})$ from $\tilde{g}_x(\cdot|\mathbf{z})$ is minimised by setting $g_{IS}(\cdot|\mathbf{z}) = g_x(\cdot|\mathbf{z})$. The average divergence in Equation (22) is, therefore, also minimised by setting $g_{IS}(\cdot|\mathbf{z}) = g_x(\cdot|\mathbf{z})$ for every $\mathbf{z} \in \mathbb{R}^d$.

Remark 4. *Among all densities of the form in Equation* (18)*, the one that most resembles the ideal density h_x (in the sense of minimum divergence) is the density:*

$$\hat{h}_x(\mathbf{z}, \boldsymbol{\ell}) := f_x(\mathbf{z}) \cdot \prod_{i=1}^{N} g_x(\ell_i|\mathbf{z}) \,, \qquad \mathbf{z} \in \mathbb{R}^d, \ \boldsymbol{\ell} \in [0, \ell_{\max}]^N \,.$$

In other words, $\hat{h}_x$ is the best possible IS density (among the class Equation (18) *and according to the criteria of minimum divergence) from which to simulate* $(\mathbf{Z}, \mathbf{L})$.

It is worth noting that the IS density $\hat{h}_x$ "gets marginal behaviour correct", in the sense that the marginal distribution of the systematic risk factors, as well as the marginal distribution of an individual loss, is the same under $\hat{h}_x$ as it is under the ideal density h_x. The dependence structure of individual losses is different under $\hat{h}_x$ and h_x—this is the price that we must pay for insisting on conditional independence (i.e., computational efficiency).

3.3. Approximating the Ideal IS Densities

Simulating directly from $\hat{h}_x$ requires an ability to simulate directly from f_x and g_x. Unfortunately, neither f_x nor g_x is numerically tractable (witness the unknown quantities in Equations (15) and (16)), and it does not appear that either is amenable to direct simulation. Our next task is to identify tractable densities that resemble f_x and g_x.

3.3.1. Systematic Risk Factors

As a tractable approximation to f_x, we suggest using that member of the parametric family in Equation (7) that most resembles f_x in the sense of minimum divergence. Using Equations (7) and (15) we get that:

$$\log\left(\frac{f_x(\mathbf{z})}{f_\lambda(\mathbf{z})}\right) = -\lambda^T S(\mathbf{z}) + K(\lambda) + \log\left(\mathbb{P}(\bar{L}_N > x|\mathbf{Z} = \mathbf{z})\right) - \log(p_x) \,,$$

whence the divergence of f_λ from f_x is:

$$D(f_\lambda||f_x) = -\lambda^T \mathbb{E}[S(\mathbf{Z})|\bar{L}_N > x] + K(\lambda) + \mathbb{E}[\log\left(\mathbb{P}(\bar{L}_N > x|\mathbf{Z} = \mathbf{z})\right)|\bar{L}_N > x] - \log(p_x) \,. \tag{23}$$

As a cgf, $K(\cdot)$ is strictly convex. As such, Equation (23) attains its unique minimum at that value of λ such that:

$$\nabla K(\lambda) = \mathbb{E}[S(\mathbf{Z})|\bar{L}_N > x] \,, \tag{24}$$

which, in light of Equation (9), is equivalent to:

$$\mathbb{E}_\lambda[S(\mathbf{Z})] = \mathbb{E}[S(\mathbf{Z})|\bar{L}_N > x] \,. \tag{25}$$

Intuitively, we suggest using that value of the IS parameter λ for which the mean of $S(\mathbf{Z})$ under the IS density matches the conditional mean of $S(\mathbf{Z})$, given that portfolio losses exceed the threshold. In what follows we let $\hat{\lambda}_x$ denote that suggested value of the IS parameter λ, i.e., that value of λ that solves Equation (24).

Remark 5. *The first-stage IS weight associated with the so-described density is:*

$$\Lambda_1(\mathbf{Z}) = \exp(-\hat{\lambda}_x^T S(\mathbf{Z}) + K(\hat{\lambda}_x)) \ . \tag{26}$$

It is entirely possible—and quite common in the examples we consider in this paper—that $K(-\hat{\lambda}_x)$ is not well-defined, in which case Equation (26) has infinite variance under $f_{\hat{\lambda}_x}$ (recall Remark 1). At first glance it might seem absurd to consider IS densities whose associated weights have infinite variance, but as we discuss in Section 4.2 it is straightforward to circumvent this issue by trimming large first-stage IS weights[5].

It remains to develop a tractable approximation to the right hand side of Equation (24), so that we can approximate the value of $\hat{\lambda}_x$. To this end we write the natural sufficient statistic as $S(\mathbf{z}) = (S_1(\mathbf{z}), \ldots, S_p(\mathbf{z}))$ and note that:

$$\mathbb{E}[S_i(\mathbf{Z})|\bar{L}_N > x] = \frac{\mathbb{E}[S_i(\mathbf{Z}) \cdot \mathbf{1}_{\{\bar{L}_N > x\}}]}{\mathbb{P}(\bar{L}_N > x)} = \frac{\mathbb{E}[S_i(\mathbf{Z}) \cdot \mathbb{P}(\bar{L}_N > x|\mathbf{Z})]}{\mathbb{E}[\mathbb{P}(\bar{L}_N > x|\mathbf{Z})]} \ .$$

Next, we use the LDA in Equation (13) to get:

$$\mathbb{E}[S_i(\mathbf{Z})|\bar{L}_N > x] \approx \frac{\mathbb{E}[S_i(\mathbf{Z}) \cdot \exp(-Nq(x, \mathbf{Z}))]}{\mathbb{E}[\exp(-Nq(x, \mathbf{Z}))]} \ . \tag{27}$$

As it only involves the systematic risk factors (and not the large number of idiosyncratic risk factors), the expectation on the right hand side of Equation (27) is amenable to either quadrature or Monte Carlo simulation.

3.3.2. Individual Losses

We encourage the reader unfamiliar with exponential tilts to consult Appendix A.3, before reading the remainder of this section. Our approximation to $g_x(\ell|\mathbf{z})$ is obtained by using the LDA of Equation (13) to approximate both conditional probabilities appearing in Equation (16) (see Appendix A.4 for details). The resulting approximation is:

$$\hat{g}_x(\ell|\mathbf{z}) := \exp(\hat{\theta}\ell - k(\hat{\theta}, \mathbf{z})) \cdot g(\ell|\mathbf{z}) \ , \tag{28}$$

where we recall that $\hat{\theta}$ is defined and discussed in Section 2.3. If the realised values of the systematic risk factors obtained in the first stage lie in the region of interest then $\hat{\theta} = 0$ and $\hat{g}_x$ is identical to g. Otherwise, $\hat{\theta}$ is strictly positive and $\hat{g}_x$ is the exponentially tilted version of g whose mean is x. Intuitively, we can interpret $\hat{g}_x$ as that density that most resembles (in the sense of minimum divergence) g_x, among all densities whose mean is at least x, and the numerical value of $\hat{\theta}$ as the degree to which the density $g(\cdot|\mathbf{z})$ must be deformed, in order to produce a density whose mean is at least x.

Remark 6. *The mean of Equation (28) is* $\max(\mu(\mathbf{z}), x)$*. The implication is that the event of interest is not a rare event under the proposed IS algorithm. Indeed,*

$$\mathbb{E}_{IS}[L_i] = \mathbb{E}_{IS}[\mathbb{E}_{IS}[L_i|\mathbf{Z}]] = \mathbb{E}_{f_{\hat{\lambda}}}[\mathbb{E}_{\hat{g}_x}[L_i|\mathbf{Z}]] = \mathbb{E}_{f_{\hat{\lambda}}}[\max(x, \mu(\mathbf{Z}))] \geq x \ ,$$

which implies that $\lim_{N\to\infty} \mathbb{P}_{IS}(\bar{L}_N > x) = 1$.

---

[5] An alternative to trimming is truncation of large weights; see Ionides (2008) for a general and rigorous treatment of truncated IS.

The second-stage IS weight associated with Equation (28) is:

$$\Lambda_2(\mathbf{Z}, \mathbf{L}) = \prod_{i=1}^{N} \exp(-\hat{\theta} L_i + k(\hat{\theta}, \mathbf{z})) = \exp(-N[\hat{\theta}\bar{L}_N - k(\hat{\theta}, \mathbf{Z})]) \; .$$

Since the second stage weight depends only on $\mathbf{Z}$ and $\bar{L}_N$, we will often write $\Lambda_2(\mathbf{Z}, \bar{L}_N)$ instead of $\Lambda_2(\mathbf{Z}, \mathbf{L})$. In order to assess the stability of the second-stage IS weight, we note that:

$$\exp(-N[\hat{\theta}\bar{L}_N - k(\hat{\theta}, \mathbf{Z})]) = \exp(-\hat{\theta} N[\bar{L}_N - x]) \cdot \exp(-N q(x, \mathbf{Z})) \; .$$

If $\mathbf{Z}$ lies in the region of interest then $\hat{\theta} = q = 0$, whence $\Lambda_2(\mathbf{Z}, \bar{L}_N) = 1$ whatever the value of $\bar{L}_N$. Otherwise, both $\hat{\theta}$ and q are strictly positive, which implies that $\Lambda_2(\mathbf{Z}, \bar{L}_N) < 1$ whenever $\bar{L}_N > x$. The net result of this discussion is that:

$$\Lambda_2(\mathbf{Z}, \bar{L}_N) \leq 1 \quad \text{whenever} \quad \bar{L}_N > x \; . \tag{29}$$

The implication is that large, unstable, IS weights in the second stage will never be a problem.

If the realised value of $\mathbf{z}$ does lie in the region of interest then $\hat{g}_x$ and g are identical, and simulation from g is straightforward. Our final task is to determine how to sample from Equation (28) in the case where $\mathbf{z}$ does not lie in the region of interest. One approach would be to identify a family of densities $\{\eta_{\mathbf{z}}(\mathbf{y}) : \mathbf{z} \in \mathbb{R}^d\}$ such that $L_i = \mathcal{L}(\mathbf{z}, \mathbf{Y}_i)$ is a draw from $\hat{g}_x(\cdot|\mathbf{z})$ whenever $\mathbf{Y}_i$ is a draw from $\eta_{\mathbf{z}}(\cdot)$, but this approach appears to be overly complicated. A simpler approach is to sample from Equation (28) using rejection sampling with g as the proposal density. To this end, we note that for fixed $\mathbf{z}$, the ratio of $\hat{g}_x$ to g is $\exp(\hat{\theta}\ell - k(\hat{\theta}, \mathbf{z}))$, which is bounded and strictly increasing on $[0, \ell_{\max}]$. The best possible (i.e., smallest) rejection constant is therefore:

$$\hat{c} = \hat{c}(x, \mathbf{z}) := \exp(\hat{\theta}\ell_{\max} - k(\hat{\theta}, \mathbf{z})) \; , \tag{30}$$

and the algorithm for sampling from $\hat{g}_x$ would proceed as follows. First, sample $\mathbf{Y}_i$ from its actual density and set $\hat{L}_i = \mathcal{L}(\mathbf{z}, \mathbf{Y}_i)$. Then generate a random number U, uniformly distributed on $[0, 1]$ and independent of $\mathbf{Y}_i$. If,

$$U \leq \frac{\hat{g}_x(\hat{L}_i|\mathbf{z})}{\hat{c} \cdot g(\hat{L}_i|\mathbf{z})} = \exp(-\hat{\theta}(\ell_{\max} - \hat{L}_i)) \; ,$$

set $L_i = \hat{L}_i$ and proceed to the next exposure. Otherwise return to the first step and sample another pair $(\mathbf{Y}_i, U)$.

3.4. Summary and Intuition

The proposed algorithm is summarised in Algorithm 3 below. The initial step is to approximate the value of the first-stage IS parameter, $\hat{\lambda}_x$. In our numerical examples we use a small pilot simulation (10% of the sample size that we eventually use to estimate p_x) and the approximation of Equation (27) in order to estimate $\hat{\lambda}_x$.

Having computed $\hat{\lambda}_x$, the first stage of the algorithm proceeds by simulating independent realisations of the systematic risk factors from the density $f_{\hat{\lambda}_x}$, and computing the associated first-stage weights of Equation (26). Recall that we can interpret these realisations as corresponding to different economic scenarios. Intuitively, sampling from $f_{\hat{\lambda}_x}$ instead of f increases the proportion of adverse scenarios that are generated in the first stage. In the examples we consider, $f_{\hat{\lambda}_x}$ concentrates most of its mass near the boundary of the region of interest, and the effect is to concentrate the distribution of $\mu(\mathbf{Z})$ near x.

In the second stage, one first checks whether or not the realised values of the systematic risk factors lie inside the region of interest. If they do then the event of interest is no longer rare and there is no need to apply further IS in the second stage. Otherwise, if we "miss" the region of interest in the

first stage, we "correct" this mistake by applying an exponential tilt to the conditional distribution of individual losses. Specifically, we transfer mass from the left tail of g to the right tail, in order to produce a density whose mean is exactly x.

Algorithm 3 Proposed IS Algorithm for Estimating p_x

1: Compute $\hat{\lambda}$ using a small pilot simulation.
2: Simulate M i.i.d. copies of the systematic risk factors from $f_{\hat{\lambda}}(\mathbf{z})$ and compute the corresponding first-stage IS weights. Denote the realised values of the factors by $\mathbf{z}_1, \ldots, \mathbf{z}_M$ and the associated IS weights by $\Lambda_1(\mathbf{z}_1), \ldots, \Lambda_1(\mathbf{z}_M)$.
3: For each scenario m, determine whether or not $\mathbf{z}_m$ lies in the region of interest (i.e., whether or not $\mu(\mathbf{z}_m) \geq x$). If it does lie in the region, proceed as follows:
   (a) Simulate the idiosyncratic risk factors for each exposure. Denote the simulated values by $\mathbf{y}_{1,m}, \ldots, \mathbf{y}_{N,m}$.
   (b) Set $\ell_{i,m} = \mathcal{L}(\mathbf{z}_m, \mathbf{y}_{i,m})$, $\bar{\ell}_m = \frac{1}{N}\sum_{i=1}^{N} \ell_{i,m}$ and $\Lambda_2(\mathbf{z}_m, \bar{\ell}_m) = 1$.
   Otherwise, proceed as follows:
   (a) Compute $\hat{\theta} = \hat{\theta}(x, \mathbf{z}_m)$, $\hat{k} = k(\hat{\theta}, \mathbf{z}_m)$ and $\hat{c} = \exp(\hat{\theta}\ell_{\max} - \hat{k})$. For each exposure i:
      (i) Simulate the exposure's idiosyncratic risk factor (denote the realised value by $\hat{\mathbf{y}}_{i,m}$) and set $\hat{\ell}_{i,m} = \mathcal{L}(\mathbf{z}_m, \mathbf{y}_{i,m})$.
      (ii) Simulate a random number drawn uniformly from the unit interval (denote the realised value by u) and determine whether or not $u \leq \exp(-\hat{\theta}(\ell_{\max} - \hat{\ell}_{i,m}))$. If it is, set $\ell_{i,m} = \hat{\ell}_{i,m}$ and proceed to the next exposure. Otherwise, return to step (i).
   (b) Set $\bar{\ell}_m = \frac{1}{N}\sum_{i=1}^{N} \ell_{i,m}$ and $\Lambda_2(\mathbf{z}_m, \bar{\ell}_m) = \exp(-N[\hat{\theta}\bar{\ell}_m - \hat{k}])$
4: Return $\widehat{p}_x = \frac{1}{M}\sum_{m=1}^{M} \Lambda_1(\mathbf{z}_m) \cdot \Lambda_2(\mathbf{z}_m, \bar{\ell}_m) \cdot \mathbf{1}_{\{\bar{\ell}_m > x\}}$.

4. Practical Considerations

In this section we discuss some of the practical issues that arise when implementing the proposed methodology.

4.1. One- and Two-Stage Estimators

The rejection sampling procedure employed in the second stage of the proposed algorithm involves repeated evaluation of $\hat{\theta}$, which requires a non-trivial amount of computational time time. In addition, rejection sampling in general requires relatively complicated code. As such, it is worth considering a simpler algorithm that only applies importance sampling in the first stage, and is therefore easier to implement and faster to run.

In what follows we will distinguish between one- and two-stage IS algorithms. A one-stage algorithm only applies IS in the first stage and samples $(\mathbf{Z}, \mathbf{L})$ from the IS density:

$$h_{1S}(\mathbf{z}, \boldsymbol{\ell}) := f_{\hat{\lambda}_x}(\mathbf{z}) \cdot \prod_{i=1}^{N} g(\ell_i|\mathbf{z}) \,. \tag{31}$$

The associated IS weight is $\Lambda_1(\mathbf{z})$ and the one-stage algorithm is summarised in Algorithm 4 below. Note the simplicity of Algorithm 4, relative to Algorithm 3. The two-stage algorithm applies IS in both the first stage and the second stage, sampling $(\mathbf{Z}, \boldsymbol{L})$ from the IS density:

$$h_{2S}(\mathbf{z}, \boldsymbol{\ell}) := f_{\hat{\lambda}_x}(\mathbf{z}) \cdot \prod_{i=1}^{N} \hat{g}_x(\ell_i|\mathbf{z}) \,. \tag{32}$$

The associated IS weight is $\Lambda_1(\mathbf{z}) \cdot \Lambda_2(\mathbf{z}, \bar{\ell}_N)$, and the two-stage algorithm was summarised previously in Algorithm 3.

Algorithm 4 Proposed One-Stage IS Algorithm for Estimating p_x

1: Compute $\hat{\lambda}_x$ using a small pilot simulation.
2: Simulate M i.i.d. copies of the systematic risk factors from $f_{\hat{\lambda}}(\mathbf{z})$ and compute the corresponding first-stage IS weights. Denote the realised values of the factors by $\mathbf{z}_1, \ldots, \mathbf{z}_M$ and the associated IS weights by $\Lambda_1(\mathbf{z}_1), \ldots, \Lambda_1(\mathbf{z}_M)$.
3: For each scenario m:
   (a) Simulate the idiosyncratic risk factors for each exposure. Denote the simulated values by $\mathbf{y}_{1,m}, \ldots, \mathbf{y}_{N,m}$.
   (b) Set $\ell_{i,m} = \mathcal{L}(\mathbf{z}_m, \mathbf{y}_{i,m})$ and $\bar{\ell}_m = \frac{1}{N}\sum_{i=1}^{N} \ell_{i,m}$.
4: Return $\widehat{p}_x = \frac{1}{M}\sum_{m=1}^{M} \Lambda_1(\mathbf{z}_m) \cdot \mathbf{1}_{\{\bar{\ell}_m > x\}}$.

Although it is simpler to implement and faster to run, the one-stage algorithm is less accurate than the two-stage algorithm. More precisely, the two-stage estimator never has larger variance than the one-stage estimator. To see this, first let $\mathbb{E}_{1S}$ denote expectation under the one-stage IS density $h_{1S}(\mathbf{z}, \boldsymbol{\ell})$ given in Equation (31). Then the variance of the one-stage estimator is:

$$\frac{\mathbb{E}_{1S}[(\Lambda_1(\mathbf{Z}) \cdot \mathbf{1}_{\{\bar{L}_N \geq x\}})^2] - p_x^2}{M},$$

where M denotes sample size. And if we let $\mathbb{E}_{2S}$ denote expectation under the two-stage IS density $h_{2S}(\mathbf{z}, \boldsymbol{\ell})$ given in Equation (32) then the variance of the two-stage estimator is:

$$\frac{\mathbb{E}_{2S}[(\Lambda_1(\mathbf{Z}) \cdot \Lambda_2(\mathbf{Z}, \bar{L}_N) \cdot \mathbf{1}_{\{\bar{L}_N \geq x\}})^2] - p_x^2}{M}.$$

In order to compare variances it suffices to compare the second moments appearing above under the actual density $h(\mathbf{z}, \boldsymbol{\ell})$, and we let $\mathbb{E}$ denote expectation with respect to this density. To this end we note that:

$$\mathbb{E}_{1S}[(\Lambda_1(\mathbf{Z}) \cdot \mathbf{1}_{\{\bar{L}_N \geq x\}})^2] = \mathbb{E}[\Lambda_1(\mathbf{Z}) \cdot \mathbf{1}_{\{\bar{L}_N \geq x\}}]$$

and

$$\mathbb{E}_{2S}[(\Lambda_1(\mathbf{Z}) \cdot \Lambda_2(\mathbf{Z}, \bar{L}_N) \cdot \mathbf{1}_{\{\bar{L}_N \geq x\}})^2] = \mathbb{E}[\Lambda_1(\mathbf{Z}) \cdot \Lambda_2(\mathbf{Z}, \bar{L}_N)] \cdot \mathbf{1}_{\{\bar{L}_N \geq x\}}]$$

In light of Equation (29) we get that:

$$\Lambda_2(\mathbf{Z}, \bar{L}_N) \cdot \mathbf{1}_{\{\bar{L}_N > x\}} \leq 1 \cdot \mathbf{1}_{\{\bar{L}_N > x\}} = \mathbf{1}_{\{\bar{L}_N > x\}}, \tag{33}$$

whence

$$\begin{aligned} \mathbb{E}_{2S}[(\Lambda_1(\mathbf{Z}) \cdot \Lambda_2(\mathbf{Z}, \bar{L}_N) \cdot \mathbf{1}_{\{\bar{L}_N \geq x\}})^2] &= \mathbb{E}[\Lambda_1(\mathbf{Z}) \cdot \Lambda_2(\mathbf{Z}, \bar{L}_N) \cdot \mathbf{1}_{\{\bar{L}_N \geq x\}}] \\ &\leq \mathbb{E}[\Lambda_1(\mathbf{Z}) \cdot \mathbf{1}_{\{\bar{L}_N \geq x\}}] \\ &= \mathbb{E}_{1S}[(\Lambda_1(\mathbf{Z}) \cdot \mathbf{1}_{\{\bar{L}_N \geq x\}})^2]. \end{aligned}$$

The two-stage estimator will therefore never have larger variance than the the one-stage estimator.

4.2. Large First-Stage Weights

In the examples that we consider in this paper, the systematic risk factors are Gaussian. When selecting their IS density, one could either (i) shift their means and leave their variances (and correlations) unchanged or (ii) shift their means and adjust their variances (and correlations). In general

the latter approach will lead to a much better approximation to the ideal density f_x, but could lead to an IS weight that has infinite variance. By contrast, the former approach will always lead to an IS weight with finite variance, but could lead to a poor approximation of the ideal density. At first glance it might seem absurd to consider IS densities whose weights are so unstable as to have infinite variance, but we have found that adjusting the variances of the systematic risk factors can lead to more effective estimators, in terms of both statistical accuracy and run time (see Section 6.1 for more details), provided one stabilises the resulting IS weights in some way. In the remainder of this section we describe a simple stabilisation technique that leads to a computable upper bound on the associated bias (an alternative would be to stabilize unruly IS weights via truncation, as discussed in Ionides (2008)).

Returning now to the general case, suppose that the first-stage IS parameter, $\hat{\lambda}_x$, is such that the first-stage IS weight, $\Lambda_1(\mathbf{Z})$, has infinite variance. We trim large first-stage weights by fixing a set $A \subset \mathbb{R}^d$ such that $\Lambda_1(\cdot)$ is bounded over A, and discarding those simulations for which $\mathbf{Z} \notin A$. Specifically, the last line of Algorithm 3 would be altered to return the trimmed estimate:

$$\widehat{p}_x = \frac{1}{M} \sum_{m=1}^{M} \Lambda_1(\mathbf{z}_m) \cdot \Lambda_2(\mathbf{z}, \bar{\ell}_m) \cdot \mathbf{1}_{\{\bar{\ell}_m > x\}} \cdot \mathbf{1}_{\{\mathbf{z}_m \in A\}} ,$$

etc. for Algorithm 4. The variance of the so-trimmed estimator is necessarily finite (recall that $\Lambda_2(\mathbf{z}, \bar{\ell}) \leq 1$ if $\bar{\ell} > x$), and its bias is:

$$\mathbb{E}_{2S}[\Lambda_1(\mathbf{Z}) \cdot \Lambda_2(\mathbf{Z}, \bar{L}_N) \cdot \mathbf{1}_{\{\bar{L}_N > x\}} \cdot \mathbf{1}_{\{\mathbf{Z} \notin A\}}] = \mathbb{E}[\mathbf{1}_{\{\bar{L}_N > x\}} \cdot \mathbf{1}_{\{\mathbf{Z} \notin A\}}] = \mathbb{E}[\mathbb{P}(\bar{L}_N > x | \mathbf{Z}) \cdot \mathbf{1}_{\{\mathbf{Z} \notin A\}}] ,$$

where we have used the tower property (conditioning on $\mathbf{Z}$) to obtain the last equality. Using Chernoff's bound in Equation (11) we get that:

$$\mathbb{E}[\mathbb{P}(\bar{L}_N > x | \mathbf{Z}) \cdot \mathbf{1}_{\{\mathbf{Z} \notin A\}}] \leq \mathbb{E}[\exp(-Nq(x, \mathbf{Z})) \cdot \mathbf{1}_{\{\mathbf{Z} \notin A\}}] . \tag{34}$$

As it only depends on the small number of systematic risk factors, and not the large number of idiosyncratic risk factors, the right-hand side of Equation (34) is a tractable upper bound on the bias committed by trimming large (first-stage) IS weights. This upper bound can be used to assess whether or not the bias associated with a given set A is acceptable.

4.3. Large Rejection Constants

The smaller the $\hat{c}$, the more efficient is the rejection sampling algorithm employed in the second stage. Indeed the average number of proposals that must be generated in order to obtain one realisation from $\hat{g}_x$ is $1/\hat{c}$. In the examples we consider in this paper, $\hat{c}$ is (essentially) a decreasing function $\mu(\mathbf{z})$, such that $\hat{c} \to 1$ as $\mu(\mathbf{z}) \to x$ and $\hat{c} \to \infty$ as $\mu(\mathbf{z}) \to 0$ (see Figure 1). The second-stage rejection algorithm is therefore quite efficient when $\mu(\mathbf{z}) \approx x$ and quite inefficient when $\mu(\mathbf{z}) \approx 0$. Now, the IS density for the first-stage risk factors is such that the distribution of $\mu(\mathbf{Z})$ concentrates most of its mass near x (where $\hat{c}$ is a reasonable size), but it is still theoretically possible to obtain a realisation of the systematic risk factors for which $\mu(\mathbf{z})$ is very small and $\hat{c}$ is unacceptably large (e.g., 10^4). In such situations the algorithm effectively grinds to a halt, as one endlessly generates proposed losses that have no realistic chance of being accepted. It is extremely unlikely that one does obtain such a scenario under the first-stage IS distribution, but it is still important to protect oneself against this unlikely event. To this end we suggest fixing some maximum acceptable rejection constant c_{max}, and only applying the second stage IS to those first-stage realizations for which $\mu(\mathbf{z}) < x$ and $\hat{c} \leq c_{max}$. In other words, even if the realised values of the systematic risk factors lie outside the region of interest, we avoid applying the second stage if the associated rejection constant exceeds the predefined threshold.

4.4. Computing $\hat{\theta}$

Repeated evaluations of $\hat{\theta}(x,\cdot)$ are necessary when computing $\hat{\lambda}_x$ at the outset of the algorithm, as well as during the second stage of the two-stage algorithm. Recall that in order to compute $\hat{\theta}(x,\mathbf{z})$ "exactly" one must numerically solve the equation $k'(\theta,\mathbf{z}) = x$, which requires a non-trivial amount of CPU time. As each evaluation of $\hat{\theta}$ is relatively costly, repeated evaluation would, in the absence of any further approximation (over and above that inherent in numerical root-finding), account for the vast majority of the algorithm's total run time.

In order to reduce the amount of time spent evaluating $\hat{\theta}$ we fit a low degree polynomial to the function $\hat{\theta}(x,\cdot)$ that can be evaluated extremely quickly, considerably reducing total run time. Specifically, suppose that we must compute $\hat{\theta}(x,\mathbf{z}_n)$ for each of n points $\mathbf{z}_1,\ldots,\mathbf{z}_n$ (either the sample points from the pilot simulation, or the first-stage realisations that did not land in the region of interest). We identify a small set $C \subset \mathbb{R}^d$ that contains each of the n points, construct a mesh of $m << n$ points in C, evaluate $\hat{\theta}$ exactly at each mesh point, and then fit a fifth degree polynomial to the resulting data. Letting $\bar{\theta}(x,\cdot)$ denote the resulting polynomial, we then evaluate $\bar{\theta}(x,\mathbf{z}_1),\ldots,\bar{\theta}(x,\mathbf{z}_n)$ instead of $\hat{\theta}(x,\mathbf{z}_1),\ldots,\hat{\theta}(x,\mathbf{z}_n)$. If m is substantially smaller than n, then the reduction in CPU time is considerable.

5. PD-LGD Correlation Framework

All of the PD-LGD correlation models listed in the introduction are special cases of the following general framework—an observation that, to the best of our knowledge, has not been made in the literature. The systematic risk factors take the form $\mathbf{Z} = (Z_D, Z_L)$, where Z_D and Z_L are bivariate normal with standard normal margins and correlation ρ_S. Idiosyncratic risk factors take the form $\mathbf{Y}_i = (Y_{i,D}, Y_{i,L})$, where $Y_{i,D}$ and $Y_{i,L}$ are bivariate normal with standard normal margins and correlation ρ_I.

Associated with each exposure is a default driver $X_{i,D}$ and a loss driver $X_{i,L}$, defined as follows:

$$X_{i,D} = \alpha_D Z_D + \sqrt{1-\alpha_D^2}\,Y_{i,D}\ , \tag{35}$$

$$X_{i,L} = \alpha_L Z_L + \sqrt{1-\alpha_L^2}\,Y_{i,L}\ . \tag{36}$$

The factor loadings α_D and α_L are constants taking values in the unit interval, and dictate the relative importance of systematic risk versus idiosyncratic risk. The correlation between default drivers of distinct exposures is $\rho_D := \alpha_D^2$ and the correlation between loss drivers of distinct exposures is $\rho_L := \alpha_L^2$. The correlation between the default and potential loss drivers of a particular exposure is:

$$\rho_{DL} := \alpha_D\alpha_L\rho_S + \sqrt{1-\alpha_D^2}\sqrt{1-\alpha_L^2}\rho_I\ ,$$

which can be positive or negative (or zero). Note that if ρ_S and ρ_I have the same sign then, since both factor loadings are positive, ρ_{DL} inherits this common sign.

The realised loss on exposure i is $L_i = \mathcal{D}_i \cdot \mathcal{L}_i$, where:

$$\mathcal{D}_i = \mathbf{1}_{\{X_{i,D} \le \Phi^{-1}(P)\}}$$

is the default indicator associated with exposure i and

$$\mathcal{L}_i = h(X_{i,L})$$

is called the potential loss (our terminology) associated with exposure i. Here P denotes the common default probability of all exposures and h is some function from $\mathbb{R}$ to $[0,\ell_{\max}]$. It is useful (but not necessary) to think of potential loss as $\mathcal{L}_i = \max(0, 1-\mathcal{C}_i)$, where $\mathcal{C}_i$ is the value of the collateral pledged to exposure i expressed as a fraction of the loan's notional value.

Models in this framework are characterised by (i) the correlation structure of the risk factors, specifically restrictions on the values of ρ_I and ρ_S, and (ii) the marginal distribution of potential loss. For instance:

- Frye (2000) assumes perfect systematic correlation ($\rho_S = 1$) and zero idiosyncratic correlation ($\rho_I = 0$);
- Pykhtin (2003) assumes perfect systematic correlation ($\rho_S = 1$) but allows for arbitrary idiosyncratic correlation (ρ_I unrestricted);
- Witzany (2011) allows for arbitrary systematic correlation (ρ_S unrestricted) but insists on zero idiosyncratic correlation ($\rho_I = 0$);
- Miu and Ozdemir (2006) allow for arbitrary systematic correlation (ρ_S unrestricted) and arbitrary idiosyncratic correlation (ρ_I unrestricted).

Note that if $\rho_S = \pm 1$ then the systematic risk factor is effectively one-dimensional. Indeed if $\rho_S = 1$ then $\mathbf{Z} = (Z, Z)$ from some standard Gaussian variable Z, and if $\rho_S = -1$ then $\mathbf{Z} = (Z, -Z)$. We refer to the case $|\rho_S| = 1$ as the one-factor case, and the case $|\rho_S| < 1$ as the two-factor case. In the one-factor case we use Z, and not $\mathbf{Z}$, to denote the systematic risk factor. The first two models listed above are one-factor models, the last two are two-factor models.

The marginal distribution of potential loss is determined by the specification of the function h. For instance:

- Frye (2000) specifies $h(x) = \max(0, 1 - a(1 + bx))$ for constants $a \in \mathbb{R}$ and $b > 0$. Potential loss takes values in $[0, \infty)$. Its density has a point mass at zero and is proportional to a Gaussian density on $(0, \infty)$. Since L_i is not constrained to lie in the unit interval, this specification violates the assumptions made in Section 2.3;
- Pykhtin (2003) specifies $h(x) = \max(0, 1 - e^{a+bx})$ for constants $a \in \mathbb{R}$ and $b > 0$. Potential loss takes values in $[0, 1)$. Its density has a point mass at zero, and is proportional to a shifted lognormal density over $(0, 1)$;
- Witzany (2011) and Miu and Ozdemir (2006) both specify $h(x) = B_{a,b}^{-1}(\Phi(x))$, where $a, b > 0$ and $B_{a,b}$ denotes the cdf of the beta distribution with parameters a and b. Potential loss takes values in $(0, 1)$. It is a continuous variable and follows a beta distribution.

The sign of ρ_{DL} and the nature of the function h (increasing or decreasing) will in general determine the sign of the relationship between $\mathcal{D}_i$ and $\mathcal{L}_i$. If $\rho_{DL} > 0$ then the relationship will be positive [negative] provided h is decreasing [increasing], and vice versa if $\rho_{DL} < 0$.

5.1. Computing $\mu(\mathbf{z})$

Here vectors $\mathbf{z} \in \mathbb{R}^2$ take the form $\mathbf{z} = (z_D, z_L)^T$. In order to obtain an expression for $\mu(\mathbf{z}) = \mathbb{E}[L_i | \mathbf{Z} = \mathbf{z}]$, we begin with the observation that:

$$\mathbb{E}[L_i|\mathbf{Z}] = \mathbb{E}[\mathcal{L}_i\mathcal{D}_i|\mathbf{Z}] = \mathbb{E}[\mathcal{L}_i\mathbb{E}[\mathcal{D}_i|\mathbf{X}_{i,L}, \mathbf{Z}]|\mathbf{Z}] = \mathbb{E}[\mathcal{L}_i\mathbb{P}(\mathcal{D}_i = 1|X_{i,L}, \mathbf{Z})|\mathbf{Z}] \, .$$

Thus,

$$\mu(\mathbf{z}) = \int_{\mathbb{R}} h(x_L) \cdot \Phi(d, m(x_L, \mathbf{z}), v) \cdot \phi(x_L, \alpha_L z_L, 1 - \alpha_L^2) \, dx_L \, , \tag{37}$$

where

$$m(x_L, \mathbf{z}) := \alpha_D z_D + \rho_I \cdot \sqrt{\frac{1 - \alpha_D^2}{1 - \alpha_L^2}} \cdot (x_L - \alpha_L z_L)$$

and

$$v = v(x_L, \mathbf{z}) := (1 - \alpha_D^2)(1 - \rho_I^2)$$

are the conditional mean and variance of $X_{i,D}$, respectively, given that $(X_{i,L}, \mathbf{Z}) = (x_L, \mathbf{z})$. In general $\mu(\mathbf{z})$ must be evaluated using quadrature, and doing so is straightforward[6]. On average (across parameter values and points $\mathbf{z} \in \mathbb{R}^2$) a single evaluation of $\mu(\cdot)$ requires approximately one millisecond. In the one-factor case with $\rho_S = 1$ [$\rho_S = -1$] the expression for $\mu(z) = \mathbb{E}[L_i|Z = z]$ is obtained by plugging $\mathbf{z} = (z, z)$ [$\mathbf{z} = (z, -z)$] into Equation (37).

5.2. Computing $k(\theta, \mathbf{z})$ and $\hat{\theta}(x, \mathbf{z})$

Here again, vectors $\mathbf{z} \in \mathbb{R}^2$ take the form $\mathbf{z} = (z_D, z_L)^T$. In order to derive an expression for $k(\theta, \mathbf{z})$ we begin with the observation that:

$$e^{\theta L_i} = \mathbf{1}(\mathcal{D}_i = 0) + e^{\theta \mathcal{L}_i}\mathbf{1}(\mathcal{D}_i > 0) = 1 + (e^{\theta \mathcal{L}_i} - 1) \cdot \mathbf{1}(\mathcal{D}_i > 0) \,,$$

and since $k(\theta, \mathbf{z}) = \log(\mathbb{E}[e^{\theta L_i}|\mathbf{Z} = \mathbf{z}])$, we get that:

$$k(\theta, \mathbf{z}) = \log\left(1 + \int_{\mathbb{R}} (e^{\theta h(x_L)} - 1) \cdot \Phi(d, m(x_L, \mathbf{z}), v) \cdot \phi(x_L, \alpha_L z_L, 1 - \alpha_L^2)\, dx_L\right) \,, \tag{38}$$

where $m(x_L, \mathbf{z})$ and v are given in the previous section. In the one-factor case with $\rho_S = 1$ [$\rho_S = -1$] the expression for $k(\theta, z) = \log(\mathbb{E}[\exp(\theta L_i)|Z = z])$ is obtained by plugging $\mathbf{z} = (z, z)$ [$\mathbf{z} = (z, -z)$] into Equation (38). As with $\mu(\mathbf{z})$, $k(\theta, \mathbf{z})$ must in general be evaluated using quadrature, which is straightforward. The time required for a single evaluation of $k(\theta, \cdot)$ is comparable to that required for a single evaluation of $\mu(\cdot)$.

In order to compute $\hat{\theta}$ we must solve the equation $k'(\theta, \mathbf{z}) = x$ with respect to θ. Differentiating Equation (38) we get:

$$k'(\theta, \mathbf{z}) = \frac{\partial k(\theta, \mathbf{z})}{\partial \theta} = \frac{\int_{\mathbb{R}} h(x_L) \cdot e^{\theta h(x_L)} \cdot \Phi(d, m(x_L, \mathbf{z}), v) \cdot \phi(x_L, \alpha_L z_L, 1 - \alpha_L^2)\, dx_L}{\exp(k(\theta, \mathbf{z}))} \,, \tag{39}$$

which is straightforward to compute using quadrature. A single evaluation of $k'(\theta, \mathbf{z})$ requires approximately twice as much time as a single evaluation of $k(\theta, \mathbf{z})$. As the root of $k'(\theta, \mathbf{z}) = x$ must be evaluated numerically, evaluating $\hat{\theta}$ is much more time consuming than evaluating k or k'. Across parameter values and points $\mathbf{z} \in \mathbb{R}^2$, and using $\theta = 0$ as an initial guess, the average time required for a single evaluation[7] of $\hat{\theta}(x, \cdot)$ is slightly less than one tenth of one second.

The right panel of Figure 1 illustrates the relationship between expected losses and the rejection constant employed in the second stage, $\hat{c} = \exp(\hat{\theta} - k(\hat{\theta}, \mathbf{z}))$. We see that $\hat{c}$ is essentially a decreasing function of $\mu(\mathbf{z})$, such that $\hat{c} \to 1$ as $\mu(\mathbf{z}) \to x$ and $\hat{c} \to \infty$ as $\mu(\mathbf{z}) \to 0$. The left panel of Figure 1 illustrates the graph of the LDA approximation $\mathbb{P}(\bar{L}_N > x|\mathbf{Z} = \mathbf{z}) \approx \exp(-Nq(x, \mathbf{z}))$. The approximation is identically equal to one inside the region of interest, and decays to zero very rapidly outside the region. In other words, most of the variability in the function $q(x, \cdot)$ occurs along, and just outside, the boundary of the region of interest.

---

[6] All calculations are carried out using Matlab 2018a on a 2015 MacBook Pro with 6.8 GHz Intel Core i7 processor and 16 GB (1600 MHz) of memory. Numerical integration is performed using the built-in `integral` function.

[7] We use the Matlab function `fzero` for the root-finding.

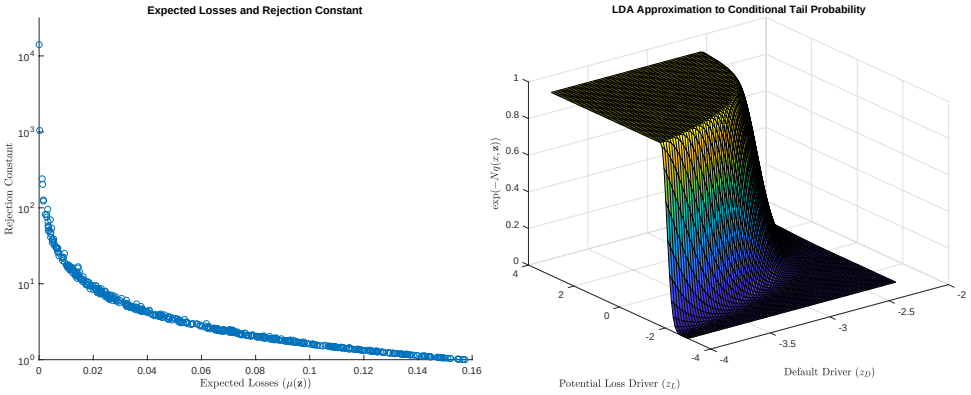

Figure 1. The left panel of this figure illustrates the relationship between expected losses $\mu(\mathbf{z})$ and the second-stage rejection constant $\hat{c} = \hat{c}(x, \mathbf{z})$, in the two-factor model. The right panel illustrates the graph of the LDA approximation of Equation (13). Parameters (randomly selected using the procedure in Section 5.3) in both panels are $(P, \rho_D, \rho_L, \rho_I, \rho_S, a, b, N)$ = (0.0063, 0.3964, 0.2794, −0.3356, −0.7599, 0.6497, 0.5033, 134) and the threshold is x = 0.1575. Mean losses are $\mathbb{E}[L_i]$ = 0.0029, and the probability that losses exceed the threshold x is on the order of 50 basis points. Points in the left panel were obtained by generating 1000 realizations of the systematic risk factors from their actual distribution (as opposed to the first-stage IS distribution) using the indicated parameter values.

5.3. Exploring the Parameter Space

The model contains five parameters, in addition to any parameters associated with the transformation h. We are ultimately interested in how well the proposed algorithms perform across a wide range of different parameter sets. As such, in our numerical experiments we will randomly select a large number of parameter sets according to the procedure described below, and assess the algorithms' performance for each parameter set.

- Generate the default probability P uniformly between 0% and 10%, and generate each of the correlations $\rho_D = \alpha_D^2$ and $\rho_L = \alpha_L^2$ uniformly between 0% and 50%;
- In the one-factor model, generate ρ_S uniformly on $\{-1, 1\}$, i.e., ρ_S takes on the value -1 or $+1$ with equal probability. If $\rho_S = 1$ we generate ρ_I uniformly between 0% and 100%, and if $\rho_S = -1$ we generate ρ_I uniformly between -100% and 0%. This allows us to control the sign of ρ_{DL}, which we must do in order to ensure a positive relationship between default and potential loss. In the two-factor model we randomly generated ρ_S uniformly on $[-1, 1]$. If ρ_S is positive, randomly generate ρ_I uniformly on $[0, 1]$, otherwise randomly generate ρ_I uniformly on $[-1, 0]$;
- We choose the transformation $h(\cdot)$ to ensure that (i) potential loss is beta distributed and (ii) there is a positive relationship between default and loss. The paramters a and b of the beta distribution are generated independently from an exponential distribution with unit mean. If $\rho_{DL} < 0$ we set $h(x) = B_{a,b}^{-1}(\Phi(x))$ and if $\rho_{DL} > 0$ we set $h(x) = B_{a,b}^{-1}(\Phi(-x))$, where $B_{a,b}(\cdot)$ is the cumulative distribution function for the beta distribution with parameters a and b. Note that under these restrictions, in the one-factor model the expected loss function $\mu(z)$ is monotone decreasing.

In order to ensure that we are considering cases of practical interest, we randomise the portfolio size and loss threshold as follows.

- Generate the number of exposures randomly between 10 and 5000;
- In the one-factor model we generate the threshold x by setting $x = \mu(\Phi^{-1}(10^{-q}))$, where q is uniformly distributed on $[1, 5]$. The LPA suggests that

$$p_x = \mathbb{P}(\bar{L}_N > x) \approx \mathbb{P}(\mu(Z) > x) = \mathbb{P}(Z < \mu^{-1}(x)) = 10^{-q}\,.$$

This means that $\log(p_x)$, the order of magnitude of the probability of interested, is approximately uniformly distributed on $[-5,-1]$. In the two-factor model we set $x = \mu(\mathbf{z}_q)$, where $\mathbf{z}_q = (\Phi^{-1}(q), \rho_S\Phi^{-1}(q))$ and q is uniformly distributed on $[-5,-1]$.

6. Implementation

In this section we discuss our implementation of the algorithm proposed in Section 3 in the general framework outlined in Section 5. As the general framework encompasses many of the PD-LGD correlations that have been proposed in the literature, this section effectively discusses implementation of the proposed algorithm across a wide variety of models that are used in practice.

6.1. Selecting the IS Density for the Systematic Risk Factors

The systematic risk factors here are Gaussian. When constructing their IS density we could either shift their means and leave their variances (and correlations) unchanged, or shift their means and adjust their variances (and correlations). Recall that the ultimate goal is to choose an IS density that closely resembles the ideal density f_x given in Equation (15). As illustrated[8] in Figure 2, the ideal density f_x tends to be very tightly concentrated about its mean, and adjusting the variance of the systematic risk factors leads to a much better approximation to the ideal density for "typical values" of the ideal density. The left tail of the ideal density is, however, heavier than the variance-adjusted IS density, an issue that can be resolved by trimming large IS weights.

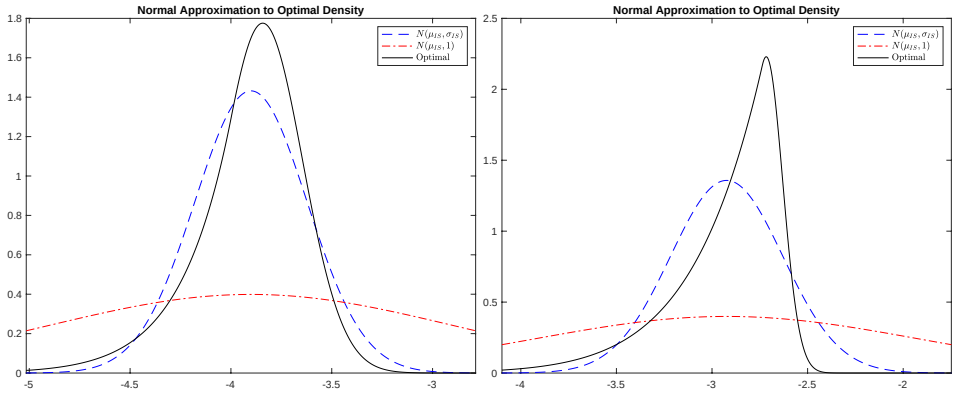

Figure 2. This figure illustrates f_x (in fact, the approximation of Equation (40)) for two randomly generated sets of parameters. Each panel superimposes (i) a normal density with the same mean and variance as f_x (dashed blue line), and (ii) a normal density with the same mean as f_x and unit variance (dash-dot red line). The mean and variance of f_x are computed via (computationally inefficient) quadrature. The mean and variance of f_x are computed using quadrature. Parameters in the right panel are $(P, \rho_D, \rho_L, \rho_I, \rho_S, a, b, N)$ = (0.02, 0.33, 0.27, 0.96, 1, 2.47, 4.32, 454), and for the left panel they are $(P, \rho_D, \rho_L, \rho_I, \rho_S, a, b, N)$ = (0.03, 0.13, 0.12, 0.85, 1, 1.81, 1.90, 271). In both cases, the transformation h is taken to be $h(x) = B^{-1}_{a,b}(\Phi(x))$.

The downside to adjusting the variance of the systematic risk factors is that it can lead to first-stage IS weights with infinite variance, but numerical evidence suggests that this issue can be mitigated by

[8] In the one-factor model, a tractable approximation to the ideal density can be obtained by using the LDA of Equation (13) to approximate both probabilities appearing in Equation (15). The result is:

$$f_x(z) \approx \frac{\exp(-Nq(x,z)) \cdot \phi(z)}{\int_{\mathbb{R}} \exp(-Nq(x,w)) \cdot \phi(w)\, dw}, \tag{40}$$

and the right-hand side of Equation (40) can be approximated via quadrature. As the integrand involves $\hat{\theta}$, the approximation is computationally very slow.

trimming large weights. Indeed, numerical experiments[9] suggest that adjusting variance and trimming large weights leads to substantially more accurate estimators of p_x. Intuitively, it is more important for the IS density to mimic the behaviour of the ideal density over its "typical range", as opposed to faithfully representing its tail behaviour. In addition to improving statistical accuracy, adjusting variance has the added benefit of making the second stage of the algorithm more computationally efficient in terms of run time. Indeed, as discussed in more detail in Section 6.3, adjusting variance tends to increase the proportion of first-stage simulations that land in the region of interest (thereby reducing the number of times the rejection sampling algorithm must be employed in the second stage) and reduces the average size of the rejection constants employed in the second stage (thereby making the rejection algorithm more effective whenever it must be employed).

6.2. First Stage

In this section we explain how to efficiently approximate the parameters of the optimal IS density for the systematic risk factors, in both the one- and two-factor models. We also explain how we trim large IS weights, and demonstrate that the resulting bias is negligible.

6.2.1. Computing Parameters in the Two-Factor Model

In the two-factor model the systematic risk factors are bivariate Gaussian with zero mean vector and covariance matrix:

$$\Sigma = \begin{bmatrix} 1 & \rho_S \\ \rho_S & 1 \end{bmatrix} .$$

The mean vector and covariance matrix that satisfy the criteria of Equation (25) are:[10]

$$\mu_{IS} := \mathbb{E}[\mathbf{Z}|\bar{L}_N > x] \tag{41}$$

and

$$\Sigma_{IS} := \mathbb{E}[(\mathbf{Z} - \mu_{IS})(\mathbf{Z} - \mu_{IS})^T|\bar{L}_N > x] , \tag{42}$$

respectively. In order to approximate the suggested mean vector and covariance matrix we use Equation (27) to get:

$$\mu_{IS} \approx \frac{\mathbb{E}[\exp(-Nq(x,\mathbf{Z})) \cdot \mathbf{Z}]}{\mathbb{E}[\exp(-Nq(x,\mathbf{Z}))]} \tag{43}$$

and

$$\Sigma_{IS} \approx \frac{\mathbb{E}[\exp(-Nq(x,\mathbf{Z})) \cdot (\mathbf{Z} - \mu_{IS})(\mathbf{Z} - \mu_{IS})^T]}{\mathbb{E}[\exp(-Nq(x,\mathbf{Z}))]} . \tag{44}$$

The expected values appearing on the right-hand sides of Equations (43) and (44) are both amenable to simulation, and we use a small pilot simulation of size $M_p << M$ to approximate them. In our numerical examples, the size of the pilot simulation is 10% of the sample size that is eventually used to estimate p_x.

---

[9] Whether or not we adjust the variance of the systematic risk factor, the standard error of the resulting estimator is of the form $\nu/\sqrt{M}$, where ν depends on the model parameters and is easily estimated via simulation. Using 100 randomly selected parameter sets from the one-factor model, selected according to the procedure described in Section 5.3, we find that for the one-stage estimator $\nu_{MS}/\nu_{VA} \approx 1.54 p_x^{-0.03}$, where ν_{MS} denotes the value of ν assuming we only shift the mean of the systematic risk factor and do not adjust its variance and ν_{VA} denotes the value when we do adjust variance. For probabilities in the range of interest, then, adjusting the variance of the systematic risk factor leads to an estimator that is nearly four times as efficient, in the sense that the sample size required to achieve a given degree of accuracy (as measured by standard error) is nearly four times larger if we do not adjust variance.

[10] As discussed in Appendix B, the natural sufficient statistic here consists of the components of $\mathbf{Z}$ plus the components of $\mathbf{Z}\mathbf{Z}^T$. As such, in order to satisfy Equation (27) we must ensure that $\mathbb{E}_{IS}[\mathbf{Z}] = \mathbb{E}[\mathbf{Z}|\bar{L}_N > x]$ and $\mathbb{E}_{IS}[\mathbf{Z}\mathbf{Z}^T] = \mathbb{E}[\mathbf{Z}\mathbf{Z}^T|\bar{L}_N > x]$, where $\mathbb{E}_{IS}$ denotes mean under the IS distribution. These conditions are clearly equivalent to Equations (41) and (42)..

In order to implement the approximation we must first simulate the systematic risk factors and then compute $q(x, \mathbf{z})$ for each sample point $\mathbf{z}$. The most natural way to proceed is to (i) sample the systematic risk factors from their actual distribution (bivariate Gaussian with zero mean vector and covariance matrix Σ) and (ii) numerically solve the equation $k'(\theta, \mathbf{z}) = x$ in order to compute compute $\hat{\theta}(x, \mathbf{z})$ for each pilot sample point $\mathbf{z}$ that lies outside the region of interest. In our experience this leads to unacceptably inefficient estimators, in terms of both (i) statistical accuracy and (ii) computational time. We deal with each issue in turn.

As most of the variation in $q(x, \cdot)$ occurs just outside the boundary of the region of interest (recall the right panel of Figure 1), we suggest using an IS distribution for the pilot simulation that is centered on the boundary of the region. Specifically, we suggest using that point on the boundary at which the density of the systematic risk factors attains its maximum value (i.e., the most likely point on the boundary):

$$\mathbf{z}_x := \arg\min\{\mathbf{z}^T \Sigma^{-1} \mathbf{z} : \mu(\mathbf{z}) = x\} . \tag{45}$$

The non-linear minimisation problem appearing above is easily and rapidly solved using standard techniques. We used `fmincon` function in Matlab.

As $\mathbf{z}_x$ lies on the boundary of the region of interest, roughly half the pilot sample will lie outside the region. In Section 5.2 we noted that it takes nearly one tenth of one second to numerically solve the equation $k'(\theta, \mathbf{z}) = x$. As such, if we are to compute $\hat{\theta}$ exactly (i.e., by numerically solving the indicated equation) for each sample point that lies outside the region of interest, the total time required (in seconds) to estimate the first-stage IS parameters will be at least $M_p/20$. In our numerical examples we use a pilot sample size of $M_p = 1000$, which means that it would take nearly one full minute to compute the first-stage IS parameters. This discussion suggests that reducing the number of times we must numerically solve the equation $k'(\theta, \mathbf{z}) = x$ could lead to a dramatic reduction in computational time.

We suggest fitting a low degree polynomial to the function $\hat{\theta}(x, \cdot)$, over a small region in $\mathbb{R}^2$ that contains all of the pilot sample points that lie outside the region of interest. Specifically, we determine the smallest rectangle that contains all of the pilot sample points, and discretize the rectangle using a mesh of n_g^2 points, equally spaced in each direction. Next, we identify those mesh points that lie outside the region of interest and compute $\hat{\theta}(x, \mathbf{z})$ exactly (i.e., by solving $k'(\theta, \mathbf{z}) = x$ numerically) for each such point. Finally, we fit a polynomial to the resulting $(\mathbf{z}, \hat{\theta}(x, \mathbf{z}))$ pairs and call the resulting function $\bar{\theta}(x, \cdot)$. Numerical evidence indicates at using a fifth-degree polynomial and a mesh with $15^2 = 225$ points leads to a sufficiently accurate approximation to $\hat{\theta}(x, \cdot)$ over the indicated range (the intersection of (i) the smallest rectangle that contains all sample points and (ii) the complement of the region of interest). Note that $\bar{\theta}$ could be an extremely inaccurate approximation to $\hat{\theta}$ outside this range, but that is not a concern because we will never need to evaluate it there.

It remains to compute $q(x, \mathbf{z})$ for each of the pilot points $\mathbf{z}$. For those points $\mathbf{z}$ that lie inside the region of interest, we set $q(x, \mathbf{z}) = 0$. For those points that lie inside the region, we set $q(x, \mathbf{z}) = \bar{\theta} x - k(\bar{\theta}, \mathbf{z})$, where $\bar{\theta} = \bar{\theta}(x, \mathbf{z})$. Evaluating $\bar{\theta}(x, \cdot)$ requires essentially no computational time (it is a polynomial), and if the mesh size and degree are chosen appropriately the difference between $\hat{\theta}$ and $\bar{\theta}$ is very small. In total, the suggested procedure reduces the number of evaluations of $\hat{\theta}$ from $M_p/2$ to $n_g/2$, for a percentage reduction of n_g^2/M_p. In our numerical examples we use $n_g = 15$ and $M_p = 1000$, which corresponds to a reduction of 75% in computational time.

To summarise, we estimate the optimal first-stage IS parameters as follows. First, we compute $\mathbf{z}_x$. Second, we draw a random sample of size M_p from the Gaussian distribution with mean vector $\mathbf{z}_x$ and covariance matrix Σ. Third, we construct $\bar{\theta}(x, \cdot)$, the polynomial approximation to $\hat{\theta}(x, \cdot)$, as described

in the previous paragraph. Fourth, for those sample points $\mathbf{z}$ that lie outside the region of interest we compute $q(x, \mathbf{z})$ using $\bar{\theta}$ instead $\hat{\theta}$. The estimates of the optimal first-stage IS parameters are then:

$$\hat{\mu}_{IS} = \frac{\sum_{m=1}^{M_p} w(\mathbf{Z}_m) \exp(-Nq(x, \mathbf{Z}_m))\mathbf{Z}_m}{\sum_{m=1}^{M_p} w(\mathbf{Z}_m) \exp(-Nq(x, \mathbf{Z}_m))}$$

and

$$\hat{\Sigma}_{IS} = \frac{\sum_{m=1}^{M_p} w(\mathbf{Z}_m) \exp(-Nq(x, \mathbf{Z}_m))(\mathbf{Z}_m - \hat{\mu}_{IS})(\mathbf{Z}_m - \hat{\mu}_{IS})^T}{\sum_{m=1}^{M_p} w(\mathbf{Z}_m) \exp(-Nq(x, \mathbf{Z}_m))},$$

where $\mathbf{Z}_1, \ldots, \mathbf{Z}_{M_p}$ is the random sample and

$$w(\mathbf{z}) = \frac{\phi(\mathbf{z}; \mathbf{0}, \Sigma)}{\phi(\mathbf{z}; \mathbf{z}_x, \Sigma)}$$

is the IS weight associated with shifting the mean of the systematic risk factors from $\mathbf{0}$ to $\mathbf{z}_x$. The upper left panel of Figure 3 illustrates a typical situation where the mean of the IS distribution lies "just inside" the region of interest.

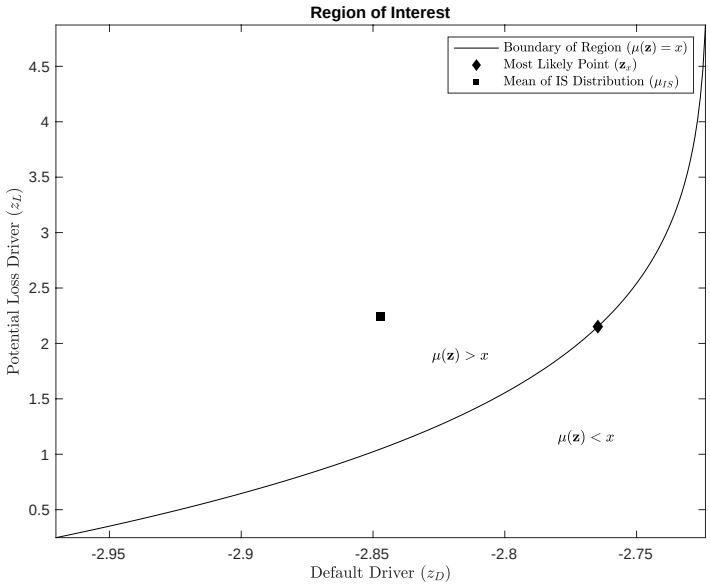

Figure 3. This figure illustrates the locations of (i) the importance sampling (IS) mean used for the pilot simulation and (ii) the IS mean used for the actual simulation, relative to the region of interest. Parameters (randomly selected using the procedure in Section 5.3) in both panels are $(P, \rho_D, \rho_L, \rho_I, \rho_S, a, b, N)$ = (0.0063, 0.3964, 0.2794, -0.3356, -0.7599, 0.6497, 0.5033, 134) and the threshold is x = 0.1575. Mean losses are $\mathbb{E}[L_i]$ = 0.0029.

6.2.2. Computing Parameters in the One-Factor Model

The procedure described in the previous section specialises in the one-factor case as follows. First, under the parameter restrictions outlined in Section 5.3, the expected loss function $\mu(z)$ is a strictly decreasing function of z. As such, the region of interest is the semi-infinite interval $(-\infty, z_x)$, where $z_x := \mu^{-1}(x)$, and its boundary is the single point z_x. In general z_x must be computed numerically, which is straightforward. Second, we draw a random sample of size M_p from the Gaussian distribution with mean z_x and unit variance. Third, the polynomial approximation to $\hat{\theta}$ is constructed by evaluating $\hat{\theta}$ exactly (i.e., by numerically solving the equation $k'(\theta, z) = x$) at each of n_g equally-spaced points z in the interval $[z_-, z_+]$, where z_- and z_+ are the largest and smallest values obtained in the pilot simulation, respectively, and then fitting a polynomial to the resulting $(z, \hat{\theta}(x, z))$ pairs. Fourth, we

evaluate $q(x,z)$ for each pilot sample point z as follows—if z lies inside the region of interest we set $q(x,z)=0$, otherwise we compute $q(x,z)$ by replacing the exact value $\hat{\theta}(x,z)$ with the approximate value $\bar{\theta}(x,z)$, where $\bar{\theta}$ is the polynomial constructed in the previous step. Note that a single evaluation of $\bar{\theta}$ requires far less computational time than a single evaluation of $\hat{\theta}$. Finally, the approximations to the first-stage IS parameters are:

$$\hat{\mu}_{IS} = \frac{\sum_{m=1}^{M_p} w(Z_m)\exp(-Nq(x,Z_m))Z_m}{\sum_{m=1}^{M_p} w(Z_m)\exp(-Nq(x,Z_m))}$$

and

$$\hat{\sigma}^2_{IS} = \frac{\sum_{m=1}^{M_p} w(Z_m)\exp(-Nq(x,Z_m))(Z_m-\hat{\mu}_{IS})^2}{\sum_{m=1}^{M_p} w(Z_m)\exp(-Nq(x,Z_m))},$$

where $Z_1,\ldots,Z_{M_p}$ is the random sample and

$$w(z) = \frac{\phi(z;0,1)}{\phi(z;z_x,1)}.$$

is the IS weight associated with shifting the mean of the systematic risk factor from 0 to z_x.

6.2.3. Trimming Large Weights

In the one-factor model the first-stage IS weight will have infinite variance whenever $\sigma^2_{IS} < 0.5$ (see Remark A1 in Appendix B). In a sample of 100 parameter sets, randomly selected according to the procedure in Section 5.3, the largest realised value of σ^2_{IS} was 0.38, and the mean and median were 0.11 and 0.09, respectively. It appears, then, that the first-stage IS weight in the one-factor model will have infinite variance in all cases of practical interest. We trim large weights as described in Section 4.2, using the set:

$$A = \{z\in\mathbb{R} : |z-\hat{\mu}_{IS}| \le C\hat{\sigma}_{IS}\}$$

for some constant C. In the numerical examples that follow we use $C=4$, in which case we expect to trim less than 0.01% of the entire sample. Specialising Equation (34) to the present context, we get that an upper bound on the associated bias is given by:

$$\int_{A^C} \exp(-Nq(x,z))\phi(z)\,dz\,, \tag{46}$$

which is straightforward (albeit slow) to compute using quadrature. Figure 4 illustrates the relationship between the probability of interest p_x and the upper bound of Equation (46) for the 100 randomly generated parameter sets, and clearly demonstrates that the bias associated with our trimming procedure is negligible. For instance, for probabilities on the order of 10^{-3} the bias is no larger than 10^{-5}, or 1% of the quantity of interest.

In the two-factor model the first-stage IS weight will have infinite variance whenever $\det(2\Sigma^{-1}-\Sigma^{-1}_{IS})<0$. In a random sample of 100 parameter sets, this condition occurred 96 times. As in the one-factor model, then, the first-stage IS weight in the two-factor model can be expected to have infinite variance in most cases of practical interest. We trim large weights using the set:

$$A = \left\{\mathbf{z}\in\mathbb{R}^2 : (\mathbf{z}-\hat{\mu}_{IS})^T\hat{\Sigma}^{-1}_{IS}(\mathbf{z}-\hat{\mu}_{IS})| \le C^2\right\}$$

for some constant C, and use $C=4$ in the numerical examples that follow.

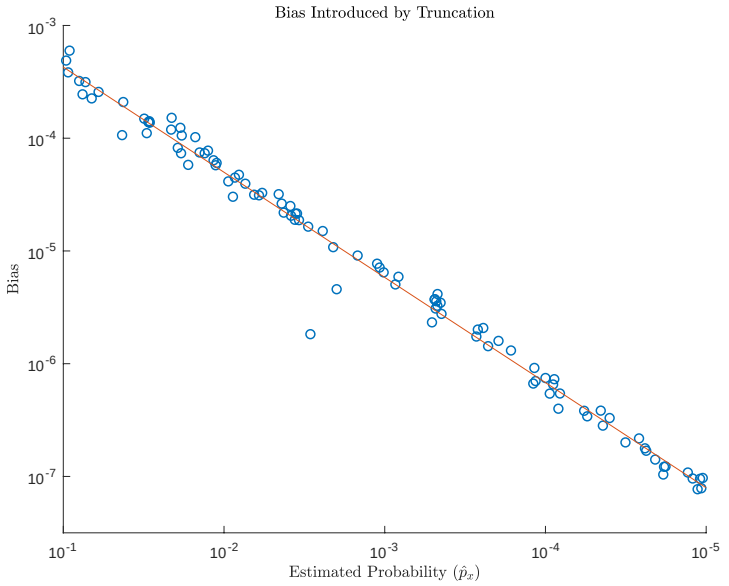

Figure 4. This figure illustrates the bias introduced by trimming large weights (vertical axis) as a function of the probability of interest (horizontal axis), for 100 randomly generated parameter sets in the one-factor case. For each set, we compute bias (in fact, an upper bound on the bias) by using quadrature to approximate Equation (46) and estimate the probability of interest using the full two-stage algorithm.

6.3. Second Stage

The first stage of the algorithm consists of (i) computing the first-stage IS parameters, (ii) simulating a random sample of size M from the systematic risk factors' IS distribution, and (iii) computing the associated IS weights, trimming large weights appropriately. Having completed these tasks, the next step is to simulate individual losses in the second stage. In the remainder of this section we let $\mathbf{z} = (z_D, z_L)$ denote a generic realisation of the systematic risk factors obtained in the first stage.

6.3.1. Approximating $\hat{\theta}$

Before generating any individual losses first construct the polynomial approximation to $\hat{\theta}$, using the same procedure described in Section 6.2.1. The basic idea is to fit a relatively low degree polynomial to the surface of $\hat{\theta}(x, \cdot)$, over a small region that contains all of the first-stage sample points. The values of $\mathbf{z}$ obtained in the pilot sample are invariably different from those obtained in the first stage, so it is essential that the polynomial is refit to account for this fact. In what follows we use $\bar{\theta}$ to approximate $\hat{\theta}$ whenever the numerical value of $\hat{\theta}$ is required, but since the difference between the two is small we do not distinguish between the two (i.e., we write $\hat{\theta}$ in this document, but use $\bar{\theta}$ in our code).

6.3.2. Sampling Individual Losses

In this section we describe how to sample individual losses in the two-factor model. The procedure carries over in an obvious way to the one-factor model, so we do not discuss that case explicitly.

If $\mathbf{z}$ lies inside the region of interest then the second stage is straightforward. For a given exposure i, we first simulate the exposure's idiosyncratic risk factors $\mathbf{Y}_i = (Y_{i,D}, Y_{i,L})$, from the bivariate normal distribution with standard normal margins and correlation ρ_I. Next, we set:

$$(X_{i,D}, X_{i,L}) = (\alpha_D z_D + \sqrt{1-\alpha_D^2}\, Y_{i,D}, \alpha_L z_L + \sqrt{1-\alpha_L^2}\, Y_{i,L}) \,.$$

If $X_{i,D} > \Phi^{-1}(P)$ then the exposure did not default and we set $L_i = 0$ and proceed to the next exposure. Otherwise the exposure did default, in which case we must compute $h(X_{i,L})$, set $\ell_i = h(x_{i,L})$ and then proceed to the next exposure. Note that we only evaluate h for defaulted exposures—this is important since evaluating h requires numerical inversion of the beta cdf, which is relatively slow. Having computed the individual losses associated with each exposure, we then compute the average loss $\bar{\ell} = N^{-1}\sum_{i=1}^{N} \ell_i$ and set $\Lambda_2(\mathbf{z}, \bar{\ell}) = 1$.

If $\mathbf{z}$ lies outside the region of interest we must compute $\hat{\theta}$, $k(\hat{\theta})$ and $\hat{c}$, which we do approximately using the polynomial approximation $\bar{\theta}$. We then sample from $\hat{g}_x(\cdot|\mathbf{z})$ as follows. First simulate the idiosyncratic risk factors $\mathbf{Y}_i = (Y_{i,D}, Y_{i,L})$ from the bivariate normal distribution with standard normal margins and correlation ρ_I. Also generate a random number U, independent of $\mathbf{Y}_i$. Then set:

$$(X_{i,D}, X_{i,L}) = (\alpha_D z_D + \sqrt{1-\alpha_D^2}Y_{i,D}, \alpha_L z_L + \sqrt{1-\alpha_L^2}Y_{i,L}) \,.$$

If the exposure did not default we set $\hat{L}_i = 0$, otherwise we compute h and set $\hat{L}_i = h(X_{i,L})$. Next we check whether or not

$$U \leq \frac{1}{\hat{c}} \cdot \frac{\hat{g}_x(\hat{L}_i|\mathbf{z})}{g(\hat{L}_i|\mathbf{z})} = \exp(-\hat{\theta}(\ell_{\max} - \hat{L}_i)) \tag{47}$$

then accept $\hat{L}_i$ as a drawing from $\hat{g}_x$, that is, set $L_i = \hat{L}_i$ and proceed to exposure i. Otherwise, draw another random number U and set of idiosyncratic factors. Once we have sampled the individual losses associated with each exposure we compute the average loss $\bar{\ell} = N^{-1}\sum_{i=1}^{N} \ell_i$ and set $\Lambda_2(\mathbf{z}, \bar{\ell}) = \exp(-N[\hat{\theta}\bar{\ell} - k(\hat{\theta}, \mathbf{z})])$, using the polynomial approximation to estimate the value of $\hat{\theta}$.

6.3.3. Efficiency of the Second Stage

The frequency with which the rejection sampling algorithm must be applied in the second stage is governed by $\mathbb{P}_{IS}(\mu(\mathbf{Z}) > x)$. The left panel of Figure 5 illustrates the empirical distribution of this probability across 100 randomly selected parameter sets. The distribution is concentrated towards small values (the median fraction is 27%) but does have a relatively thick right tail (the mean fraction is 35%). In some cases—particularly when the value of the parameter ρ_D is close to zero, in which case individual losses are very nearly independent and systematic risk is largely irrelevant—the vast majority of first-stage simulations require further IS in the second stage.

The efficiency of the rejection sampling algorithm, when it must be applied, is governed by the conditional distribution of $\hat{c} = \hat{c}(x, \mathbf{Z})$ given that $\mu(\mathbf{Z}) < x$. For each of the 100 parameter sets we estimate $\mathbb{E}_{IS}[\hat{c}(x, \mathbf{Z})|\mu(\mathbf{Z}) < x]$, which determines the average size of the rejection constant for a given set of parameters, by computing the associated value of $\hat{c}$ for each first-stage realisation that lies outside the region of interest and then averaging the resulting values. The right panel of Figure 5 illustrates the results, and we note that the mean and median of the data presented there are 1.09 and 1.17, respectively. The figure clearly indicates that the rejection sampling algorithm can be expected to be quite efficient, whenever it must be applied.

The distributions of $\mathbb{P}_{IS}(\mu(\mathbf{Z}) < x)$ and $\mathbb{E}_{IS}[\hat{c}(x, \mathbf{Z})|\mu(\mathbf{Z}) < x]$ across parameters depend heavily on whether or not we adjust the variance of the systematic risk factors in the first stage. When we do not adjust variance, the mean and median of $\mathbb{P}_{IS}(\mu(\mathbf{Z}) < x)$ (across 100 randomly selected parameter sets) rise to 49% and 45% (as compared to 35% and 27% when we do adjust variance), and the mean and median of $\mathbb{E}_{IS}[\hat{c}(x, \mathbf{Z})|\mu(\mathbf{Z}) < x]$ rise to 18.6 and 1.8, respectively (as compared to 1.17 and 1.09 when we do adjust variance).

Remark 7. *If we do not adjust the variance of the systematic risk factors in the first stage, then (i) the rejection sampling algorithm must be applied more frequently and (ii) is less efficient whenever it must be applied. As such, adjusting the variance of the systematic risk factors reduces the total time required to implement the two-stage algorithm.*

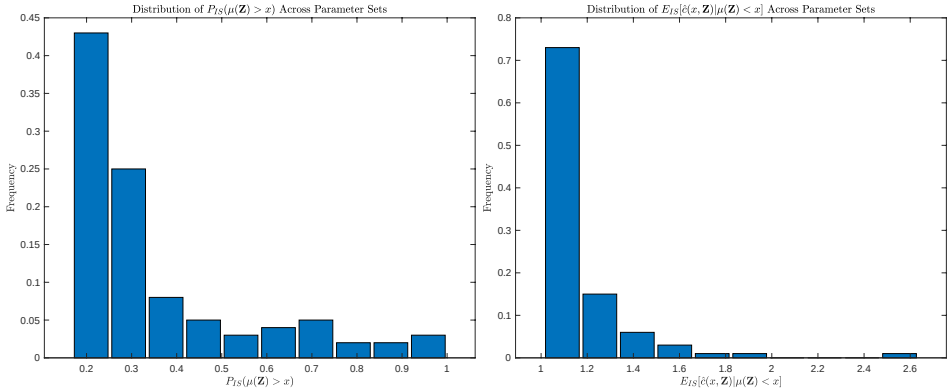

Figure 5. This figure illustrates the variation of $\mathbb{P}_{IS}(\mu(\mathbf{Z}) < x)$ (left panel) and $\mathbb{E}_{IS}[\hat{c}(x,\mathbf{Z})|\mu(\mathbf{Z}) < x]$ (right panel) across model parameters. Recall that the former quantity determines the frequency with which the second-stage rejection sampling algorithm must be applied and the latter quantity determines the efficiency of the algorithm when it must be applied. For each of 100 parameter sets, randomly selected according to the procedure described in Section 5.3, we compute the first-stage IS parameters and then draw 10,000 realisations of the systematic risk factors from the variance adjusted first-stage IS density.

The intuition behind this fact is as follows. First recall that the mean of the systematic risk factors tends to lie just inside the region of interest (recall Figure 3). In such cases the effect of reducing the variance of the systematic risk factors is to concentrate the distribution of $\mathbf{Z}$ just inside the boundary of the region of interest. Not only will this ensure that more first-stage realisations lie inside the region of interest (thereby reducing the fraction of points that require further IS in the second stage), it will also ensure that those realisations that lie outside the region (i.e., for which $\mu(\mathbf{z}) < x$) do not lie "that far" outside the region (i.e., that $\mu(\mathbf{z})$ is not "that much less" than x), which in turn ensures that the typical size of $\hat{c}$ is relatively close to one (recall the left panel of Figure 1).

7. Performance Evaluation

In this section we investigate the proposed algorithms' performance in terms of statistical accuracy, computational time, and overall. Unless otherwise mentioned, we use a pilot sample size of $M_p = 1000$ to estimate the first-stage IS parameters and a sample size of M = 10,000 to estimate the probability of interest (p_x). We use the value $C = 4$ to trim large first-stage IS weights, and a value of $c_{\max} = 10$ to trim large rejection constants.

7.1. Statistical Accuracy

The standard error of any estimator that we consider is of the form $\nu_x/\sqrt{M}$ for some constant ν_x that depends on the algorithm used and the model parameters. For instance, for the one-stage estimator in the two-factor case we have $\nu_x = \mathrm{SD}_{1S}(\Lambda_1(\mathbf{Z}) \cdot \mathbf{1}_{\{\bar{L}_N > x\}})$, where SD_{1S} denotes standard deviation under the one-stage IS density of Equation (31). Note that in the absence of IS we have $\nu_x = \sqrt{p_x(1-p_x)} \sim p_x^{0.5}$ as $p_x \to 0$.

Figure 6 illustrates the relationship between ν_x and p_x using 100 randomly selected parameters sets, for the two-stage algorithm and in the two-factor case. Importantly, we see that (i) ν_x seems to be a function of p_x (i.e., it only depends on model parameters through p_x) and (ii) for small probabilities the functional relationship appears to be of the form $\nu_x = a p_x^b$ for constants a and b. These features are also present in the case of the one-stage estimator, as well as for both estimators in the one-factor model. The numerical values of a and b are easily estimated using the line of best fit (on the logarithmic scale), and the estimated values for both the one- and two-factor cases are summarised in Table 1. Of particular note is the fact that the value of b is extremely close to one in every case.

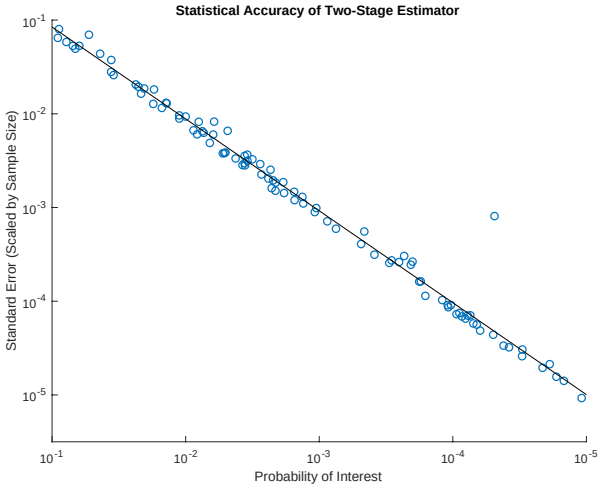

Figure 6. This figure illustrates the relationship between ν_x and p_x, where ν_x is the standard deviation of $\Lambda_1(\mathbf{Z})\Lambda_2(\bar{L}_N, \mathbf{Z})\mathbf{1}_{\{\bar{L}_N \geq x\}}$ under the two-stage IS density of Equation (32), in the two-factor case. The numerical values of p_x and ν_x are estimated for each of 100 randomly generated parameters sets, according to the procedure described in Section 5.3.

Table 1. This table reports fitted values of the relationship $\nu_x \approx a p_x^b$ for each estimator (one- and two-stage) and each model (one- and two-factor). Values of a and b are obtained by determining the line of best fit on the logarithmic scale (i.e., the line appearing in Figure 6). Note that in the absence of IS we would have $\nu_x = \sqrt{p_x(1-p_x)} \approx p_x^{0.5}$.

	One-Stage Algorithm	**Two-Stage Algorithm**
One-Factor Model	$0.91 p_x^{0.98}$	$0.81 p_x^{0.99}$
Two-Factor Model	$0.98 p_x^{0.98}$	$0.81 p_x^{0.98}$

Of particular interest in the rare event context is an estimator's relative error, defined as the ratio of its standard error to the true value of the quantity being estimated. For any of the estimators that we consider, the component of relative error that does not depend on sample size is $\nu_x / p_x \approx a p_x^{b-1}$. In the absence of IS we have $b - 1 = -0.5$, in which case relative error grows rapidly as $p_x \to 0$ (i.e., $\nu_x \to 0$ but $\nu_x / p_x \to \infty$ as $p_x \to 0$). By contrast, $b \approx 1$ for any of our IS estimators, in which case there is weak dependence of relative error on p_x. The minimum sample size required to ensure that an estimator's relative error does not exceed the threshold ϵ is $v_x^2/(p_x\epsilon)^2 \approx a^2 p_x^{2(b-1)} \epsilon^{-2}$. In the absence of IS we have $b \approx 0.5$, in which case the sample size (and therefore computational burden) required to achieve a given degree of accuracy increases rapidly as $p_x \to 0$. By contrast, for all of our IS estimators we have $b \approx 1$, in which case the minimum sample size (and computational burden) is nearly independent of p_x.

Our ultimate goal is to reduce the computational burden associated with estimating p_x, in situations where p_x is small. To see how effective the proposed algorithms are in this regard, note that the sample size required to achieve a given degree of accuracy using the proposed algorithm, relative to that required to achieve the same degree of accuracy in the absence of IS, is approximately

$$\frac{a^2 p_x^{2(b-1)} \epsilon^{-2}}{p_x^{-1} \epsilon^{-2}} = a^2 p_x^{2b-1},$$

which does not depend on ϵ. Since $a < 1$ and $b > 0.5$ (recall Table 1), we have that $a^2 p_x^{2b-1} < p_x$.

Remark 8. *The relative sample size required to achieve a given degree of accuracy using the proposed algorithm, relative to that required in the absence of IS, is not larger than the probability of interest. For example, if the probability of interest is approximately 1%, then the proposed algorithm requires a sample size that is less than 1% of what would be required in the absence of IS (regardless of the desired degree of accuracy). And if the probability of interest is 0.1%, then the proposed algorithm requires a sample size that is less than 0.1% of what would be required in the absence of IS. In other words, the proposed algorithm is extremely effective at reducing the sample size required to achieve a given degree of accuracy.*

It is also insightful to compare the efficiency of the two-stage estimator, relative to the one-stage estimator. In the one-factor case, the minimum sample size required using the two-stage algorithm, relative to that required using the one-stage algorithm, is approximately:

$$\frac{0.66 p_x^{-0.02} \epsilon^{-2}}{0.83 p_x^{-0.04} \epsilon^{-2}} = 0.80 p_x^{0.02} .$$

As p_x ranges from 1% to 0.01% the estimated relative sample size ranges from 0.73 to 0.67. In the two-factor case, the relative sample size is approximately 0.69, regardless of the value of p_x.

Remark 9. *In both the one- and two-factor models, the two-stage algorithm is more efficient than the one-stage algorithm, in the sense that it requires a smaller sample size in order to achieve a given degree of accuracy. Indeed, in cases of practical interest (probabilities in the range of 1% to 0.01%) the minimum sample size required to achieve a given degree of accuracy using the two-stage algorithm is roughly 70% of what would be required using the one-stage algorithm.*

7.2. Computational Time

Figure 7 illustrates the relationship between sample size (M) and run time (total time required to estimate p_x using a particular algorithm), for one randomly selected set of parameters. Across both models and algorithms, the relationship is almost perfectly linear. In the absence of IS the intercept is zero (i.e., run time is directly proportional to sample size), whereas the intercepts are non-zero for the IS algorithms. The non-zero intercepts are due to the overhead associated with (i) computing the first-stage IS parameters, which accounts for almost all of the difference between the intercepts of the solid (no IS) and dashed (one-stage IS) intercepts, and (ii) computing the second-stage polynomial approximation to $\hat{\theta}$, which accounts for almost all of the difference between the intercepts of the the dashed (one-stage IS) and dash-dot (two-stage IS) lines. It is also worth noting that a given increase in sample size will have a greater impact on the run times for the IS algorithms than it will on the standard algorithm. This is because we only calculate $h(X_{i,L})$ for defaulted exposures (evaluating $h(\cdot)$ is slow because it requires numerical inversion of the beta distribution function), and the default rate is higher under the IS distribution.

Across 100 randomly generated parameter sets, portfolio size (N) is most highly correlated with run time and the relationship is roughly linear. Table 2 reports summary statistics on run times, across algorithms and models.

Table 2. This table reports summary statistics—in seconds, and across 100 randomly selected parameter sets—for total run time (first three columns), time required to estimate the first-stage IS parameters (fourth column) and time required to fit the second-stage polynomial approximation to $\hat{\theta}$ (final column).

Average Run Times					
	No IS	**One-Stage IS**	**Two-Stage IS**	μ_{IS}, Σ_{IS}	$\hat{\theta}$
One Factor	7.3	25.6	33.7	1.5	0.8
Two Factor	7.4	39.0	55.5	14.3	8.9

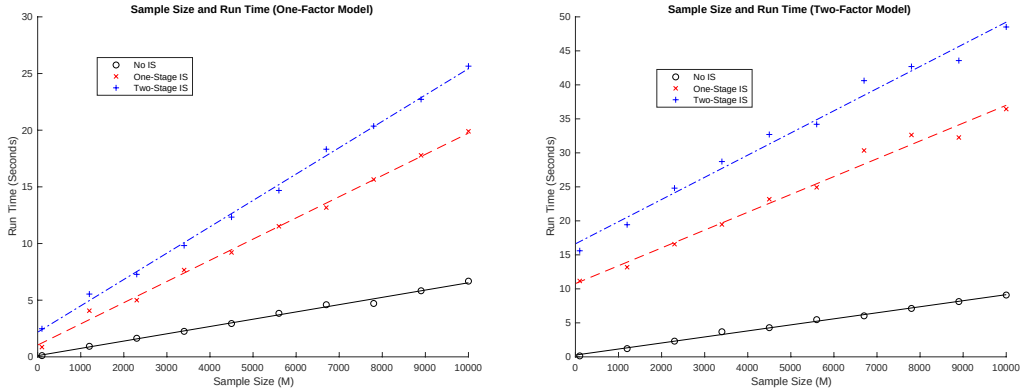

Figure 7. This figure illustrates the relationship between sample size (M) and run time (total CPU time required to estimate p_x by a particular algorithm), using a set of parameters randomly selected according to the procedure described in Section 5.3. For each value of M we use a pilot sample that is 10% as large as the sample that is eventually used to estimate p_x (i.e., we set $M_p = 0.1M$). The left panel corresponds to the one-factor model and parameter values are $(P, \rho_D, \rho_L, \rho_I, \rho_S, a, b)$ = (0.0827, 0.1000, 0.3629, −0.0180, −1, 0.6676, 0.8751) and $N = 2334$. The right panel corresponds to the two-factor model and parameter values are $(P, \rho_D, \rho_L, \rho_I, \rho_S, a, b)$ = (0.0241, 0.2322, 0.0343, 0.1650, 0.4135, 0.4056, 0.4942) and $N = 3278$.

7.3. Overall Performance

Recall that the ultimate goal of this paper is to reduce the computational burden associated with estimating p_x, when p_x is small. The computational burden associated with a particular algorithm is a function of both its statistical accuracy and total run time. We have seen that the proposed algorithms are substantially more accurate, but require considerably more run time. In this section we demonstrate that the benefit of increased accuracy is well worth the cost of additional run time, by considering the amount of time required by a particular algorithm in order to achieve a given degree of accuracy (as measured by relative error).

To begin, let $t(M)$ denote the total run time required by a particular algorithm to estimate p_x using a sample of size M. As illustrated in Figure 7 we have $t(M) \approx c + dM$ for constants c and d that depend on the underlying model parameters (particularly portfolio size, N) as well as the algorithm being used. In Section 7.1 we saw that the minimum sample size required to ensure that the estimator's relative error does not exceed the threshold ϵ isL

$$M(\epsilon) \approx a^2 p_x^{2(b-1)} \epsilon^{-2} ,$$

for constants a and b depending on the underlying model (one- or two-factor) and algorithm being used. Thus, if $T(\epsilon)$ denotes the total CPU time required to ensure that the estimator's relative error does not exceed ϵ, we have:

$$T(\epsilon) \approx c + da^2 p_x^{2(b-1)} \epsilon^{-2} . \tag{48}$$

Table 3 contains sample calculations for several different values of p_x and ϵ, using the data appearing in the left panel of Figure 7 to estimate c and d and the values of a and b implicitly reported in Table 1. The results reported in the table are representative of those obtained using different parameter sets. It is clear that the proposed algorithms can substantially reduce the computational burden associated with accurate estimation of small probabilities. For instance, if the probability of interest is on the order of 0.1% then either of the proposed algorithms can achieve 5% accuracy within 2–3 s, as compared to 4 min (80 times longer) in the absence of IS.

Table 3. This table reports the time (in seconds) required to achieve a given degree of accuracy (computed using Equation (48)) for several values of p_x and ϵ, for the parameter values corresponding to the left panel of Figure 7. Note that this is for the one-factor model. Values of c and d are obtained from the lines of best fit appearing in the left panel of Figure 7, values of a and b are obtained from Table 1.

No IS				One-Stage IS (Two-Stage IS)			
ϵ \ p_x	1%	0.1%	0.01%	ϵ \ p_x	1%	0.1%	0.01%
10%	6	60	600	10%	1.2 (2.3)	1.2 (2.3)	1.3 (2.4)
5%	24	240	2400	5%	1.8 (2.8)	1.9 (2.9)	1.9 (2.9)
1%	600	6000	60,000	1%	20.0 (18.8)	21.8 (19.6)	23.8 (20.4)

The two-stage estimator is statistically more accurate (Section 7.1) but computationally more expensive (Section 7.2) than the one-stage estimator. It is important to determine whether or not the benefit of increased accuracy outweighs the cost of increased computational time. Table 3 suggests that, in some cases at least, implementing the second stage is indeed worth the effort, in the sense that it can achieve the same degree of accuracy in less time.

Figure 8 illustrates the overall efficiency of the proposed algorithms, as a function of the desired degree of accuracy. Specifically, the left panel illustrates the ratio of (i) the total CPU time required to ensure the standard estimator's relative error does not exceed a given threshold to (ii) the total time required by the proposed algorithms, for a randomly selected set of parameter values in the one-factor model. The right panel illustrates the same ratio for a randomly selected set of parameters in the two-factor model.

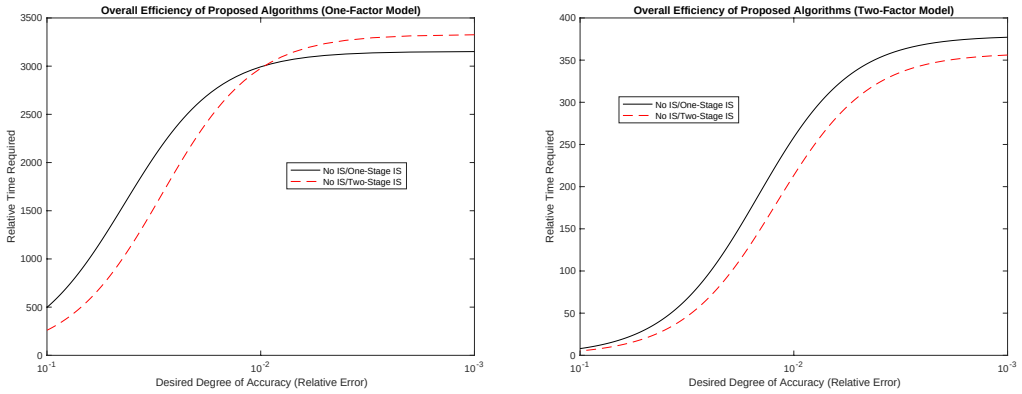

Figure 8. This figure illustrates the overall efficiency of the proposed algorithms. Specifically, the solid [dashed] line in the left panel illustrates the ratio of (i) the total run time (in seconds) required to ensure that the standard estimate's relative error does not exceed a given threshold to (ii) the run time required by the one-stage [two-stage] algorithm, in the one-factor model. The right panel corresponds to the two-factor model. Parameter values are the same as in Figure 7 and Table 3.

In the one-factor model, it would take hundreds of times longer to obtain an estimate of p_x whose relative error is less than 10%, and thousands of times longer to obtain an estimate whose relative error is less than 1%. The figure also suggests that, since it requires less run time to obtain very accurate estimates, the two-stage algorithm is preferable to the one-stage algorithm in the one-factor model. In the two-factor model—where estimating IS parameters and fitting the second-stage polynomial approximation to $\hat{\theta}$ is more time consuming—the proposed algorithms are hundreds of times more efficient than the standard algorithm. In addition, it appears that the one-stage algorithm is preferable to the two-stage algorithm in this case. Although the numerical values discussed here are specific to

the parameter set used to produce the figure, they are representative of other parameter sets. In other words, the behaviour illustrated in Figure 8 is representative of the general framework overall.

8. Concluding Remarks

This paper developed an importance sampling (IS) algorithm for estimating large deviation probabilities for the loss on a portfolio of loans. In contrast to existing literature, we allowed loss given default to be stochastic and correlate with the default rate. The proposed algorithm proceeded in two stages. In the first stage one generates systematic risk factors from an IS distribution that is designed to increase the rate at which adverse macroeconomic scenarios are generated. In the second stage one checks whether or not the simulated macro environment is sufficiently adverse—if it is then no further IS is applied and idiosyncratic risk factors are drawn from their actual (conditional) probability distribution, if it is not then one indirectly applies IS to the conditional distribution of the idiosyncratic risk factors. Numerical evidence indicated that the proposed algorithm could be thousands of times more efficient than algorithms that did not employ any variance reduction techniques, across a wide variety of PD-LGD correlation models that are used in practice.

Author Contributions: Both authors contributed equally to all parts of this paper. Both authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSERC Discovery Grant 371512.

Acknowledgments: This work was made possible through the generous financial support of the NSERC Discovery Grant program. The authors would also like to thank Agassi Iu for invaluable research assistance.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Exponential Tilts and Large Deviations

Let $X_1, X_2, \ldots,$ be independent and identically distributed random variable with common density $f(x)$, having bounded support $[x_{\min}, x_{\max}]$, and common mean $\mu = \mathbb{E}[X_i]$. For $\theta \in \mathbb{R}$ we let $m(\theta) = \mathbb{E}[\exp(\theta X_i)]$ and $k(\theta) = \log(m(\theta))$ denote the common moment generating function (mgf) and cumulant generating function (cgf) of the X_i, respectively. Note that $\mu = m'(0) = k'(0)$.

Appendix A.1. Properties of $k(\theta)$

Elementary properties of cgfs ensure that $k'(\cdot)$ is a strictly increasing function that maps $\mathbb{R}$ onto $(x_{\min}, x_{\max})$. One implication is that, for fixed $t \in (x_{\min}, x_{\max})$, the graph of the function $\theta \mapsto \theta t - k(\theta)$ is $\cap$-shaped. The graph also passes through the origin, and its derivative at zero is $t - \mu$. If this derivative is positive (i.e., if $\mu < t$) then the unique maximum is strictly positive and occurs to the right of the origin. If it is negative (i.e., if $\mu > t$) then the unique maximum is strictly positive and occurs to the left of the origin. If it is zero (i.e., if $\mu = t$) then the unique maximum of zero is attained at the origin.

For a given $t \in (x_{\min}, x_{\max})$, there is a unique value of θ for which $k'(\theta) = t$. We let $\widetilde{\theta} = \widetilde{\theta}(t)$ denote this value of θ. Note that $\widetilde{\theta}(t)$ is a strictly increasing function of t and that $\widetilde{\theta}(\mu) = 0$. Thus $\widetilde{\theta}$ is positive [negative] whenever $t > \mu$ [$t < \mu$]. An important quantity in what follows is $\widehat{\theta} = \widehat{\theta}(t) := \max(0, \widetilde{\theta}(t))$, which can be interpreted as the unique value of θ for which $k'(\theta) = \max(\mu, t)$. Note that if $t \leq \mu$ then $\widehat{\theta} = 0$, and if $t > \mu$ then $\widehat{\theta}(t) > 0$.

Appendix A.2. Legendre Transform of $k(\theta)$

We let $q(\cdot)$ denote the Legendre transform of $k(\cdot)$ over $[0, \infty)$. That is,

$$q(t) := \max_{\theta \geq 0}(\theta t - k(\theta)) = \hat{\theta} t - k(\hat{\theta}) \,, \tag{A1}$$

where $\hat{\theta} = \hat{\theta}(t)$ was defined in the previous section, and is the (uniquely defined) point at which the function $\theta \mapsto \theta t - k(\theta)$ attains its maximum on $[0, \infty)$. Based on the discussion in the preceding

paragraph, we see that $\hat{\theta}(t) = q(t) = 0$ whenever $\mu \geq t$, whereas both $\hat{\theta}(t)$ and $q(t)$ are strictly positive whenever $\mu < t$.

The derivative of the transform q is demonstrably equal to:

$$q'(t) = \hat{\theta}(t) + \hat{\theta}'(t) \cdot [t - k'(\hat{\theta}(t))] \, .$$

Since $\hat{\theta} = 0$ whenever $t \leq \mu$ and $k'(\hat{\theta}) = t$ whenever $t > \mu$, the second term above vanishes for all t, and we find that:

$$q'(t) = \hat{\theta}(t) \, . \tag{A2}$$

Appendix A.3. Exponential Tilts

For $\theta \in \mathbb{R}$ we define:

$$f_\theta(x) := \exp(\theta x - k(\theta)) \cdot f(x) \, . \tag{A3}$$

The density f_θ is called an exponential tilt of f. As the value of the tilt parameter θ varies, we obtain an exponential family of densities (exponential families have lots of very useful properties, and this is an easy way of constructing them). If θ is positive then the right and left tails of f_θ are heavier and thinner, respectively, than those of f. The opposite is true if θ is negative. The larger in magnitude is θ, the greater the discrepancy between f and f_θ; indeed the Kullback–Leibler divergence from f_θ to f is $-\theta\mu + k(\theta)$, which is a strictly convex function of θ that attains its minimum value (of zero) at $\theta = 0$.

It is readily verified that $k'(\theta) = \mathbb{E}_\theta[X_i]$, where $\mathbb{E}_\theta$ denotes expectation with respect to f_θ. This observation, in combination with the developments in Section A.1, implies that it is always possible to find a density of the form (A3) whose mean is t, whatever the $t \in (x_{\min}, x_{\max})$. Indeed $f_{\hat{\theta}}$ is precisely such a density. Under mild conditions, $f_{\hat{\theta}}(\cdot)$ can be characterised as that density that most resembles f (in the sense of minimum divergence), among all densities whose mean is x (and are absolutely continuous with respect to f).

Recall that $\hat{\theta}$ is the unique value of θ for which $k'(\theta) = \max(t, \mu)$. We can therefore interpret $f_{\hat{\theta}}$ as that density that most resembles f, among all densities whose mean is at least t (and that are absolutely continuous with respect to f). Note in particular hat the mean of $f_{\hat{\theta}}$ is $\max(\mu, t)$. The numerical value of $\hat{\theta}$ can therefore be interpreted as the degree to which we must deform the density f, in order to produce a density whose mean is at least t. If $\mu \geq t$ then $\hat{\theta} = 0$ and no adjustment is necessary. If $\mu < t$ then $\hat{\theta} > 0$ and mass must be transferred from the left tail to the right; the larger the discrepancy between μ (the mean of f) and t (the desired mean), the larger is $\hat{\theta}$.

Appendix A.4. Behaviour of X_i, Conditioned on a Large Deviation

Let $f_t(x)$ denote the conditional density of X_i, given that $\overline{X}_N > t$, where $\overline{X}_N = \frac{1}{N}\sum_{i=1}^N X_i$. We suppress the dependence of f_t on N for simplicity. Using Bayes' rule we get

$$f_t(x) = \frac{\mathbb{P}(\bar{X}_N > t | X_i = x)}{\mathbb{P}(\bar{X}_N > t)} \cdot f(x) \, ,$$

and since the X_i are independent, we get

$$\mathbb{P}(\bar{X}_N > t | X_i = x) = \mathbb{P}(\bar{X}_{N-1} > t + \tfrac{t-x}{N-1}) \, .$$

Now, using the large deviation approximation $\mathbb{P}(\overline{X}_N \geq t) \approx \exp(-N \cdot q(t))$, we get that

$$\frac{\mathbb{P}(\bar{X}_N > t | X_i = x)}{\mathbb{P}(\bar{X}_N > t)} \approx \exp(-(N-1)q(t + \tfrac{t-x}{N-1}) + Nq(t)) \, .$$

Now if N is large then

$$q(t + \tfrac{t-x}{N-1}) \approx q(t) + q'(t) \cdot \tfrac{t-x}{N-1} = q(t) + \hat{\theta} \cdot \tfrac{t-x}{N-1} \, ,$$

where we have used the fact that $q'(t) = \hat{\theta}(t)$. Putting everything together we arrive at the approximation

$$\frac{\mathbb{P}(\overline{X}_N > t | X_i = x)}{\mathbb{P}(\overline{X}_N > t)} \approx \exp(\hat{\theta}x - k(\hat{\theta})) \, ,$$

which leads to the approximation

$$f_t(x) \approx \exp(\hat{\theta}x - k(\hat{\theta})) \cdot f(x) \, . \tag{A4}$$

We may thus interpret the conditional density f_t as that density which most resembles the unconditional density f, but whose mean is at least t.

Appendix A.5. Approximate Behaviour of $(X_1, X_2, \ldots, X_N)$*, Conditioned on a Large Deviation*

Let $\hat{f}_t(\mathbf{x}) = \hat{f}_t(x_1, \ldots, x_N)$ denote the conditional density of $(X_1, \ldots, X_N)$, given that $\overline{X}_N > t$. Then

$$\hat{f}_t(\mathbf{x}) = \frac{\prod_{i=1}^N f(x_i)}{p_t} \, , \quad \mathbf{x} \in A_{N,t} \, ,$$

where $p_t = \mathbb{P}(\overline{X}_N > t)$ and $A_{N,t}$ is the set of those points $\mathbf{x} \in [x_{\min}, x_{\max}]^N$ whose average value exceeds t.

We seek a density $h(x)$, supported on $[x_{\min}, x_{\max}]$, which minimizes the Kullback-Leibler divergence (KLD) of

$$\hat{h}(\mathbf{x}) := \prod_{i=1}^N h(x_i)$$

from $\hat{f}_t$. In other words, we seek an independent sequence $Y_1, Y_2, \ldots, Y_N$ (whose density is $\hat{h}$) whose behaviour most resembles (in a certain sense) the behaviour of $X_1, X_2, \ldots, X_N$, conditioned on the large deviation $\overline{X}_N > t$.

Now let $\mathbb{E}_g$ denote expectation with respect to the density g. Then the divergence of $\hat{h}$ from $\hat{f}_t$ is

$$\begin{aligned}
\mathbb{E}_{\hat{f}_t}[\log(\hat{f}_t(\mathbf{X})/\hat{h}(\mathbf{X}))] &= \sum_{i=1}^N \mathbb{E}_{\hat{f}_t}\left[\log\left(f(X_i)/h(X_i)\right)\right] - \log(p_t) \\
&= N \cdot \mathbb{E}_{\hat{f}_t}\left[\log\left(f(X_1)/h(X_1)\right)\right] - \log(p_t) \\
&= N \cdot \mathbb{E}_{f_t}\left[\log\left(f(X_1)/h(X_1)\right)\right] - \log(p_t) \\
&= N \cdot \mathbb{E}_{f_t}\left[\log\left(f(X_1)/f_t(X_1)\right)\right] + N \cdot \mathbb{E}_{f_t}\left[\log\left(f_t(X_1)/h(X_1)\right)\right] - \log(p_t)
\end{aligned}$$

Now, the middle term in the above display is the KLD of h from f_t. As such it is non-negative, and is equal to zero if and only if $h = f_t$. It follows immediately that the divergence of $\hat{h}$ from $\hat{f}_t$ is minimised by setting $h = f_t$.

Appendix B. Important Exponential Families

This appendix considers two important special cases—the Gaussian and t families—of the general setting discussed in Section 2.2.

Appendix B.1. Gaussian

Suppose first that the $\mathbf{Z}$ is Gaussian with mean vector $\mu_0 \in \mathbb{R}^d$ and positive definite covariance matrix Σ_0. When specifying the IS distribution, one can either (i) shift the mean of $\mathbf{Z}$ but leaves its covariance structure unchanged or (ii) shift its mean and adjust its covariance structure. In general the latter approach will lead to a better approximation of the ideal IS density but more volatile IS weights.

If we take the former approach (shifting mean, leaving covariance structure unchanged), the implicit family in which we are embedding f is the Gaussian family with arbitrary mean vector $\mu \in \mathbb{R}^d$ and fixed covariance matrix Σ_0. To this end, let $f(\mathbf{z}) = \phi(\mathbf{z}; \mu_0, \Sigma_0)$ denote the Gaussian density with mean vector μ_0 and covariance matrix Σ_0 and let $f_\lambda(\mathbf{z}) = \phi(\mathbf{z}; \mu, \Sigma_0)$. It remains to identify the natural sufficient statistic and write the natural parameter λ in terms of the mean vector μ. To this end, note that

$$\frac{f_\lambda(\mathbf{z})}{f(\mathbf{z})} = \exp\left((\mu^T - \mu_0^T)\Sigma_0^{-1}\mathbf{z} - \frac{1}{2}\mu^T\Sigma^{-1}\mu + \frac{1}{2}\mu_0^T\Sigma^{-1}\mu_0\right) .$$

The natural sufficient statistic is therefore

$$S(\mathbf{z}) = (z_1, \ldots, z_d) ,$$

the natural parameter is

$$\lambda(\mu) = \Sigma_0^{-1}(\mu - \mu_0) .$$

Note that we can write $\mu(\lambda) = \mu_0 + \Sigma_0\lambda$, so that the natural parameter represents a sort of normalized deviation from the actual mean μ_0 to the IS mean μ. Lastly, we see that the cgf of $S(\mathbf{Z})$ is

$$K(\lambda) = \frac{1}{2}\left[\mu_\lambda^T\Sigma_0^{-1}\mu_\lambda - \mu_0^T\Sigma_0^{-1}\mu_0^T\right] = \lambda^T\mu_0 + \frac{1}{2}\lambda^T\Sigma_0\lambda ,$$

where we have written μ_λ instead of $\mu(\lambda)$ in the above display. Clearly, we have that both $K(\lambda)$ and $K(-\lambda)$ are well-defined for all $\lambda \in \mathbb{R}^d$. The implication is that if we shift the mean of $\mathbf{Z}$ but leave its covariance structure unchanged, the IS weight will have finite variance regardless of what IS mean we choose.

If we take the former approach (shifting mean, adjusting covariance) the implicit family in which we are embedding f is the Gaussian family with arbitrary mean vector μ and arbitrary positive definite covariance matrix Σ. In this case we have $f_\lambda(\mathbf{z}) = \phi(\mathbf{z}; \mu, \Sigma)$ and the ratio of $f_\lambda(\mathbf{z})$ to $f(\mathbf{z})$ is

$$\exp((\mu^T\Sigma^{-1} - \mu_0^T\Sigma_0^{-1})\mathbf{z} + \frac{1}{2}\mathbf{z}^T(\Sigma_0^{-1} - \Sigma^{-1})\mathbf{z} - \hat{K}(\mu, \Sigma)) ,$$

where

$$\hat{K}(\mu, \Sigma) = \frac{1}{2}\left[\mu^T\Sigma^{-1}\mu - \mu_0^T\Sigma_0^{-1}\mu_0 + \log(\det(\Sigma) - \log(\det(\Sigma_0)\right] .$$

The natural sufficient statistic therefore consists of the d elements of the vector $\mathbf{z}$ plus the d^2 elements of the vector $\mathbf{z}\mathbf{z}^T$. The natural parameter λ consists of the elements of the vector

$$\lambda_1 := \lambda_1(\mu, \Sigma) = \Sigma^{-1}\mu - \Sigma_0^{-1}\mu_0$$

plus the elements of the matrix

$$\lambda_2 := \lambda_2(\Sigma) = \frac{1}{2}(\Sigma_0^{-1} - \Sigma^{-1}) .$$

Note that since we have assumed Σ is positive definite, we are implicitly assuming that the matrix λ_2 is such that the determinant of $\frac{1}{2}\Sigma_0^{-1} - \lambda_2$ is strictly positive. The natural parameter space is therefore unrestricted for λ_1, but restricted (to matrices such that the indicated determinant is strictly positive) for λ_2.

The above relations can be inverted to write μ and Σ in terms of λ_1 and λ_2, indeed

$$\Sigma = \Sigma(\lambda_2) = (\Sigma_0^{-1} - 2\lambda_2)^{-1}$$

and

$$\mu = \mu(\lambda_1, \lambda_2) = (\Sigma_0^{-1} - 2\lambda_2)^{-1}(\lambda_1 + \Sigma_0^{-1}\mu_0) \, .$$

The cgf of the natural sufficient statistic is

$$K(\lambda) = K(\lambda_1, \lambda_2) = \hat{K}(\mu_{\lambda_1,\lambda_2}, \Sigma_{\lambda_2}) = \frac{1}{2}\mu_{\lambda_1,\lambda_2}^T \Sigma_{\lambda_2}^{-1} \mu_{\lambda_1,\lambda_2} - \frac{1}{2}\mu_0^T \Sigma_0^{-1}\mu_0 + \frac{1}{2}\log(\det(\Sigma_{\lambda_2})) - \frac{1}{2}\log(\det(\Sigma_0))$$

It is now clear that $K(\lambda)$ is well defined if and only if the determinant of $\Sigma(\lambda_2)$ is strictly positive, which we have implicitly assumed to be the case since we have insisted Σ be positive definite. It is also clear that $K(-\lambda)$ is well-defined if and only if the determinants $\Sigma(-\lambda_2)$ is strictly positive, which will occur if and only if the determinant of $(2\Sigma_0^{-1} - \Sigma^{-1})$ is strictly positive.

Remark A1. *Suppose that f and f_λ are Gaussian densities with respective positive definite covariance matrices Σ_0 and Σ. Further suppose that $\mathbf{Z} \sim f_\lambda$. Then the variance of $f(\mathbf{Z})/f_\lambda(\mathbf{Z})$ is finite if and only if* $\det(2\Sigma_0^{-1} - \Sigma^{-1}) > 0$.

In the one-dimensional case $d = 1$ we write $\mathbf{Z} = Z$. The condition in Remark A1 is satisfied whenever $\sigma^2 > \sigma_0^2/2$. In other words, if the variance of the IS distribution is too small, relative to actual variance of Z, then the IS weight will have infinite variance.

Appendix B.2. Chi-Square Family

In preparation for the multivariate t family, we first consider the chi-square family. Suppose that Z follows a chi-square distribution with ν_0 degrees of freedom, and that the goal is to allow Z to have arbitrary degrees of freedom $\nu > 0$ under the IS density. In order to identify the natural sufficient statistic $S(z)$ and natural parameter $\lambda = \lambda(\nu)$, we let $f(z)$ denote the chi-square density with ν_0 degrees of freedom and $f_\lambda(z)$ the chi-square density with ν degrees of freedom. Then

$$\frac{f_\lambda(z)}{f(z)} = \exp\left(\left(\frac{\nu - \nu_0}{2}\right)\log(z) - \left[\frac{\nu - \nu_0}{2}\log(2) + \log\left(\Gamma\left(\frac{\nu}{2}\right)\right) - \log\left(\Gamma\left(\frac{\nu_0}{2}\right)\right)\right]\right)$$

from which we see that $S(z) = \log(z)$ and $\lambda = \lambda(\nu) = (\nu - \nu_0)/2$. In addition we see that the cgf of $S(z)$ is

$$K(\lambda) = \lambda\log(2) + \log\left(\Gamma\left(\lambda + \nu_0/2\right)\right) - \log(\Gamma(\nu_0/2)) \, .$$

In order that $K(\lambda)$ be will defined, we require $\nu > 0$, which is obvious. In order that $K(-\lambda)$ is well-defined we require $-\lambda + \frac{\nu_0}{2}$ be positive, which in turn requires $\nu < 2\nu_0$. In other words, if the IS degrees of freedom are more than twice the actual degrees of freedom, then the IS weight will have infinite variance.

Appendix B.3. t Family

The t family is not a regular exponential family, so it does not fit directly into the framework discussed in Section 2.2. That being said, a multivariate t vector can be constructed from a Gaussian vector and an independent chi-square variable. Indeed if $\hat{\mathbf{Z}}$ is Gaussian with mean zero and covariance matrix Σ_0, and R is chi-square with ν_0 degrees of freedom (independent of $\hat{\mathbf{Z}}$), then

$$\mathbf{Z} = \mu_0 + \sqrt{\frac{\nu_0}{R}} \cdot \hat{\mathbf{Z}} \, , \tag{A5}$$

is multivariate t with ν_0 degrees of freedom, mean μ_0 and covariance matrix $\frac{\nu_0}{\nu_0 - 2}\Sigma_0$.

In the case that $\mathbf{Z}$ is multivariate t, then, we can take our systematic risk factors to be the components of $(\hat{\mathbf{Z}}, R)$. In this case the joint density of the systematic risk factors can be embedded into the parametric family

$$f_{\lambda,\eta}(\hat{\mathbf{z}}, r) := \exp(\lambda^T S(\hat{\mathbf{z}}) - K(\lambda)) \cdot \exp(\eta^T T(r) - L(\eta)) \cdot f(\hat{\mathbf{z}}) \cdot g(r) \,, \tag{A6}$$

where λ is and S are the natural parameter and sufficient statistic for the Gaussian family, η and L are those for the chi-square family, and f and g are the Gaussian and chi-square densities.

References

Bickel, Peter J., and Kjell A. Doksum. 2001. *Mathematical Statistics: Basic Ideas and Selected Topics*, 2nd ed. Upper Saddle River: Prentice Hall, Volume 1.

Chan, Joshua C.C., and Dirk P. Kroese. 2010. Efficient estimation of large portfolio loss probabilities in t-copula models. *European Journal of Operational Research* 205: 361–67.

Chatterjee, Sourav, and Persi Diaconis. 2018. The sample size required in importance sampling. *Annals of Applied Probability* 28: 1099–135. [CrossRef]

de Wit, Tim. 2016. Collateral Damage—Creating a Credit Loss Model Incorporating a Dependency between Defaults and LGDs. Master's thesis, University of Twente, Enschede, The Netherlands.

Deng, Shaojie, Kay Giesecke, and Tze Leung Lai. 2012. Sequential importance sampling and resampling for dynamic portfolio credit risk. *Operations Research* 60: 78–91. [CrossRef]

Eckert, Johanna, Kevin Jakob, and Matthias Fischer. 2016. A credit portfolio framework under dependent risk parameters PD, LGD and EAD. *Journal of Credit Risk* 12: 97–119. [CrossRef]

Frye, Jon. 2000. Collateral damage. *Risk* 13: 91-94.

Frye, Jon, and Michael Jacobs Jr. 2012. Credit loss and systematic loss given default. *Journal of Credit Risk* 8: 109–140. [CrossRef]

Glasserman, Paul, and Jingyi Li. 2005. Importance sampling for portfolio credit risk. *Management Science* 51: 1643–56. [CrossRef]

Ionides, Edward L. 2008. Truncated importance sampling. *Journal of Computational and Graphical Statistics* 17: 295–311. [CrossRef]

Jeon, Jong-June, Sunggon Kim, and Yonghee Lee. 2017. Portfolio credit risk model with extremal dependence of defaults and random recovery. *Journal of Credit Risk* 13: 1–31. [CrossRef]

Kupiec, Paul H. 2008. A generalized single common factor model of portfolio credit risk. *Journal of Derivatives* 15: 25–40. [CrossRef]

Miu, Peter, and Bogie Ozdemir. 2006. Basel requirements of downturn loss given default: Modeling and estimating probability of default and loss given default correlations. *Journal of Credit Risk* 2: 43–68. [CrossRef]

Pykhtin, Michael. 2003. Unexpected recovery risk. *Risk* 16: 74–78.

Scott, Alexandre, and Adam Metzler. 2015. A general importance sampling algorithm for estimating portfolio loss probabilities in linear factor models. *Insurance: Mathematics and Economics* 64: 279–93.

Sen, Rahul. 2008. A multi-state Vasicek model for correlated default rate and loss severity. *Risk* 21: 94–100.

Witzany, Jiří. 2011. A Two-Factor Model for PD and LGD Correlation. Working Paper. Available online: http://dx.doi.org/10.2139/ssrn.1476305 (accessed on 9 March 2020).

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).