Approximating the Effects of Parameter Uncertainty on Value at Risk Estimates

Steven Major
SAS Institute, Cary North Carolina
e-mail: Steven.Major@sas.com

Jacques Rioux
SAS Institute, Cary North Carolina
e-mail: Jacques.Rioux@sas.com@sas.com

Donald Erdman
SAS Institute, Cary North Carolina
e-mail: Donald.Erdman@sas.com

10/25/2010

Abstract

This article examines how parameter uncertainty leads to uncertainty in Value at Risk (VaR) estimates. Lacking a closed form solution, we use a first order approximation of VaR to map parameter changes into changes in the VaR value. When the asymptotic distribution of parameter estimates is normal, this implies a normal approximation of the confidence interval about VaR. We describe this approximation in detail and explore some extensions to risk factor modeling.

1 Introduction

To comply with Basel II [BIS (2005)] regulatory requirements for managing operational risks, it is increasingly common for banks to adopt the Loss Distribution Approach (LDA). In this framework, the number and size of losses are modeled across businesss lines and event types according to

$$S = \sum_{i=1}^{N} X_i \tag{1}$$

where S is the aggregate loss, N is the random variable modeling the number of loss events, and the X_i's are independent and idententically distributed random variables modeling the severity of each event. The level α Value at Risk, VaR_α, associated with S is the aggregate loss that is exceded with probability $(1 - \alpha)$. Since decisions based on reported VaR values often involve the holding of large amounts of capital in reserve, the quality of VaR estimates is of great interest.

There are several sources of potential error in VaR estimates. Two of the more commonly discussed with respect to the LDA are model risk and parameter risk [Mignola and Ugoccioni (2006)]. Model risk addresses the possibility that the model selected to fit the data wasn't appropriate. Parameter risk addresses the fact that, given the correct model, the estimated parameters will be incorrect due to limited sample size and the inherent randomness of the data. The issue of parameter risk is particularly worrisome in the case of operational risk management as the quantity of data is often limited. Quantifying the way this uncertainty in parameter estimates translates into uncertainty in VaR estimates can be challenging, as a closed form solution for VaR as a function of the model parameters is generally unavailable.

Previous contributions to parameter risk have used a Bayesian approach [Carvalho et al. (2008)] or used extreme value theory (EVT) [Embrechts et al. (2003)]. The bayesian approach is simulation based and therefore (as we discuss later) can be very time consuming. The approach taken by Embrechts et al. (2003) is ideal when losses are modeled with an extreme value distribution as it allows for the computation of the confidence bounds about VaR implied by the distribution of parameter estimates, but is not applicable to non EVT models.

We would like to be able to produce confidence bounds for VaR estimates for a wide class of models. To avoid resorting to simulation based methods, which can be time consuming, we resort to a first order approximation of VaR that can be computed quickly. This work is a reframing of the work in Erdman et al. (2010). We focus here on the computational process for approximating confidence intervals about VaR as opposed to the general class of distortion risk measures. Also, we emphasize the use of the discrete Fourier transform over Panjer's recursive method. We show that VaR is a linear function of any scale parameter so that the linear relationship perfectly models the effect of the scale parameter on VaR. We use this fact to suggest a more generalized model for S which allows for the incorporation of risk factors into the modeling process.

In the next section we describe how one might compute confidence intervals about VaR with a simulation based approach with the intent of demonstrating the limitations of this method. In section 3 we outline how a first order approximation of VaR can be used to approximate the confidence interval and provide the computational details in section 4. We provide details of the process for a specific example in section 5. We then show how the linear behavior of VaR with respect to a scale parameter can be used to incorporate risk factors into the modeling process and how parameter risk estimation is performed in this framework in section 6.

2 The Monte Carlo approach to parameter risk

A firm observes loss amounts over a time period and uses the data to estimate a specific form of (1) by selecting models to describe X and N with parameter vectors θ_X and θ_N respectively. For simplicity, we combine θ_X and θ_N into the parameter vector $\theta_S \in \mathbb{R}^m$ which completely describes the aggregate distribution. We assume that the parameter estimation method yields estimates that are asymptotically normal and unbiased. For example, the parameters can be estimated using the maximum likelihood method. Thus,

$$[\hat{\theta_S}]^T \sim N([\theta_S]^T, \Sigma_{\theta_S}) \tag{2}$$

From here the firm computes its VaR one of many ways. One approach is to discretize X and use the discrete Fourier transform [Press et al. (1992)] or Panjer's recursive formula [Panjer (1981)] to obtain a discretized approximation of S. Alternatively, a simulation based approach can be used where each draw of S is arrived at by drawing a value n from the distribution of N and then drawing n independent observations from X. Regardless of the approach, the end result is a value $\text{VaR}(\hat{\theta_S})$ that represents a potentially large amount of money that will be held in reserve against potential losses.

Note that (2) implies that, while $\hat{\theta_S}$ is the most likely value of θ_S given the data, there were many other choices that could also have produced the data with very similar likelihoods. For each one of these probable choices of θ is a corresponding VaR value. So the confidence in the VaR estimate depends on the distribution of VaR estimates implied by (2). To obtain this distribution, we can turn to simulation. We first simulate a draw from $\hat{\theta_S}$ and use this value of θ_S to simulate from S repeatedly to compute VaR. This process is repeated until a suitable number of VaR values have been produced. The simulated VaR values are finally sorted to form an empirical distribution from which confidence intervals may be determined.

If computation time is not a concern, then the approach outlined above works very well. However, the computation time increases linearly with the parameter space, which increases exponentially with the number of parameters. This is the 'curse of dimensionality' [Bellman (Princeton University Press)] and it can quickly make simulation based approaches impractical. For example, if N and X have one parameter each, then the parameter space is in $\mathbb{R}^2$ and, depending on the variance in the estimates, a reasonable number of parameter draws could be as small as 1,000. This means that VaR must be computed 1,000 times. Adding one more parameter to the models of N and X means that the parameter space is now in $\mathbb{R}^4$, so that to simulate over a similar grid would require $1,000^2 = 10^6$ VaR calculations. As the number of parameters increases, the problem quickly becomes unmanageable, especially if one hopes to compute VaR values across business lines and event types.

3 Approximating parameter risk

The main difficulty in the approach outlined above is the fact that, given a parameter vector θ_S, there is no closed form solution for the corresponding VaR value. Ideally, the VaR would depend on θ_S in such a way that the distribution of VaR given the distribution of θ_S was known, so that a confidence interval could be written down without a need for simulation. A solution is to use a linear approximation of VaR

$$\text{VaR}_1(\theta_S) = \text{VaR}(\hat{\theta}_S) + D^T(\theta_S - \hat{\theta_S}) \tag{3}$$

where D is the gradient vector

$$D = \left[\frac{\partial \text{VaR}}{\theta_{S1}}, \cdots, \frac{\partial \text{VaR}}{\theta_{Sm}} \right]^T .$$

Since the first order approximation of VaR is just an affine transformation of the multivariate normal random vector $\hat{\theta}_S$, its distribution is known

$$\text{VaR}_1(\theta_S) \sim N\left(\text{VaR}(\hat{\theta_S}), D^T \Sigma_{\theta_S} D\right). \tag{4}$$

Thus the first order approximation of VaR is asymptotically normal with mean $\text{VaR}(\hat{\theta})$ and covariance matrix $D^T\Sigma_{\theta_S}D$ and a 95% confidence interval would be computed as

$$\text{VaR}(\hat{\theta_S}) \pm 1.96 \cdot \sqrt{D^T \Sigma_{\theta_S} D}. \tag{5}$$

Once one has a specification for (3), an approximate distribution of VaR is immediate. The values of $\hat{\theta}_S$, Σ_{θ_S}, and $\text{VaR}(\hat{\theta_S})$ are immediate biproducts of the usual VaR computation process. The only missing piece is the gradient vector D.

3.1 Derivative of VaR with respect to θ_S

If for $R^* \in [0, +\infty)$ and $\theta_S^* \in \mathbb{R}^m$ we have that $\alpha = F_S(R^*, \theta_S^*)$, then by definition, R^* is the level α VaR corresponding to the parameter vector θ_S^*. If we then fix α, the implicit function theorem [Marsden and Hoffman (1993)] provides for the existence of the function $\text{VaR}(\theta_S)$ so that $F_S(\text{VaR}(\theta_S), \theta_S) = \alpha$ in a neighborhood about θ_S^*, provided $\frac{\partial F_S}{\partial s}\Big|_{s=R^*} \neq 0$. Note that this condition is equivalent to the condition that S has support almost everywhere in a neighborhood about R^*. Taking the derivative with respect to the i_{th} element of θ_S on both sides yields

$$\frac{\partial}{\partial \theta_{Si}} F_S(\text{VaR}(\theta_S), \theta_S) = \frac{\partial \alpha}{\partial \theta_{Si}} = 0 \qquad \forall \theta_{Si}$$

which implies that

$$\frac{\partial \text{VaR}(\theta_S)}{\partial \theta_{Si}} = -\frac{1}{\dfrac{\partial F_S(s, \theta_S)}{\partial s}} \cdot \frac{\partial F_S(s, \theta_S)}{\partial \theta_{Si}} . \tag{6}$$

At first glance we are not in any better shape as we now have the two unknown quantities

$$\left.\frac{\partial F_S(s,\theta_S)}{\partial s}\right|_{s=R^*,\theta_S=\hat{\theta}_S} \tag{7}$$

and

$$\left.\frac{\partial F_S(s,\theta_S)}{\partial \theta_{Si}}\right|_{s=R^*,\theta_S=\hat{\theta}_S} , \tag{8}$$

which are both dependent on the aggregate loss distribution, which we don't have a closed form solution to. A discrete approximation of S provides a simple approximation for each.

4 Aggregate distribution approximation

From (1) we can express F_S using convolutions as

$$\begin{aligned} F_S(s) = P(S \le s) &= \sum_{i=0}^{\infty} P(N=i) P\Big(\sum_{j=1}^{i} X_j \le s\Big) \\ &= \sum_{i=0}^{\infty} f_N(i) F_X^{*(i)}(s) \end{aligned} \tag{9}$$

where $F_X^{*(i)}(s)$ is the i^{th} convolution of F evaluated at s;

$$F_X^{*(i)}(s) = \int_0^s F_X^{*(i-1)}(s-u)\ \mathrm{d}F_X(u) \ \ ; \qquad F_X^{*(0)}(s) = 1 \ . \tag{10}$$

If the distribution of X is discrete, we can evaluate (9) recursively using the discrete Fourier transform (DFT). Since X is generally modeled with a continuous distribution, a discrete approximation of X must first be computed. The discretization must be done over a lattice of width h, which will be the same lattice over which the aggregate distribution is discretized. The trick is to choose an h that is small enough to capture the behavior of X, but not so small that the number of discretization points for S becomes unmanageable. For a discussion of discretization methods, see Embrechts and Frei (2008).

Let $\tilde{X}$ and $\tilde{S}$ denote the discretized representations of X and S respectively, with $f_k^{\tilde{X}} = P(\tilde{X} = h{\cdot}k)$ and $f_k^{\tilde{S}} = P(\tilde{S} = h{\cdot}k)$. The rounding method, which assigns to each point x the mass between $x - h/2$ and $x + h/2$ is ideal for our purposes as it allows for an easy approximation of (7). The weight on the point $j * h$ is

$$f_j^{\tilde{X}} = F_X\left(jh + \frac{h}{2}\right) - F_X\left(jh - \frac{h}{2}\right) \quad h = 0, 1, 2, ...$$

Because the DFT is used for the computation of $\tilde{S}$ it is necessary to have the number of discretization points be a power of 2.

For completeness, we proceed with a description of the DFT in the next subsection. Readers familiar with the DFT can skip this portion. The details provided are then used to show how one computes (7) and (8) when F_S is approximated with the DFT.

4.1 Discrete Fourier transform for compound distributions

We want to compute the probability mass function (PMF) of the compound random variable

$$S = \sum_{i=1}^{N} X_i.$$

We assume that X is discretized and drop the '$\sim$' notation for convenience. Therefore X can take on the values $\{X_0, X_1, ..., X_{M-1}\}$ where M is a power of 2. Since S, X, and N are all discrete, let their corresponding PMFs be f_S, f_X, and f_N respectively.

The strategy is to express the PMF of S in such a way that allows us to make use of some powerful machinery so that computation is done quickly. This is done by looking at the characteristic function of S and then linking this to the discrete Fourier transform. In the process of doing this, we will need the probability generating function. Let $\mathcal{P}_N$ be the probability generating function (PGF) of N. The PGF of a random variable Y is the power series representation of Y evaluated at a complex point z

$$\mathcal{P}_Y(z) = E[z^Y] = \sum_{y=0}^{\infty} f_Y(y) \cdot z^y.$$

The characteristic function of a discrete random variable Y at a real value u is defined to be

$$\widehat{f_Y}(u) = E[e^{iuY}].$$

When Y is discrete, this is also

$$\widehat{f_Y}(u) = \sum_{k=0}^{\infty} f_Y(k)\cdot e^{iuk}.$$

So the characteristic function of S is

$$\begin{aligned}\widehat{f_S}(u) &= E_S[e^{iuS}]\\ &= \mathcal{P}_N\big(\widehat{f_X}(u)\big).\end{aligned}$$

The next step is to show how to interpret this relationship in terms of the discrete Fourier Transform. Consider the sequence of values $\{f_{X0}, f_{X1}, ..., f_{XM-1}\}$. The DFT creates a new sequence, $\widehat{f_X} = \{\widehat{f_{X0}}, \widehat{f_{X1}}, ..., \widehat{f_{XM-1}}\}$, whose j^{th} element is

$$\widehat{f_{Xj}} = \sum_{k=0}^{M-1} f_{Xk} e^{-2\pi ijk/M}. \tag{11}$$

The j^{th} element of the original sequence can be recovered from the transformed sequence, $\{\widehat{f_{Xj}}\}_{j=0}^{M-1}$, as

$$f_{Xj} = \frac{1}{M}\sum_{k=0}^{M-1} \widehat{f_{Xk}} e^{2\pi ijk/M}. \tag{12}$$

The DFT is the same as the characteristic function evaluated at the point $u = -2\pi j/M$. The j^{th} element of the transformed PMF of S is then

$$\widehat{f_{Sj}} = \mathcal{P}_N\big(\widehat{f_{Xj}}\big).$$

So, if we can quickly compute the DFT of X and the PGF of N then we have a way to quickly compute the transformed PMF of S. The fast Fourier transform will compute the DFT and its inverse efficiently. Computation time increases with $Mlog(M)$, where M is the number of discretization points [Press et al. (1992)]. The PMF of S is then recovered by applying the inverse transform given by equation (12).

To summarize, use the following notation:

$$\begin{aligned} T(f) &= \left\{\widehat{f}_0, \widehat{f}_1, \cdots, \widehat{f}_{M-1}\right\}\\ T^{-1}(\widehat{f}) &= \{f_0, f_1, \cdots, f_{M-1}\}\\ \mathcal{P}(f) &= \{\mathcal{P}(f_0), \mathcal{P}(f_1), \cdots, \mathcal{P}(f_{M-1})\}. \end{aligned}$$

The PMF of S is then

$$f_S = T^{-1}\Big(\mathcal{P}_N\big(T(f_X)\big)\Big). \tag{13}$$

4.2 Aggregate CDF derivatives

Approximations for (7) and (8) can both be derived from (13). For (7), we use the second order approximation

$$\left.\frac{\partial F_S(s;\theta_S)}{\partial s}\right|_{s=R^*,\theta_S=\hat{\theta}_S} \approx \frac{F_S(r^*h+\frac{h}{2};\hat{\theta}_S) - F_S(r^*h-\frac{h}{2};\hat{\theta}_S)}{h} = \frac{f_{\tilde{S}_{r^*}}}{h} \tag{14}$$

where r^* is an integer chosen so that $r^*h \approx \mathrm{VaR}(\hat{\theta}_S)$. Note that this approximation depends on the use of the rounding method for discretization.

From the discrete approximation of S, computation of (8) reduces to computing the derivative of the PMF at each point with respect to the model parameters since

$$F_S(\mathrm{VaR}) \approx \sum_{j=0}^{r*} f_{S_j},$$

which implies that

$$\left.\frac{\partial F_S(s,\theta_S)}{\partial \theta_{Si}}\right|_{s=R^*,\theta_S=\hat{\theta}_S} \approx \sum_{j=0}^{r*} \left.\frac{\partial f_{S_j}}{\partial \theta_S}\right|_{\theta_S=\hat{\theta}_S}.$$

Here we break into 2 cases θ is a severity parameter and θ is a frequency parameter. When θ is a severity parameter, $\mathcal{P}$ does not depend on θ and the differentiation is evaluated via the chain rule

$$\begin{aligned}\frac{\partial f_{Si}}{\partial \theta} &= T^{-1}\Big(\frac{\partial}{\partial \theta}\big[\mathcal{P}_N(T(f_X))\big]\Big)_i, \\ &= T^{-1}\left(\Big\{\mathcal{P}'_N\big(T(f_X)\big)_j \cdot T\Big(\frac{\partial f_X}{\partial \theta}\Big)_j\Big\}_{j=0}^{M-1}\right)_i\end{aligned} \tag{15}$$

where $\mathcal{P}'_N$ is the complex derivative of $\mathcal{P}_N$.

When θ is a frequency parameter, the differentiation is applied to the operator $\mathcal{P}_N$ itself

$$\frac{\partial f_{Si}}{\partial \theta} = T^{-1}\left(\frac{\partial \mathcal{P}_N}{\partial \theta}\Big(T(f_X)\Big)\right)_i. \tag{16}$$

4.3 Linearized VaR computation with discretized severity

Combining (14), (15), and (16), we write (6) as

$$\frac{\partial \mathrm{VaR}(\theta_S)}{\partial \theta_{Si}} \approx \tag{17}$$

$$\begin{cases} -\frac{h}{f^{\tilde{S}}_{r^*}} \left[\sum_{j=o}^{r^*} T^{-1} \left(\left\{ \mathcal{P}'_N \big(T(f_X)\big)_j \cdot T\Big(\frac{\partial f_X}{\partial \theta}\Big)_j \right\}_{j=0}^{M-1} \right)_i \right] & \theta_{Si} \in \theta_X \\ -\frac{h}{f^{\tilde{S}}_{r^*}} \left[\sum_{j=o}^{r^*} T^{-1} \left(\frac{\partial \mathcal{P}_N}{\partial \theta} \Big(T(f_X)\Big) \right)_i \right] & \theta_{Si} \in \theta_N \end{cases}$$

Computation is straightforward and requires very little time. From a programming standpoint, the requirements are:

- a function to compute the derivative of the severity CDF,
- an implementation of the discrete Fourier transform,
- a simple discretization routine,
- a function to compute the PGF of the frequency distribution,
- a function to compute the complex derivatives of the PGF.

5 Example

Here we present some details for the case when $X \sim$ Lognormal(μ, σ) and $N \sim$ Poisson(λ). We compare the approximate confidence interval about VaR with one obtained from simulation. The estimated parameters are

$$\hat{\theta} = \left[\hat{\lambda}, \hat{\mu}, \hat{\sigma}\right]^T = [10, 6.56, 0.69]^T$$

from 10 frequency observations and 100 severity observations. The parameter covariance matrix is

$$\sum = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.0048 & 0 \\ 0 & 0 & 0.0045 \end{pmatrix}$$

and the 99% VaR is estimated to be 19,055.

The logmormal cdf and it's derivatives with respect to its parameters are:

$$\begin{aligned} F_X =& \frac{1}{2} + \frac{1}{\sqrt{\pi}} \int_0^y e^{-t} \mathrm{d}t \\ \frac{\partial F_X}{\partial \mu}(x,\mu,\sigma) =& -\frac{1}{\sqrt{2\pi}\sigma} e^{-y^2} \qquad \text{where} \qquad y = \frac{ln\,(x) - \mu}{\sigma\sqrt{2}} \\ \frac{\partial F_X}{\partial \sigma}(x,\mu,\sigma) =& -\frac{1}{\sqrt{\pi}\sigma} e^{-y^2} y \qquad . \end{aligned}$$

We refer to these functions as ‘FX’, ‘dFX1’, and ‘dFX2’, respectively.

The PGF for a Poisson random variable is

$$\mathcal{P}_N(z,\lambda) = e^{\lambda(z-1)},$$

where z is a complex argument. The derivative with respect to z is

$$\mathcal{P}'_N(z,\lambda) = \lambda e^{\lambda(z-1)} = \lambda \mathcal{P}_N(z)$$

and the derivative with respect to λ is

$$\frac{\partial \mathcal{P}_N}{\partial \lambda}(z,\lambda) = (z-1)e^{\lambda(z-1)}.$$

We refer to these functions as ‘PN’,‘dPNz’ and ‘dPN1’ respectively.

The only remaining function we need is an implementation of the discrete Fourier transform and it's inverse, which we will refer to as ‘T’ and ‘T^{-1}’. An excellent resource for implementing the fast Fourier transform is Press et al. (1992).

We start by discretizing the severity distribution. If we use 2^{13} dscretization points and discretize up until we reach the VaR value, then the discretization interval width is roughly 2.3, which allows for a high degree of precision considering the severity mean is close to 900. However, we need to consider the aliasing error associated with the DFT. Essentially, any mass not captured by the discretization is distributed to the body of the distribution by the algorithm. One method of handling this is to employ exponential tilting. Since VaR is defined as a point beyond which the probability distribution of S has very little mass, we find it sufficient to discretize over a larger interval, possibly with more discretization points. In this example, we can use the 2^{13} points to discretize up to the point $2\cdot$ VaR and maintain an interval width of $h \approx 4.6$, which is still reasonable. With this scheme, the amount of mass which is aliased is very small and given by $1 - F_S(2\cdot \text{VaR})$.

The mass at each point of the discretized distribution of X is

$$f_j^{\tilde{X}} = FX(jh + \frac{h}{2}, 6.6, 0.69) - FX(jh + \frac{h}{2}, 6.6, 0.69).$$

The discretized severity, $\tilde{X}$, is an array of 2^{13} elements. From this array we now need 2 more, the arrays of derivatives with respect to μ and σ, $\text{dF1}(\tilde{X})$ and $\text{dF2}(\tilde{X})$. They are computed as

$$\begin{aligned} \text{dF1}(\tilde{X})_j &= \text{dF1}(f_j^{\tilde{X}}, 6.6, 0.69) \\ \text{dF2}(\tilde{X})_j &= \text{dF2}(f_j^{\tilde{X}}, 6.6, 0.69). \end{aligned}$$

We then apply the DFT to the 3 arrays to obtain $T(\tilde{X})$, $T\big(\text{dF1}(\tilde{X})\big)$, and $T\big(\text{dF2}(\tilde{X})\big)$.

From $T(\tilde{X})$, we next apply each of the PGF derivatives to each point of $T(\tilde{X})$ and obtain the arrays $dPN1\big(T(\tilde{X})\big)$ and $dPNz\big(T(\tilde{X})\big)$.

Each of the arrays $T\big(\text{dF1}(\tilde{X})\big)$ and $T\big(\text{dF2}(\tilde{X})\big)$ is now multiplied element by element with the array $dPNz\big(T(\tilde{X})\big)$ to produce $dPNz\big(T(\tilde{X})\big)\cdot T\big(\text{dF1}(\tilde{X})\big)$ and $dPNz\big(T(\tilde{X})\big)\cdot T\big(\text{dF2}(\tilde{X})\big)$.

At this point, we are nearly finished. The derivatives of the discretized aggregate PMF are computed as

$$\begin{aligned} \frac{\partial f^{\tilde{S}}}{\partial \lambda} &= T^{-1}\Big(dPN1\big(T(\tilde{X})\big)\Big), \\ \frac{\partial f^{\tilde{S}}}{\partial \mu} &= T^{-1}\Big(dPNz\big(T(\tilde{X})\big)\cdot T\big(\text{dF1}(\tilde{X})\big)\Big), \\ \frac{\partial f^{\tilde{S}}}{\partial \sigma} &= T^{-1}\Big(dPNz\big(T(\tilde{X})\big)\cdot T\big(\text{dF2}(\tilde{X})\big)\Big), \end{aligned}$$

from which we compute the derivatives of the aggregate CDF with respect to the parameters at VaR as

$$\begin{aligned}
\frac{\partial F_S}{\partial \lambda} &= \sum_{j=1}^{r^*}\Big(\frac{\partial f^{\tilde{S}}}{\partial \lambda}\Big)_j, \\
\frac{\partial F_S}{\partial \mu} &= \sum_{j=1}^{r^*}\Big(\frac{\partial f^{\tilde{S}}}{\partial \mu}\Big)_j, \\
\frac{\partial F_S}{\partial \sigma} &= \sum_{j=1}^{r^*}\Big(\frac{\partial f^{\tilde{S}}}{\partial \sigma}\Big)_j,
\end{aligned}$$

where r^* is the index associated with the VaR value. In this example, r^* is 2^{13}.

The discretized aggregate pmf is next computed as

$$f^{\tilde{S}} = T^{-1}\Big(PN\big(T(\tilde{X})\big)\Big)$$

and the derivative of VaR with respect to the parameters is

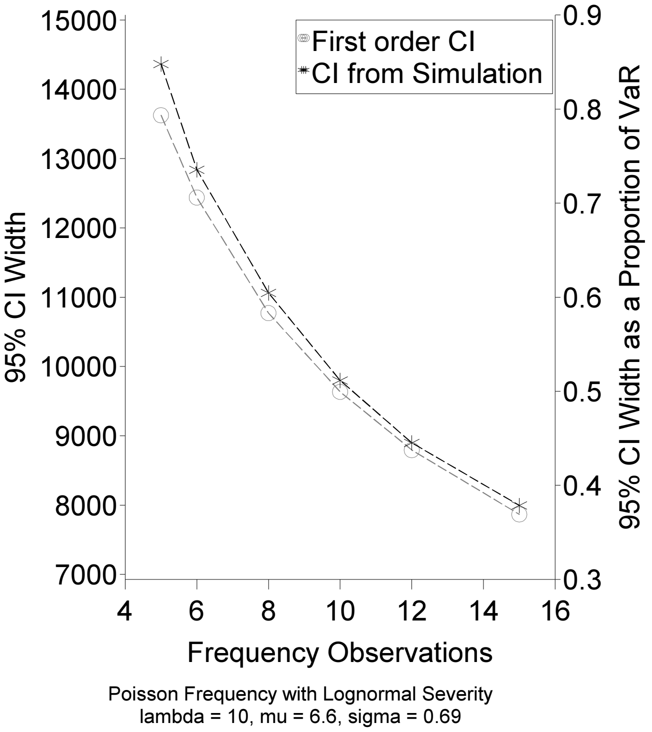
$$D^T = [\frac{-h}{f^{\tilde{S}}_{r^*}}\frac{\partial F_S}{\partial \lambda}, \frac{-h}{f^{\tilde{S}}_{r^*}}\frac{\partial F_S}{\partial \mu}, \frac{-h}{f^{\tilde{S}}_{r^*}}\frac{\partial F_S}{\partial \sigma}] \quad = [1323.1, 19,029.4, 23,502.4]$$

The variance in VaR is $D^T \Sigma D \approx 6,040,485.2$, which implies a 95% confidence interval width of 9,634.3.

As a point of comparison, the confidence interval can be determined by simulation. We used 10^4 parameter draws, computing VaR with 10^5 draws from N for each set of parameters. The 95% confidence interval width computed this way is 9,798.8. The simulations took nearly 1 hour to perform, while the approximate confidence interval was computed in under 5 seconds. Using the model parameters in the example, we allowed the number of frequency observations to vary and compared the approximated and simulated confidence intervals. The results are shown in figure 1.

Looking at figure 1, it is apparent that even for a seemingly reasonable number of observations, estimations of VaR have relatively wide confidence bounds. For example, a firm with 10 years of data would need to hold more than 150% of it's estimated VaR in reserve in order to reasonably believe that there is sufficient coverage against outlier events. With 4 years of data, the 95% interval about VaR is nearly as large as the VaR value itself. In this case, the validity of risk assesments using the LDA are questionable.

Figure 1: Actual confidence interval obtained by simulation vs. first order approximation.



6 Incorporating risk factors into the model through a scale parameter

Besides loss data, firms have access to a wide variety of statistics like employee turnover rates, customer complaints per quarter, etc. It is likely that at least some of this information may affect the operational loss process. We refer to such variables as risk factors and are interested in how they can be used to incorporate knowledge about the state of affairs in various parts of an organization into the modeling process.

Risk factor modeling is common practice when managing financial risks

[Duffie and Pan (1997)]. In this context, they are underlying market states that effect the value of instruments in a firm's portfolio. In an operational risk context, determining how VaR is affected by risk factors can aid in forecasting future losses and can also help determine where resources should be spent to most effectively reduce losses from operational events. Confidence intervals about VaR that incorporate the uncertainty in estimates of the effects of risk factors on losses can easilly be approximated using the methods in this paper.

In this section we show that when the severity distribution contains a scale parameter, θ, VaR is a linear function of θ. We then suggest a simple model that allows the risk factors to affect losses by acting on the scale parameter.

6.1 Scale parameters and VaR

Suppose θ is a scale parameter for X, i.e. $F_X(x;\theta) = F_X(x/\theta;1)$. The i^{th} convolution of X is given by (10). We show that θ is a scale parameter for $F_X^{(i)*}, i \in \mathbb{N}$ by mathematical induction. By definition, θ is a scale parameter for $F_X^{(1)*}$. Now suppose that θ is a scale parameter for $F_X^{(i-1)*}$

$$
\begin{aligned}
F_X^{*(i)}(s;\theta) &= \int\limits_0^s F_X^{*(i-1)}(s-u;\theta)dF(u) \\
&= \int\limits_0^s F_X^{*(i-1)}(s/\theta - u/\theta;1)dF(u/\theta) \\
&= \int\limits_0^{s/\theta} F_X^{*(i-1)}(s/\theta - u;1)dF(u) \\
&= F_X^{*(i)}(s/\theta;1).
\end{aligned}
$$

Then θ is a scale parameter for $F_X^{(i)*}$. From (9) it is clear that θ is also a scale parameter for S since

$$\begin{aligned} F_S(s;\theta) &= \sum_{i=0}^{\infty} f_N(i) F_X^{*(i)}(s;\theta) \\ &= \sum_{i=0}^{\infty} f_N(i) F_X^{*(i)}(s/\theta;1) \\ &= F_S(s/\theta;1). \end{aligned}$$

This means $\frac{\partial \mathrm{VaR}_\alpha}{\partial\theta}$ is easy to compute. The edfinition of VaR implies that $F_S(\mathrm{VaR};\theta) = \alpha$, which can be written as $F_S(\frac{F_S^{-1}(\alpha;\theta)}{\theta};1) = \alpha$. If we fix α and apply the implicit function theorem, we can take the total derivative with respect to the first argument of F_S and write

$$\frac{\partial F_S}{\partial s}(\mathrm{VaR};\theta)\big[\frac{\frac{\partial F_S^{-1}}{\partial\theta}(\alpha;\theta)}{\theta} - \frac{F_S^{-1}(\alpha;\theta)}{\theta^2}\big] = 0,$$

which implies $\frac{\partial F_S^{-1}}{\partial\theta}(\alpha;\theta) = \frac{F_S^{-1}(\alpha;\theta)}{\theta}$ Therefore VaR is linear in θ, with slope given by

$$\frac{\partial \mathrm{VaR}}{\partial\theta} = \frac{\mathrm{VaR}}{\theta} \tag{18}$$

6.2 Modeling risk factors as regressors on a scale parameter

Suppose that the severity, X, is modeled so that it has scale parameter θ. The set of risk factors is $\{k_1, k_2, ..., k_n\}$. We assume that risk factors adjust the scale of losses, but not the shape of the loss distribution. Since the scale must be positive, a simple model has the log of the scale as a linear function of the risk factors $\theta = e^{\beta_0 + \sum \beta_i k_i}$.

While simple, this model has some very nice features. First, since VaR is linear in the scale parameter, it makes for an unambiguous interpretation of the model parameters; β_i is the relative increase in VaR caused by a marginal increase in the risk factor k_i. A second feature of the model is that the VaR can be easily recomputed as the risk factors change. This makes forecasting very simple. For example, suppose that the number of workers at a plant is modeled as a risk factor and the coefficient is estimated to be 0.001. Then an additional 10 workers would cause VaR to increase by a factor of $e^{0.01}$, which amounts to an approximate 1% increase.

The first order effect of a change in one of the risk factor coefficients on VaR follows directly from the chain rule and (18)

$$\frac{\partial \text{VaR}}{\partial \beta_i} = \frac{\partial \text{VaR}}{\partial \theta} \frac{\partial \theta}{\partial \beta_i} = k_i \cdot \text{VaR}.$$

Therefore, to approximate the confidence interval about VaR, we can proceed exactly as in (5).

We can also consider the impact on VaR of a change in the risk factors themselves. The derivative with respect to risk factor k_i is given by

$$\frac{\partial \text{VaR}}{\partial k_i} = \frac{\partial \text{VaR}}{\partial \theta} \frac{\partial \theta}{\partial k_i} = \beta_i \cdot \text{VaR}. \tag{19}$$

A firm interested in reducing its value at risk can use the derivatives in (19) to determine the best approach. For example, if risk factor k_i can be changed at a per unit cost of p_i, then a cost effective approach to reducing VaR would be to first invest in changing k_{j^*}, where $j^* = argmin_i \left\{ \frac{p_i}{\beta_i} \right\}$.

For a more robust analysis, we could consider the cost of VaR as the lost interest earnings from holding capital equivalent to VaR in reserve. If it is possible to alter the values of the risk factors as discussed above, then we are faced with a cost minimization problem. For example, suppose we find that the derivative of VaR with respect to the scale parameter θ is m and we expect to lose a rate of return r on capital held in reserve to cover operational risks. Then the cost of VaR is

$$r \cdot \text{VaR} = r \cdot m \cdot e^{\beta_0 + \sum_i \beta_i k_i}$$

Further suppose that we can reduce the value of k_i at a per unit cost of p_i and let c_i represent the level to which we choose to do so. The cost minimization problem is then

$$\underset{c_i}{min} \quad \text{COST} = r \cdot m \cdot e^{\beta_0 + \sum_i \beta_i (k_i - p_i c_i)} + \sum_i p_i c_i \qquad \text{s.t.} \quad c_i > 0 \forall i \tag{20}$$

In this example, provided there are no corner solutions, each risk factor is adjusted until the marginal benefit of further reduction is exactly offset by the marginal cost of further reduction.

7 Conclusion

We derived a first order approximation to VaR as a function of the model parameters and showed that it is computationally inexpensive and easy to implement. An immediate application of the first order approximation is the ability to approximate confidence intervals about VaR resulting from parameter uncertainty. Because we are using a linear approximation of the VaR functional, large amounts of variation in the parameter estimates can make the approximated confidence intervals about VaR very inaccurate. It should be noted, however, that the derivatives of VaR with respect to the model parameters are still accurate in this case. Also, when the parameter uncertainty is very large, the LDA methodology may not be appropriate.

The inclusion of risk factors is a common aspect of market risk management, but in terms of operational risk, what qualifies as a risk factor and how one might include them in the modeling process is not obvious. The linear relationship between VaR and a severity scale parameter provides an intuitive approach to this problem. Of course the inclusion of additional parameters means that for a given number of observations, the confidence in VaR estimates will be smaller.

The way confidence in VaR estimates might be used in practice depends on the way VaR is used by the firm. With regard to regulatory requirements, a firm may report drastically different VaR values period to period. In this case, the sensitivity of VaR calculations with respect to model parameters might be of interest to a regulator wondering why the capital charge is so variable. For internal purposes, one may use the assymptotic distribution of VaR to test for variation in VaR across business lines.

References

Richard Ernest Bellman. *Dynamic Programming.* 1957, Princeton University Press.

BIS. Basel II: International convergence of capital measurement and capital standards: a revised framework. *Bank for International Settlements (BIS)*, 2005. URL `www.bis.org`.

R. Carvalho, H. Mignon, and M. Paez. Dynamic bayesian models as an alternative to the estimation of operational risk measures. *The Journal of Operational Risk*, 3(1):25–50, 2008.

D. Duffie and J. Pan. An overview of value at risk. *The Journal of Derivatives*, 4(3):7–49, 1997.

P. Embrechts and M. Frei. Panjer recursion versus fft for compound distributions. *Mathematical Methods of Operations Research*, 69(3):497–508, 2008.

P. Embrechts, P. Furror, and R. Kauffman. Quantifying regulatory capital for operational risk. *Derivatives Use, Trading and Regulation*, 9(3):217–233, 2003.

D. Erdman, S. Major, and J. Rioux. Evaluation of parameter risk via first-order approximation of distortion risk measures. *Journal of Operational Risk*, 5(1):29–46, 2010.

Jerrold E. Marsden and Michael J. Hoffman. *Elementary Classical Analysis, second edition.* Macmillan, 1993.

G. Mignola and R. Ugoccioni. Sources of uncertainty in modeling operational risk losses. *The Journal of Operational Risk*, 1(2):33–51, 2006.

H. Panjer. Recursive evaluation of a family of compound distributions. *ASTIN Bulletin*, 12(2):88–101, 1981.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing. Second Edition.* 1992.