Biometrika (2012), **xx**, x, *pp.* 1–16

Printed in Great Britain

Approximating a convolution of lognormal random variables by a lognormal

1. Introduction

Lognormal random variables and their convolutions often arise in engineering problems such as wireless communications, and in bayesian models in the context of proteomics. While it is well known that the convolution of Normal random variables is Normal, no such result exists for the convolution of lognormal random variables, so we must rely on approximations. The litterature has largely framed the problem of convolving lognormals in the context of wireless communications, but the requirements imposed by the bayesian context are very different, and in particular require that the convolution be not only accurate, but also fast and stable as it is usually used inside of a sampling mechanism (e.g Gibbs sampler, Metropolis-Hastings, etc..).

The approximations suggested in the litterature can be broadly divided between lognormal and non-lognormal approximations. The latter category includes efforts such as Lam & Tho (2007) and Renzo et al. (2009) who proposed to approximate the convolution respectively with a Log Shifted Gamma and a Pearson Type IV family of distribution. While these methods have their merits, the distributions they propose are not standard in the Bayesian litterature and more generally have none of the analytical convenience, such as conjugacy and simplicity, usually sought in Bayesian models.

In this paper, we propose a lognormal approximation to the convolution of lognormal random variables. The lognormal approximations were pioneered by Fenton (1960) and Schwartz & Yeh (1982), who proposed to match the moments of the lognormal approximation with the moments of the convolution, on the power and the log-scale respectively. More recently, Mehta et al. (2007) proposed a more flexible method allowing to fit different parts of the convolution by tuning two additional parameters. We discuss these different methods in section 2.1 and describe our method in section 2.2.

Implicitely, all the methods seeking to approximate the lognormal convolution by a lognormal rely on an asymptotic result derived by Marlow (1967), the validity of which has not been thoroughly established in the non asymptotic case. In section 3, we empirically assess the validity of the approximation when convolving two lognormals using extensive simulations, and provide a simple map of the parameter space highlighting the regions for which the convolution is approximately lognormal (domain of validity).

Our method has many of the desired properties we mentioned earlier. It uses a standard distribution that is analytically easy to work with and compares favorably to other methods in terms of accuracy and stability as demonstrated in section 4.1. We identify scenarii in which our method may not be the most appropriate, and document them in section 4.2. We conclude with a discussion of future directions.

2. Methods

2·1. *General*

We first give a brief overview of the main methods for approximating the convolution of lognormals by another lognormal random variable. We point the reader to the Appendix for pseudo-code implementations of each method.

Let:

$$Y_1 \sim Normal(\mu_1, \sigma_1) \text{ and } Y_2 \sim Normal(\mu_2, \sigma_2)$$

where μ_i and σ_i are the location and scale parameters. All the methods that we examine in this paper share the common goal of approximating the distribution of $X \equiv e^{Y_1} + e^{Y_2}$ by a lognormal random variable $\tilde{X} \equiv LogNormal(\mu, \sigma)$, but differ in how they estimate its parameters μ and σ.

Fenton (1960) seeks an approximation whose first two moments match those of the convolution on the power scale, that is, $E[\tilde{X}] = E[X]$ and $Var[\tilde{X}] = Var[X]$. Mehta et al. (2007) showed that while the resulting approximation was accurate in the tail portion of the distribution (large values of X), it was quite inaccurate in the head portion of the distribution (small values of X). In contrast, Schwartz & Yeh (1982) choses an approximation whose first two moments match those of the convolution on the log scale, that is, $E[\log(\tilde{X})] = \mu = E[\log(X)]$ and $Var[\log(\tilde{X})] = \sigma^2 = Var[\log(X)]$. Mehta et al. (2007) showed that contrary to Fenton (1960), this approximation is rather accurate in the head portion of the distribution, but can be very inaccurate in the tail portion. Finally, Mehta et al. (2007) proposed to match the Moment Generating Functions (MGFs), that is for a given s_1 and s_2, $MGF_{\tilde{X}}(s_1) = MGF_X(s_1)$ and $MGF_{\tilde{X}}(s_2) = MGF_X(s_2)$. They showed that by tuning the parameters s_1 and s_2, it was possible to obtain accurate approximations of different parts of the density.

The key to understanding the proliferation of approximation methods is to keep in mind that the convolution of lognormal random variables is not exactly lognormal: there is no true lognormal that is equal to the convolution and whose parameters we can estimate, but rather, there are many sets of parameters that would yield acceptable approximations. The three methods' different choices of acceptable approximations are reflected in their different approaches to estimating the parameters of the lognormal approximation: Fenton's method (refered to as F-W) matches the first 2 moments of the convolution on the power scale, Shwartz & Yeh's method (refered to as S-Y) matches them on the log-scale, and Mehta et al. (referred to as Wu) matches the MGF. We make the distinction between these different *approaches* of lognormal approximation, and the methods used to analytically obtain the estimate, which we refer to as the *implementations*. This distinction between *approaches* of approximation and *implementations* is further discussed in section 5.

2·2. *Our method*

We present a new lognormal approximation, based on the work of Schwartz & Yeh (1982): our method uses the same *approach* of lognormal approximation as S-Y, but our *implementation* differs in that it deals with numerical integration using only Laplace approximations. Most of the derivations being identical, we only emphasize the differences between our method and S-Y, using the same notations that the original paper. We reproduce the full derivations in Appendix.

[Comment to Edo: Read appendix first]

The major difference between our method and S-Y is how we estimate the 3 quantities $G_1(\sigma_w, m_w)$, $G_2(\sigma_w, m_w)$, and $G_3(\sigma_w, m_w)$. The evaluation of the three quantities being similar, we illustrate our point with G_1.

Schwartz & Yeh (1982) evaluates the following integral, where $W \sim Normal(m_w, \sigma_w)$, with ($m_w$, σ_w) the location and scale :

$$\begin{aligned}
G_1 &= E[ln(1+e^w)] \\
&= \int_{-\infty}^{+\infty} [log(1+e^w)] f(w)\,\mathrm{d}w \\
&= \int_{-\infty}^{0} [log(1+e^w)] f(w)\,dw + \int_{0}^{+\infty} [log(1+e^{-w})+w] f(w)\,\mathrm{d}w \\
&= -\frac{\sigma_w}{\sqrt{2\pi}} exp(\frac{-m_w^2}{2\sigma_w^2}) + m_w \Phi(\frac{m_w}{\sigma_w}) \\
&+ \sum_{k=1}^{+\infty} C_k e^{k^2\sigma_w^2/2} \left[e^{km_w} \Phi(\frac{-m_w - k\sigma_w^2}{\sigma_w}) + e^{-km_w} \Phi(\frac{m_w - k\sigma_w^2}{\sigma_w}) \right]
\end{aligned}$$

with $C_k = \frac{(-1)^{k+1}}{k}$. Then the infinite series in the last equality is truncated to the 40^{th} order, and interpolated with the following low-order polynomials using least square fits:

$$log_{10} G_1(\sigma_w, m_w) = \sum_{j=0}^{4} \sum_{k=0}^{4} A_{jk} \sigma_w^{j/2} |m_w|^{k/2}$$

where the A_{jk} are tabulated in Schwartz & Yeh (1982).

One drawback of that method is that it has two layers of approximation: a truncation in the polynomial expansion, and a least squares polynomial interpolation. The second inconvenience of the method is that the coefficients A_{ij} are calculated only once, which makes the method very fast, but makes it lose in accuracy. Finally, large sums are prone to numerical instability.

Our method takes a different approach: we use a Laplace approximation to evaluate the integral, that is, we approximate the integrand by a normal pdf.

$$\begin{aligned}
G_1 &= E[ln(1+e^w)] \\
&= \int_{-\infty}^{+\infty} [log(1+e^w)] f_W(w)\,\mathrm{d}w \\
&= \int_{-\infty}^{+\infty} exp\{f_1(w|\mu_w, \sigma_w)\}\,\mathrm{d}w \\
&\approx \sqrt{\frac{2\pi}{M|f_1''(\hat{w})|^{-\frac{1}{2}}}} e^{f_1(\hat{w})}
\end{aligned}$$

with $f_1(w) = log[\frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{(w-m_w)^2}{2\sigma^2}) log(1+e^w)]$ and $\hat{w} = argmax_w(f_1(w))$. We see that contrary to S-Y, our method requires an optimization routine every time it is called. We thus trade some speed for a gain in accuracy, as will be illustrated in section 4.

The method also completely eliminates the large sum, reducing the risk of numerical instability.

3. Domain of validity

The different methods surveyed in the previous section implicitely assume that the convolution of lognormal random variables is close to a lognormal in some sense. Such a result has been established by Marlow (1967) when the number of lognormal random variables to be convolved goes to infinity, and although some efforts have been made towards evaluating its validity in the non-asymptotic case, there has not been a thorough study in the case of a finite number of summands. In this section, we partially adress this issue by delimiting a region of the parameter space for which the convolution of two lognormal random variables is approximately lognormal. The procedure can be readily extended to the case of more that two random variables.

The two methods we propose share the same goal: find the quadruples $(\mu_1, \mu_2, \sigma_1, \sigma_2)$ for which $e^{Y_1} + e^{Y_2} \overset{approx}{\sim} Lognormal(\mu, \sigma)$, where μ and σ are obtained by matching the first two moments of the convolution with those of the approximation, on the log scale. Due to the fact that the first two moments are not available in closed form, we must obtain them numerically. We could use S-Y or our method to estimate the moments, but we can obtain better precision if we use the slower but more accurate Monte-Carlo simulations (at this point, speed is not relevant). We then measure how far this approximation is from the convolution, using two separate metrics to ensure that the patterns observed are not artifacts of any method. For reasons that we shall explain below, we did not use the usual location-scale parametrization of the lognormal distribution, but instead used $m_i = E[e^{Y_i}]$ and $sd_i = SD(e^{Y_i})$.

We give an outline of the two methods, more detailed descriptions being left in Appendix.

Empirical CDF Method

The first method computes, for a given (m_1, m_2, sd_1, sd_2), the empirical CDF of the convolution as well as confidence bounds, using Monte-Carlo simulations. We then estimate the parameters of a lognormal whose first two moments match those of the convolution on the log scale, also using Monte-Carlo simulations. We then count how often the CDF of the approximation is outside the confidence bounds of the empirical cdf of the convolution.

More precisely, we obtain the empirical CDF with N_1 draws from the convolution, and then we get the confidence bounds by repeating the procedure N_2 times. The choice of N_1 is rather delicate, in particular: if N_1 is too large, the confidence bounds will be very narrow, and the CDF of the approximation will fall outside the bounds even if it is actually close to the convolution. Conversely, if N_1 is too small, the confidence bounds will be too wide and the CDF of the approximation will always fall inside the bounds, even if it far from the convolution. We found that fixing $N_1 = 100$ and $N_2 = 200$ allows us to capture the main patterns, which we will confirm with the second method.

Rejection Method

The second method runs, for a given (m_1, m_2, sd_1, sd_2), ten independent Markov Chain Monte Carlo (MCMC) whose common target distribution is the convolution of two lognormals, and the proposal is a lognormal distribution whose first two moments match those of the target, on the log scale, and considers the average rejection rate. The initialization of the MCMCs is very important here. Suppose for instance that the approximation models the convolution accurately between the 1^{st} and 90^{th} percentile, but then diverges significantly. If all the MCMC are initialized with values that fall between the 90^{th} and 99^{th} percentile, the initial rejection rates (before the chain mixes properly) might crudely misrepresent the approximation. We could partially remedy that issue by discarding some number of initial samples, but this could obscure patterns happening at high percentiles. We solved this issue by initializing the MCMC at multiple points spread between the 1^{st} and 99^{th} percentiles of the convolution, allowing us to capture patterns across a large part of the distribution.

We ran the two methods over many grids of parameters, and while no obvious patterns arose whith the location-scale (μ_i, σ_i) parametrization, we observed that the domain of validity could be described very succintly by the following set of inequalities, using the log(mean) - log(standard deviation) parametrization $(m_i, sd_i) = (\log[\text{mean}(Y_i)], \log[\text{SD}(Y_i)])$:

1. $sd_1 \leq sd_2 \leq m_1 \leq m_2$
2. $sd_1 \leq m_1 \leq sd_2 \leq m_2$
3. $m_1 \leq sd_1 \leq sd_2 \leq m_2$
4. $m_1 \leq m_2 \leq sd_1 \leq sd_2$
5. $m_1 \leq m_2 \leq sd_2 \leq sd_1$

Figure 1 summarizes visually the domain of validity. Since this was identified empirically, we cannot guarantee that the domain of validity holds for all possible values of m_1, m_2, sd_1 and sd_2 satisfying one of the previous inequalities. However, we observed that it holds well for $-3 \leq m_i \leq 25$ and $-3 \leq sd_i \leq 10$, which covers the range of values we are interested in in practice. It is also worth keeping in mind that the boundaries are soft, so that the closer the parameters are to the borders, the less reliable the approximation becomes.

[Eric observes: need a better justification for range of values above]

We now illustrate the perfomance of the different methods in the domain of validity.

4. Comparisons and Applications

4·1. *Methods Comparison*

Experimental Protocol

Bayesian methods are concerned with sampling from a posterior distribution that is not necessarily available analytically, and therefore often require the use of iterative sampling schemes. Markov Chain Monte Carlo algorithms, may require many iterations before convergence, so each individual step must be fast, accurate, and stable: fast because a gain of even 0.01 seconds in a step results in a gain of almost 3 hours over a million iterations, accurate because a systematic inaccuracies at each step can throw off the

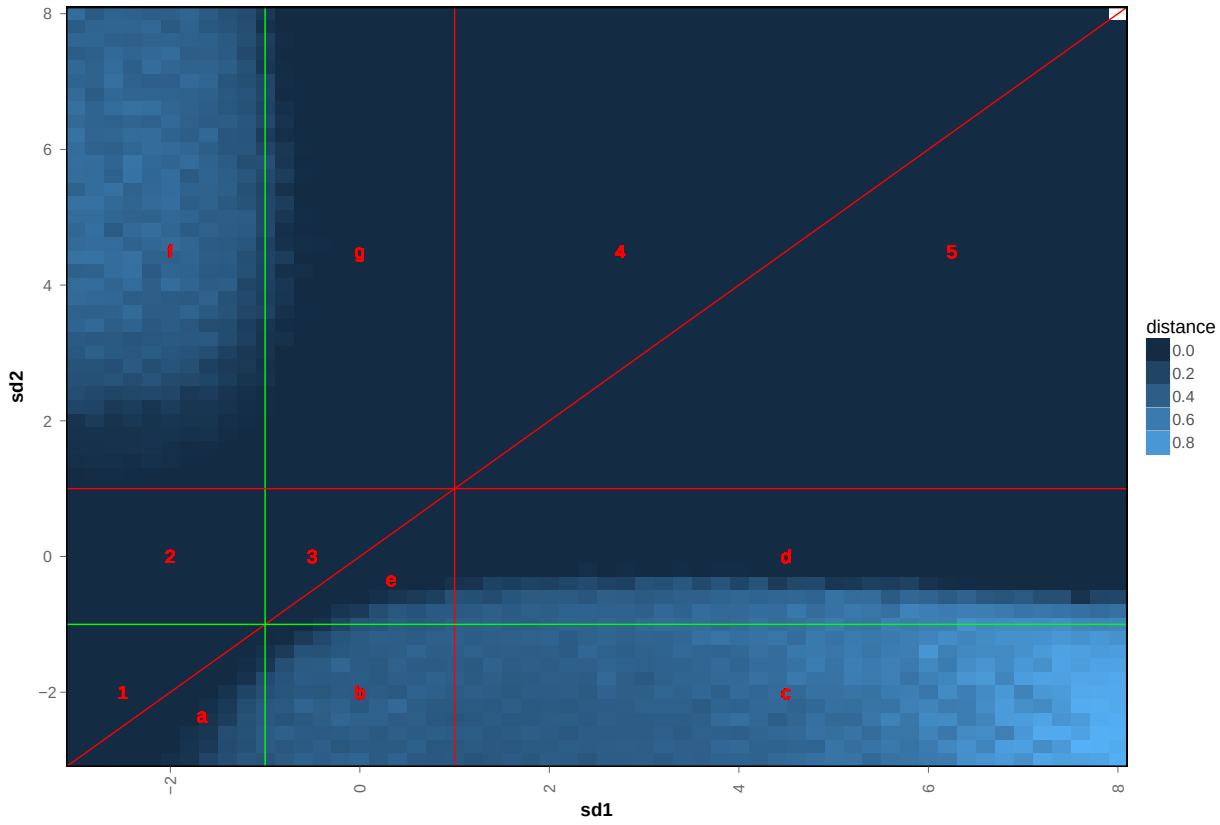

Fig. 1. CDF method, $m_1 = -1$ and $m_2 = 1$

entire inference, and stable because unpredictable errors or crashes can invalidate the whole inference, and result in a waste of computation time.

We conducted a large simulation in order to compare the speed, accuracy, and stability of the four methods described in this paper: S-Y, F-W, Wu, and ours. The performance of the different methods were evaluated on four grids of parameters covering the range useful in most applications (the range of practical parameters being usually given in the location-scale parametrization, we had to convert them to the mean-standard deviation parametrization, hence the slightly unusual bounds).

[Eric observes: the paragraph above is awkward]

- $m_1 = 1, m_2 = 6$ and $sd_1, sd_2 \in [-3, -2.8, ..., 9.8, 10]$
- $m_1 = 10, m_2 = 20$ and $sd_1, sd_2 \in [5, 5.3, ..., 25]$
- $sd_1 = 0.25, sd_2 = 3$ and $m_1, m_2 \in [-3, -2.85, ..., 7]$
- $sd_1 = 3, sd_2 = 6$ and $m_1, m_2 \in [0, 0.15, ..., 10]$

where $m_i = log(E[e^{Y_i}])$ and $sd_i = log(SD[e^{Y_i}])$ as in section 3. The first two grids have fixed means, and varying standard deviations, while the last two have fixed standard deviations and varying means, allowing us to cover a broader range of combinations. We then carried our measurements only on the subset of combinations that are in the domain of validity, as defined in section 3. Speed was measured by calling each method

	Rejection Rate				Execution Time (second per 10000 simulations)			
	m	sd	5%	95%	m	sd	5%	95%
US	0.01	0.02	0.001	0.06	12.4	3.2	9.0	17.7
WU	0.38	0.42	0.002	0.99	14.3	7.7	6	30
SY	0.1	0.22	0.002	0.76	2.9	0.89	2.15	4.78
FW	0.76	0.23	0.25	0.99	0.28	0.08	0.21	0.44

Table 1. *Global measures (averaged over simulations and regions of the domain of validity)*

10000 times for each combinations of parameters, over the 4 grids. The accuracy of the methods were measured using the rejection method described in section 3, except that the parameters of the approximation were estimated using the methods to be evaluated, rather than Monte-Carlo simulations. Stability was measured by counting the proportion of simulations that failed (i.e did not terminate or returned no values, for numerical reasons) when measuring the accuracy. How to measure stability is debatable: it is implementation-dependent, and can often be improved with enough efforts (use of taylor approximations, work on the log scale, etc..). So what we measured is the stability out of the box: we implemented all the algorithms exactly as described by their authors, using only standard libraries (in R) and without making additional efforts to improve stability (this also goes for our own method).

[Eric observes: the paragraph above is not precise enough]
Results

Table 1 sums up the performance of the four methods in the domain of validity, averaged over the four grids of parameters. Since the domain of validity was established using a lognormal approximation that matches the first two moments of the convolution in the log-scale, we would expect our method and S-Y to perform better than F-W and Wu, which rely on another approach of lognormal approximation. The simulations confirm that intuition, and show that both our method and S-Y perform significantly better than the other two methods. We also observe that, as predicted, our method is slower than S-Y due to the use of an optimization routine, but is significantly more accurate and stable. An important point concerning Wu's performance is that we did not allow s_1 and s_2 to be optimized as suggested in Mehta et al. (2007) but fixed them to values also suggested in the article: $s_1 = 0.001$ and $s_2 = 0.005$. We believe that allowing s_1 and s_2 to be optimized would significantly increase the accuracy of the method, but would slow the method even further, and would not solve the stability problem. So optimized or not, the method is not fit for our practical applications.

Although we outperform the other methods in terms of accuracy and stability, our method is significantly slower than FW and SY. This is an acceptable compromise in most Bayesian applications, but there may be some situations in which we would be willing to trade some accuracy and stability for speed. In the next section, we give examples of possible applications, and develop a cost model to help decide between the different methods, in different contexts.

Validity Domain		1	2	3	4	5	Overall
Proportion		0.17	0.33	0.31	0.13	0.06	-
Failure Rate	US	0	0	0	0	0	0
	WU	0	0	0.09	0.14	0.6	0.08
	SY	0.23	0.09	0.01	0.2	0.6	0.13
	FW	0	0	0	0	0	0

Table 2. *Failure rates in the different regions of the domain of validity*

4·2. *Applications*

Real world scenario

Before looking at more general contexts of applications, we turn to a case from our one research in which the problem of convolving lognormal random variables arises. When examining data from a mass-spectrometer, we realize that whether or not a particular peptide is seen depends on its overall abundance, that is, on the abundance of all the proteins from which it could come from. Since the abundance of proteins is modelled by lognormal random variables, the overall abundance arises as a convolution of lognormals. In our study, the inference mechanism makes use of a Markov Chain Monte Carlo, whithin which we sample from the convolution of lognormals at each step, and thus have to compute a an approximation for this convolution at every step. This is the typical case for which our method was designed: accuracy and stability are of paramount importance, and speed, while important, can be compromised with. But in following the progression of the paper, we still need to verify that the parameters are located in the zone of validity. Given a specfic collection of 800 lognormal variables and their paramters modelling 800 proteins, we sample 10000 pairs (with repetitions) and record where in the parameter space do the parameters fall. We see that 91% of the parameters fall in the valid zone, while 9% fall outside of it. We consider is an acceptable rate.

We recognize that while our method is better that its competitors in the context we designed it for, namely bayesian modelling, there may be cases for which it is ill suited. We now present two more general cost models illustrating different scenarios in which the choice of the optimal method is not straightforward.

Below are summarized the results of the scenario:
General Cost Model

Bayesian modeling is only one of many possible applications for the lognormal approximations. Having shown that our method is well suited for the Bayesian types of applications, we shall now present scenarii where other methods might be more appropriate. Consider the following scenario: let $r =$ rejection rate, $f =$ failure rate and suppose that the approximation takes place inside of a broader simulation. Each simulation takes a time $t + h$ where t is the execution time intrinsic to each method (Wu, F-W, S-Y, and ours), and h is a constant additional execution time independant of the execution time t. Assuming that if a simulation fails or is rejected we have to re-run it, we can write the following cost model for a successful simulation:

			Rejection Rate				Stability
			m	sd	5%	95%	
US	OZ	N=2	0.0045	0.0042	0.0016	0.012	1
		N=3	0.0054	0.0048	0.0018	0.014	
		N=4	0.0057	0.0053	0.0020	0.016	
		N=5	0.0061	0.0047	0.002	0.017	
	NOZ	N=2	0.0028	0.002	0.0015	0.0061	
		N=3	0.004	0.0042	0.002	0.01	
	AV		0.0047	0.0044	0.0016	0.013	
WU	OZ	N=2	0.0045	0.0043	0.0016	0.012	0.94
		N=3	0.0057	0.0048	0.0018	0.015	
		N=4	0.0088	0.039	0.002	0.019	
		N=5	0.027	0.14	0.0026	0.019	
	NOZ	N=2	0.0028	0.002	0.0016	0.0063	
		N=3	0.0057	0.0054	0.0024	0.012	
	AV		0.0051	0.015	0.0016	0.0132	
SY	OZ	N=2	0.26	0.064	0.14	0.36	1
		N=3	0.38	0.054	0.28	0.46	
		N=4	0.46	0.048	0.38	0.52	
		N=5	0.52	0.033	0.46	0.57	
	NOZ	N=2	0.0028	0.002	0.0016	0.006	
		N=3	0.0057	0.0054	0.0022	0.012	
	AV		0.31	0.097	0.15	0.48	
FW	OZ	N=2	0.82	0.035	0.76	0.87	1
		N=3	0.84	0.023	0.81	0.88	
		N=4	0.85	0.022	0.81	0.89	
		N=5	0.87	0.03	0.83	0.92	
	NOZ	N=2	0.82	0.032	0.76	0.87	
		N=3	0.84	0.016	0.82	0.86	
	AV		0.83	0.033	0.76	0.88	

	OK Zone				NOK Zone	
	N=2	N=3	N=4	N=5	N=2	N=3
proportion	0.632	0.198	0.0999	0.104	0.0588	0.0009

		Execution time						Execution time			
		m	sd	5%	95%			m	sd	5%	95%
US	N=2	14.3	6.1	7.5	23.5	WU	N=2	22.0	7.5	15.2	38.2
	N =3	27.5	11.6	14.6	45.2		N =3	25.6	8.9	16.7	43.5
	N=4	40.0	16.5	21.7	67.0		N=4	26.3	9.8	15.8	45.8
	N=5	51.1	21.8	28.8	90.8		N=5	25.7	9.5	15.5	42.0
	AV	19.9	12.9	7.5	45.0		AV	23.2	8.3	15.4	41.2
		m	sd	5%	95%			m	sd	5%	95%
SY	N=2	6.8	2.9	4.1	13.1	FW	N=2	4.9	1.7	3.2	8.7
	N =3	12.9	5.3	7.8	24.6		N =3	5.4	1.9	3.6	9.7
	N=4	18.8	7.6	11.6	35.6		N=4	6.1	2.2	3.9	11.0
	N=5	26.0	10.3	15.4	48.3		N=5	7.2	2.9	4.4	12.3
	AV	9.4	6.1	4.2	22.4		AV	5.2	1.9	3.3	9.1

Method	Average Time per 10000 successes ($h = 10$)	Average Time per 10 000 successes ($h = 0.5$)
FW	41.6	3.25
US	10.1	13.0
SY	12.6	4.3
WU	25.0	26

Table 3. *Average time (in seconds) for 10000 successful draws, under the extended cost model, with* $h = 10$ *and* $h = 0.5$

$$\begin{aligned} C &= (t + h + C)f + (1 - f)[(t + h + C)r + (t + h)(1 - r)] \\ &= \frac{t + h}{1 - f - (1 - f)r} \end{aligned}$$

If $h \gg t$, the scenario can model the cost of a successful draw using an MCMC, in which our approximation is just a small part. But other scenarii can also flexibly be modeled by changing the value of the overhead h. Depending on the value of h, different methods stand out as being most appropriate. Table 3 uses the values of t, f and r in Table 1 and Table 2 to give the cost of the different methods under different values of h: when h is large ($h = 10$) the scenario models the sampling from an MCMC, and our method has indeed the smallest cost. When h is small however ($h = 0.5$), speed is the most important factor, and F-W ends up having the smallest cost.

We don't expect this cost model to adequately model every situation, but we hope it illustrates the fact that which method is the most appropriate largely depends on the specifics of the application it is used for.

5. Discussion

Two generalizations of the convolution of lognormals have been studied in the litterature: the convolution of more than two lognormal random variables, and the convolution of correlated lognormal random variables. Our method can be readily extended to cover both cases: more than two lognormal random variables can be convolved by applying the described algorithm recursively. As for the correlation terms, it can be accounted for in the correlation term between Y_1 and W in the derivations in section 2. We omitted it for readability reasons.

Keeping in mind the fact that the convolution of lognormals is fundamentally not lognormal, it is helpful to think of all the methods as having two sources of errors: a first source of error (type A) that is intrinsic to *approach* of lognormal approximation (matching moments on the regular scale? on the log scale?), which can be evaluated by using Monte Carlo simulations. The second source of error (type B) depends entirely on the *implementation*, that is, the set of approximations made in the analytical derivations (Laplace approximation, polynomial interpolation, etc..). To give a simple example: SY and our method have identical type A errors since both of them seek to match moments in the log scale, however, the errors of type B are different since the analytical derivations rely on different sets of approximations. It helps to think of the type A error as the minimum error achievable for a certain method of approximation: if it was possible, given a certain *approach*, to obtain exact analytical derivations, then there would be no

type B error. However, there would still be a non-zero type A error, inherent to the *approach* taken.

These concepts of "sources of errors" and *approaches* give some useful insight into the "domain of validity" described in section 2. We explained that we used Monte-Carlo simulations to determine the parameters of the approximations by matching moments on the log-scale. In other words, we measured the error of type A intrinsic to the moment matching on the logscale. So what the "domain of validity" really tells us is how good we can hope to be if our method could get exact analytical expressions for the moment matching on the logscale. An interesting feature of this procedure is that one could easily adapt it to devise the domain of validity corresponding to another *approach*, for instance the moment matching on the power scale. In fact, one would obtain a different domain of validity for each *approach*, and could then compare their size, or practical implications. In other words, the caracteristics of the domain of validity could become another criterion for comparing the different methods.

In fact, we believe that there would be little to be gained in improving the accuracy of our method in the domain of validity, since it has reached a level of accuracy that is more than enough for most practical applications. The two points that would greatly benefit from improvements are the speed of the algorithm, and the size of the domain of validity.

References

Fenton, L. F. (1960). The Sum of Log-Normal Probability Distributions inScatterTransmission Systems. *IRE Transactions on Communications Systems* , 1–11.

Lam, C. L. J. & Tho, L.-N. (2007). Log-Shifted Gamma Approximation toLognormal Sum Distributions. *IEEE Trans. Veh. Technol.* , 1–9.

Marlow, N. A. (1967). A Normal Limit Theorem for Power Sums of Independent Random Variables. *The Bell System Technical Journal* , 1–10.

Mehta, N., Wu, J., Molisch, A. & Zhang, J. (2007). Approximating a Sum of Random Variables with a Lognormal. *IEEE Transactions on Wireless Communications* , 1–12.

Renzo, M., Graziosi, F. & Graziosi, F. (2009). Approximating the linear combination of log-normal RVs via pearson type IV distribution for UWB performance analysis. *IEEE Transactions on Communications* **57**, 388–403.

Schwartz, S. C. & Yeh, S. C. (1982). On the Distribution Function and Moments of Power Sums With Log-Normal Components. *The Bell System Technical Journal* **61**, 1–22.

6. Appendix A: Our method - details

We are interested in the distribution of $Z \equiv log(e^{Y_1} + e^{Y_2})$, where we assume Y_1 and Y_2 to be independent. We briefly discuss the extension to correlated random variables in secton 5.

While it is known that this sum is generally not another Normal random variable, we seek a reasonable approximation of the form:

$$Z \overset{approx}{\sim} Normal(\mu_z, \sigma_z^2)$$

We attempt to do so by matching μ_z and σ_z^2 with the mean and variance of Z. We define a new random variable:

$$W \equiv Y_2 - Y_1$$

and note:

$$\mu_w = \mu_2 - \mu_1$$
$$\sigma_w^2 = \sigma_1^2 + \sigma_2^2$$

Deriving E(Z)

We compute the expectation of Z:

$$\begin{aligned} E[Z] &= E\log(e^{Y_1} + e^{Y_2}) \\ &= E\log(e^{Y_1}(1 + e^W)) \\ &= E[Y_1] + E\log(1 + e^W) \\ &= \mu_1 + E\log(1 + e^W) \end{aligned}$$

Using LOTUS we can write:

$$\begin{aligned} E[log(1 + e^W)] &= \int_{-\infty}^{\infty} \log(1 + e^w)\phi(\frac{w - \mu_w}{\sigma_w})\,\mathrm{d}w \\ &= \int_{-\infty}^{\infty} \exp\left\{\frac{-1}{2\sigma_w^2}(w - \mu_w)^2 + \log[\log(1 + e^w)] - \frac{1}{2}\log(2\pi\sigma_w^2)\right\}\,\mathrm{d}w \\ &= \int_{-\infty}^{\infty} \exp\{f_1(w|\mu_w, \sigma_w)\}\,\mathrm{d}x \end{aligned}$$

Which can be approximated using a Laplace approximation:

$$\int_{-\infty}^{\infty} \exp\{f_1(w|\mu_w, \sigma_w)\}\,\mathrm{d}w \approx \sqrt{\frac{2\pi}{M|f_1''(\hat{w})|^{-\frac{1}{2}}}} e^{f_1(\hat{w})} \equiv c_1$$

where $\hat{w} \equiv \underset{w \in \mathbb{R}}{\text{argmax}} f(w)$. The derivations of the mode and second derivatives are provided in Appendix.

And so finally, we have:

$$E[Z] = \mu_1 + c_1$$

Deriving Var(Z)

We compute $Var(Z) = E[Z^2] - E[Z]$. Where:

$$\begin{aligned} E[Z^2] &= E\{[Y_1 + \log(1 + e^w)]^2\} \\ &= E\{Y_1^2 + 2Y_1\log(1 + e^w) + \log(1 + e^w)^2\} \\ &= \mu_1^2 + \sigma_1^2 + 2E[Y_1\log(1 + e^w)] + E[\log(1 + e^w)^2] \end{aligned}$$

We note that (Y_1, w) are jointly normal with correlation:

$$\begin{aligned}\rho &= E\left[\frac{(Y_1-\mu_1)(w-\mu_w)}{\sigma_1\sigma_w}\right] \\ &= \frac{-\sigma_1}{\sigma_w}\end{aligned}$$

Further,

$$\begin{aligned}E[Y_1|w] &= \mu_1 + \rho\sigma_1\frac{(w-\mu_w)}{\sigma_w} \\ &= \mu_1 - \rho^2(w-\mu_w)\end{aligned}$$

Using the previously stated result, along with the law of iterated expectations, we have:

$$\begin{aligned}A &\equiv 2E[Y_1\log(1+e^w)] \\ &= 2E\{\log(1+e^w)E[Y|w]\} \\ &= 2E\{\log(1+e^w)[\mu_1-\rho^2(w-\mu_w)]\} \\ &= 2\mu_1E[\log(1+e^w)] - 2\rho^2E[\log(1+e^w)(w-\mu_w)] \\ &= A_1 - A_2\end{aligned}$$

where

$$\begin{aligned}A_1 &\equiv 2\mu_1E[\log(1+e^w)] \\ &= 2\mu_1(\mu_z-\mu_1)\end{aligned}$$

and

$$\begin{aligned}A_2 &\equiv 2\rho^2E[\log(1+e^w)(w-\mu_w)] \\ &= 2\rho^2\int_{-\infty}^{\infty}\log(1+e^w)(w-\mu_w)\phi\left(\frac{w-\mu_w}{\sigma_w}\right)\,\mathrm{d}w \\ &= 2\rho^2\sigma_w^2\int_{-\infty}^{\infty}\frac{e^w}{1+e^w}\phi\left(\frac{w-\mu_w}{\sigma_w}\right)\,\mathrm{d}w \\ &= 2\rho^2\sigma_w^2\int_{-\infty}^{\infty}e^{f_2(w|\mu_w,\sigma_w^2)}\,\mathrm{d}w \\ &= 2\rho^2\sigma_w^2c_2\end{aligned}$$

using the same method as above. Finally, the last term can be re-written:

$$\begin{aligned}E[log(1+e^w)^2] &= \int_{-\infty}^{\infty}\log(1+e^w)^2(2\pi\sigma^2)^{-1/2}\exp\left\{\frac{-1}{2\sigma^2}(w-\mu_w)^2\right\}\,\mathrm{d}w \\ &= \int_{-\infty}^{\infty}\exp\left\{f_3(w|\mu_w,\sigma_w)\right\}\,\mathrm{d}w \\ &= c_3\end{aligned}$$

Putting all the pieces together, we get:

$$\begin{aligned} E[Z^2] &= \mu_1^2 + \sigma_1^2 + 2E[Y_1 \log(1+e^w)] + E[\log(1+e^w)^2] \\ &= \mu_1^2 + \sigma_1^2 + (2\mu_1 c_1 - 2\rho^2\sigma^2 c_2) + c_3 \end{aligned}$$

which enables us to simply compute the variance of Z as:

$$\begin{aligned} \sigma_z^2 &= E[Z^2] - E[Z]^2 \\ &= \mu_1^2 + \sigma_1^2 + (2\mu_1 c_1 - 2\rho^2\sigma^2 c_2) + c_3 - (\mu_1 - c1)^2 \end{aligned}$$

The main advantage of our method over S-Y is the elimination of the polynomial interpolation and uncentered Gauss-Hermite expansion. This results in a more accurate method as we shall show in section 3.

Appendix B: Other methods

Input: $\mu_1, \mu_2, \sigma_1, \sigma_2$
Output: E_z, V_z
begin
  $u_1 \leftarrow e^{\mu_1+\sigma_1^2/2} + e^{\mu_2+\sigma_2^2/2}$;
  $u_2 \leftarrow e^{2\mu_1+2\sigma_1^2} + e^{2\mu_2+2\sigma_2^2} + 2e^{\mu_1+\mu_2+(\sigma_1^2+\sigma_2^2)/2}$;
  $E_z \leftarrow 2\log(u_1) - 0.5\log(u_2)$;
  $V_z \leftarrow \log(u_2) - 2\log(u_1)$;
  return *(E_z, V_z)*;
end

Algorithm 1: The FW method

Input: $\mu_1, \mu_2, \sigma_1, \sigma_2, s_1, s_2, w, a, maxiter, initv$
Output: μ_z, σ_z
begin
  $\hat{\psi}_{11} \leftarrow \sum_{i=1}^{N} \frac{w_i}{\sqrt{\pi}} \exp\left[-s_1 \exp\left(\sqrt{2}\sigma_1 a_i + \mu_1\right)\right]$;
  $\hat{\psi}_{12} \leftarrow \sum_{i=1}^{N} \frac{w_i}{\sqrt{\pi}} \exp\left[-s_2 \exp\left(\sqrt{2}\sigma_1 a_i + \mu_1\right)\right]$;
  $\hat{\psi}_{21} \leftarrow \sum_{i=1}^{N} \frac{w_i}{\sqrt{\pi}} \exp\left[-s_1 \exp\left(\sqrt{2}\sigma_2 a_i + \mu_2\right)\right]$;
  $\hat{\psi}_{22} \leftarrow \sum_{i=1}^{N} \frac{w_i}{\sqrt{\pi}} \exp\left[-s_2 \exp\left(\sqrt{2}\sigma_2 a_i + \mu_2\right)\right]$;
  $\psi_1(\mu_z, \sigma_z) \leftarrow \sum_{i=1}^{N} \frac{w_i}{\sqrt{\pi}} \exp\left[-s_1 \exp\left(\sqrt{2}\sigma_z a_i + \mu_z\right)\right]$;
  $\psi_2(\mu_z, \sigma_z) \leftarrow \sum_{i=1}^{N} \frac{w_i}{\sqrt{\pi}} \exp\left[-s_2 \exp\left(\sqrt{2}\sigma_z a_i + \mu_z\right)\right]$;
  $(\mu_z, \sigma_z) \leftarrow solve([\psi_1(\mu_z, \sigma_z) == \hat{\psi}_{11} \cdot \hat{\psi}_{21}] \& [\psi_2(\mu_z, \sigma_z) ==$
  $\hat{\psi}_{21} \cdot \hat{\psi}_{22}]$, initv, maxiter);
  return *(μ_z, σ_z)* ;
end

Algorithm 2: The WU method

Input: $\mu_1, \mu_2, \sigma_1, \sigma_2, A_{jk}^{(1)}, A_{jk}^{(2)}, A_{jk}^{(3)}$
Output: μ_z, σ_z
begin
$\mu_w \leftarrow \mu_2 - \mu_1$;
$\sigma_w^2 \leftarrow \sigma_1^2 + \sigma_2^2$;
$G_{(1)} \leftarrow 10^{\sum_{j=0..4} \sum_{k=0..4} A_{jk}^{(1)} \sigma_w^{j/2} |\mu_w|^{k/2}}$;
$G_{(2)} \leftarrow 10^{\sum_{j=0..4} \sum_{k=0..4} A_{jk}^{(2)} \sigma_w^{j/2} |\mu_w|^{k/2}}$;
$G_{(3)} \leftarrow 10^{\sum_{j=0..4} \sum_{k=0..4} A_{jk}^{(3)} \sigma_w^{j/2} |\mu_w|^{k/2}}$;
$\sigma_z^2 \leftarrow \sigma_1^2 - G_{(1)}^2 - 2\frac{\sigma_1^2}{\sigma_w^2} G_{(3)} + G_{(2)}$;
$\mu_z \leftarrow \mu_1 + G_{(1)}$;
return (μ_z, σ_z) ;
end

Algorithm 3: The SY method

Appendix C: Domain of Validity Methods

Empirical CDF method

1. Draw 100 points from the convolution, using MC simulations, and compute its empirical cdf (using the R function *ecdf* for instance).
2. Repeat the first step 200 times.
3. Put all the 100x200 draws together, compute the 1^{st} and 99^{nth} quantiles, and take 2000 points uniformly spread between the quantiles.
4. For each of the 2000 points picked in step 4, compute the value of each of the 200 empirical cdf computed in step 2, and from use these 200 values to obtain the bounds of the 95% confidence interval, along with the mean of the 200 values. At the end of this step, you should have, for each of the 2000 points, three values: a mean, an upper bound (for the 95% CI) and a lower bound (for the 95% CI).
5. We then draw 2000000 points from the convolution, and match the first and second moments to those of a lognormal, *on the log scale*. That is, if X and Y are the two lognormal rv to be convolved, we approximate their convolution by a random variable Z, with a lognormal distribution, whose parameters are $\mu = mean(log(X+Y))$ and $\sigma = \sqrt{Var(log(X+Y))}$.
6. We compute evaluate the theoretical CDF of the approximate lognormal random variable at each of the 2000 points picked in step 4
7. We count the proportion of the 2000 points for which the approximate lognormal falls *outside* of the 95% CI.
8. We identify the "OK-zone" with the regions where the proportion computed in step 7 is low.

Rejection rate method

1. Draw 1000000 points from the convolution (MC) and pick 10 points uniformly between the 1^{st} and 99^{nth} quantiles. These 10 points will serve as starting points for the independent MCMC that we run next.

2. Draw another 1000000 points from the convolution, and compute the mean (μ) and standard deviation (σ) on the log scale. We will use $Y \sim LogNormal(\mu, \sigma)$ as a proposal in the following mcmc.
3. Draw 5000000 points from the convolution, transform the data in the log scale, compute the 0.01^{st} and the 99.99^{nth} quantiles and use the values in between to compute the approximate pdf of the convolution (using the *density*) function in R for instance), and use the transformation theorem ($f_Z(z) = f_X(log(z)) * \frac{1}{z}$)to bring back the approximate pdf on the regular scale. The pdf thus obtained will be used as target density in the following mcmc.
4. For each starting points, run an independent mcmc using the proposal and target distributions defined in step 2 and 3. Each of the mcmc has 2000 iterations. For each mcmc, record the proportion of rejections.
5. Compute the average rejection rates over all the 10 parallel mcmcs' and report the result.