Risk aggregation in the presence of discrete causally connected random variables

Peng Lin*

Phd candidate, School of Electronic Engineering and Computer Science, Queen Mary, University of London, UK

Martin Neil

Agena Ltd and Professor of Computer Science and Statistics, School of Electronic Engineering and Computer Science, Queen Mary, University of London, UK

Norman Fenton

CEO, Agena Ltd and Professor of Risk Information Management, School of Electronic Engineering and Computer Science, Queen Mary, University of London, UK

Abstract

Risk aggregation is a popular method used to estimate the sum of a collection of financial assets or events, where each asset or event is modelled as a random variable. Applications include insurance, operational risk, stress testing and sensitivity analysis. In practice, the sum of a set of random variables involves the use of two well-known mathematical operations: *n-fold* convolution (for a fixed number n) and *N-fold* convolution, defined as the compound sum of a frequency distribution N and a severity distribution, where the number of constant *n-fold* convolutions is determined by N, where the severity and frequency variables are independent, and continuous, currently numerical solutions such as, Panjer's recursion, fast Fourier transforms and Monte Carlo simulation produce acceptable results. However, they have not been designed to cope with new modelling challenges that require hybrid models containing discrete explanatory (regime switching) variables or where discrete and continuous variables are inter-dependent and may influence the severity and frequency in complex, non-linear, ways. This paper describes a *Bayesian Factorisation and Elimination* (BFE) algorithm that performs convolution on the hybrid models required to aggregate risk in the presence of causal dependencies. This algorithm exploits a number of advances from the field of Bayesian Networks, covering methods to approximate statistical and conditionally deterministic functions to factorise multivariate distributions for efficient computation. Experiments show that BFE is as accurate on conventional problems as competing methods. For more difficult hybrid problems BFE can provide a more general solution that the others cannot offer. In addition, the BFE approach can be easily extended to perform deconvolution for the purposes of stress testing and sensitivity analysis in a way that competing methods do not.

*Correspondence to: Peng Lin, Department of EECS, Queen Mary, University of London, UK. Tel: +44 (0) 20 78828027; Fax: +44 (0) 870 131 8460; E-mail: p.lin@qmul.ac.uk

1. Motivation and Introduction

Risk aggregation is a popular method used to estimate the sum of a collection of financial assets or events, where each asset or event is modelled as a random variable. Existing techniques make a number of assumptions about these random variables. First, they are almost always continuous. Second, if they are independent then they are identically distributed. Third, should they be dependent, these dependencies are best represented by correlation functions, such as copulas (Nelsen, 2007; Embrechts, 2009), where marginal distribution functions are linked by some dependence structure. These statistical methods have tended to model associations between variables as a purely phenomenological artefact extant in historical statistical data. Recent experience, at least since the beginning of the financial crisis in 2007, has amply demonstrated the inability of these assumptions to handle non-linear effects or "shocks" on financial assets and events, resulting in models that are inadequate for prediction, stress testing and model comprehension (Laeven & Valencia, 2008; IMF, 2009).

It has been extensively argued that modelling dependence as correlation is insufficient, as it ignores any views that the analyst may, quite properly, hold about those causal influences that help generate and explain the statistical data observed (Meucci, 2008; Rebonato, 2010). Such causal influences are commonplace and permeate all levels of economic and financial discourse. For example, does a dramatic fall in equity prices cause an increase in equity implied volatilities or is it an increase in implied volatility that causes a fall in equity prices? The answer is trivial in this case, since a fall in equity prices is well known to affect implied volatility, but correlation alone contains no information about the direction of causation. To incorporate causation we need to involve the analyst or expert and "fold into" the model views of how discrete events interact and the effects of this interaction on the aggregation of risk. This approach extends the methodological boundaries last pushed back by the celebrated Black–Litterman model (Black & Litterman, 1991). In that approach a risk manager's role is as an active participant in the risk modelling, and the role of the model is to accommodate their subjective "views", expressed as Bayesian priors of expectations and variances of asset returns. In this paper we aim to represent these Bayesian "views" in an explicit causal structure, whilst providing the computational framework for solutions. Such causal models would involve discrete explanatory (regime switching) variables and hybrid mixtures of inter-dependent discrete and continuous variables. A causal risk aggregation model might incorporate expert derived views about macro-economic, behavioural, operational or strategic factors that might influence the assets or events under "normal" or "abnormal" conditions. Applications of the approach include insurance, stress testing, operational risk and sensitivity analysis.

At its heart risk aggregation requires the sum of n random variables. In practice, this involves the use of two well-known mathematical operations: *n-fold* convolution (for a fixed value of n) and *N-fold* convolution (Heckman & Meyers, 1983), defined as the compound sum of a frequency distribution N, and a severity distribution S, where the number of constant *n-fold* convolutions is determined by N, stochastically. Currently, popular methods such as Panjer's recursion (Panjer, 1981), fast Fourier transforms (FFT, Heckman & Meyers, 1983) and Monte Carlo (MC) simulation (Fishman, 1996) perform risk aggregation numerically using parameters derived from historical data to estimate the distributions for both S and N. Where S and N are independent and continuous, these approaches produce acceptable results. However, they have not been designed to cope with the new modelling challenges outlined above. In the context of modelling general dependencies among severity variables, a popular approach is to use copulas, both to model the dependent variables and to perform risk aggregation.

Our aim then is to show how we can carry out a stochastic risk aggregation (*N-fold* convolution) in a causal Bayesian framework, in such a way that subjective views about inter-dependencies can be explicitly modelled and numerically evaluated, i.e., where discrete and continuous variables are inter-dependent and may influence N and S in complex, non-linear ways. We see this as the first of many financial modelling problems that are amenable to this new approach.

This paper describes a *Bayesian Factorisation and Elimination* (BFE) algorithm that performs convolution on the hybrid models required to aggregate risk in the presence of causal dependencies. This algorithm exploits a number of advances from the field of Bayesian Networks (BNs), covering methods to approximate statistical and conditionally deterministic functions and to factorise multivariate distributions for efficient computation.

Section 2 provides an overview of popular methods for risk aggregation. Section 3 describes BN technology with a view to explaining some of the core foundational algorithms used in this paper. The BFE convolution algorithm is described in section 4, showing how it builds and extends on the standard BN algorithms presented in section 3. Section 5 presents a version of BFE that performs deconvolution and section 6 presents experimental results showing the performance of BFE. Section 7 concludes the paper.

2. Risk Aggregation

An encyclopaedic overview of the current state of the art in risk aggregation is presented in McNeil *et al.* (2010). The general aggregation formula for fixed, n, assets, is:

$$T = S_0 + S_1 + \dots + S_n \tag{1}$$

where T is the sum of n asset valuations and each S_i is from the same common continuous distribution f_x, which can be thought of as a return (severity) distribution S. This is called an *n-fold* convolution. If $S \sim f_x$ and if we have a variable number of assets, N, then equation (1) can be rewritten as an *N-fold* convolution:

$$f_T(x) = \sum_{j=0}^{\infty} f^{*j}(x) P(N = j) \tag{2}$$

where $f^{*j}(x) = \int_0^{\infty} f^{*(j-1)}(x-y) f(dy)$ is a recursive *n-fold* convolution on S. We can therefore rewrite equation (2) in a discrete form: $P(N = j) = a_j$, for $j = 0, 1, \dots, L$, where L is the length of discretised frequency N. The following expressions hold:

$$P(T) = a_0 P(T_0) + a_1 P(T_1) + \dots + a_L P(T_L) \tag{3}$$

$$T_0 = S_0, \ T_1 = S_0 + S_1, \dots, \ T_L = S_0 + S_1 + \dots + S_L \tag{4}$$

where each T_j is a constant *n-fold* convolution. The equation (3) represents a mixture distribution where the mixture components consist of mutually exclusive variables, themselves composed using the conditionally deterministic functions stated in equation (4).

For the sake of clarity in insurance, and similar, applications N is interpreted as a frequency distribution and S is defined as a severity (loss) distribution.

General numerical solutions to computing the aggregate distribution include Panjer's recursion (Panjer, 1981), fast Fourier transform (Heckman & Meyers, 1983) and MC simulation (Fishman, 1996).

300

In this paper severity variables can depend on discrete explanatory variables with dependencies expressed via conditioning in BN. This contrasts with the classic approach for dependency modelling among severity variables using copulas. Rather than use dependency and conditioning the copula approach models the dependency structure independently with marginal functions, which supports the construction of high-dimensional models.

In the context of copula-based risk aggregation Bruneton (2011) proposes the use of hierarchical aggregation using copulas. Also, Arbenz & Canestraro (2012) proposes hierarchical risk aggregation based on tree dependence modelling using step-wise low-dimensional copulas, and also gives a sample reordering algorithm for numerical approximation. Brechmann (2014) suggests hierarchical Kendall copulas to achieve flexible building blocks, where risk aggregation is supported by the Kendall function. These approaches capture the joint dependencies from a hierarchical structure and exploit use of small building blocks. In contrast to correlation modelling, our work assumes causality and dependency, where joint dependency is decomposed by conditional dependencies using the BN framework.

3. BN

3.1. Background

A BN (Pearl, 1993; Lauritzen, 1996; Jensen & Nielsen, 2009) consists of two main elements:

1. *Qualitative*: This is given by a directed acyclic graph with nodes representing random variables, which can be discrete or continuous, and may or may not be observable, and directed arcs (from parent to child) representing causal or influential relationships between variables.

2. *Quantitative*: A probability distribution associated with each node X. For a node with parents this is a conditional probability distribution (CPD), $P(X|\text{pa}(X))$ that defines the probabilistic relationship of node given its respective parents $\text{pa}(X)$. For each node X without parents, called root nodes, this is their marginal probability distribution $P(X)$. If X is discrete, the CPD can be represented as a node probability table (NPT), $P(X|\text{pa}(X))$, which lists the probability that X takes, on each of its different values, for each combination of values of its parents $\text{pa}(X)$. For continuous variables, the CPDs represent conditional probability density functions.

Together, the qualitative and quantitative parts of the BN encode all relevant information contained in a full joint probability model. The conditional independence assertions about the variables, represented by the absence of arcs, allow decomposition of the underlying joint probability distribution as a product of CPDs. Specifically:

$$P(X_1, \ldots, X_n) = \prod P(X_i|\text{pa}(X_i))$$

This significantly reduces the complexity of inference tasks on the BN (Spiegelhalter & Lauritzen, 1990; Fenton & Neil, 2012).

BNs have already been employed to address financial problems. For example, in Cowell *et al.* (2007) BNs were used for overall loss distribution and making predictions for insurance; in Neil & Fenton (2008) BNs were used for modelling operational risk in financial institutes, while the work in Politou & Giudici (2009) combines MC simulation, graphic models and copula functions to build operational risk models for a bank. Likewise, Rebonato (2010) discusses a coherent stress testing approach using BNs.

We have chosen to use BNs because the latest algorithms can model causal dependencies between hybrid variables during inference, to produce approximate posterior marginal distributions for the variables of interest. Also, by virtue of Bayes' theorem they are agnostic about causal direction and can perform inference from cause to effect and vice versa (or convolution to deconvolution, as is the case here). Until very recently BN tools were unable to properly handle non-Gaussian continuous variables, and so such variables had to be discretised manually, with inevitable loss of accuracy. A solution to this problem was described in Neil *et al.* (2007) based on an extension of the junction tree (JT) inference algorithm, and is described below in section 3.2. The result of inference is a set of queries on the BN in the form of univariate or multivariate posterior marginal distributions. This allows the approximate solution of classical Bayesian statistical problems, involving continuous variables as well as hybrid problems involving both discrete and continuous variables, without any restriction on distribution family or any requirement for conjugacy. This scheme iteratively converges on the posterior solution and has provided highly efficient solutions in a number of domains (Marquez, Neil, & Fenton, 2010); (Fenton & Neil, 2012).

Both exact and approximate inference in BNs is NP-hard (Cooper & Herskovits, 1992) and the efficiency of the JT architecture depends on the size of the clusters in the associated tree. To help reduce conditional probability table (CPT) size we employ a factorisation scheme called binary factorisation (BF, described below in section 3.3) to reduce the size and associated computation time required for continuous variables in the model (Neil *et al.*, 2012).

We have used AgenaRisk (2014), a commercial BN package and extended it to incorporate the new BFE algorithm to carry out the experiments described in section 4.

3.2. Dynamic discretisation (DD) on hybrid BNs

Static discretisation has historically been used to approximate the domain of the continuous variables in a BN using predefined, fixed piecewise constant partitions. This approximation will be accurate so long as the posterior high-density region remains in the specified domain during inference. However, the analyst will not know, in advance, which areas of the domain require the greater number of intervals, ultimately resulting in an inaccurate posterior estimate. DD is an alternative discretisation approach that searches for the high-density region during inference and adds more intervals where they are needed while removing intervals where they are not (by merging or deletion). The algorithm iteratively discretises the target variables by the convergence of relative entropy error threshold.

Formally, let X be a continuous node in the BN. The range of X is denoted by Ω_X and the probability density function of X is denoted by f_X. The idea of discretisation is to approximate f_X as follows:

1. Partition Ω_X into a set of interval $\Psi_X = \{w_j\}$.
2. Define a locally constant function f_x on the partitioning intervals.

We estimate the relative entropy error induced by the discretised function using an upper bound of the Kullback–Leibler (KL) metric between two density functions f and g:

$$D(f||g) = \int f(x) \log \frac{f(x)}{g(x)} dx \qquad (5)$$

Under the KL metric the optimal value for the discretised function $\tilde{f}_x$ is given by the mean of the function in each of the intervals of the discretised domain. The discretisation task reduces then to finding an optimal partition set $\hat{\Psi}_x$.

302

DD searches Ω_X for the most accurate specification of the high-density regions given the model and the evidence, calculating a sequence of discretisation intervals in Ω_X iteratively. At each stage in the iterative process, a candidate discretisation, $\Psi_x = \{w_j\}$, is tested to determine whether the relative entropy error of the resulting discretised probability density $\tilde{f}_X$ is below a given threshold, defined according to some stopping rule. After each variable in the model is discretised the inference algorithm, such as JT, calculates the joint posterior distributions for all variables in the model. This gives a new posterior probability density for all variables and these are then re-discretised. This process continues until the stopping rule is triggered.

3.3. BF

The cost of using off-the-shelf BN algorithms to calculate *N-fold* convolution can be computationally expensive. The conditional probability density expression of node T is defined by all of its parent nodes by equation (1):

$$T_n = S_0 + S_1 + \ldots + S_n$$

If each node has a node state of size m and the total number of parents is n, then the CPT for T has a total size of m^{n+1} given the intervals computed under DD. To help reduce the CPT size we employ BF to factorise the BN graph according to the statistical and deterministic functions declared in it.

To illustrate the BF process, we consider constant *n-fold* convolution models for both the independent and common cause case, as represented by BNs $G1$ and $G2$, respectively, in Figure 1. This is just equation (1).

After employing BF, the BNs $G1$ and $G2$ are transformed into $G1'$ and $G2'$, respectively, as shown in Figure 1. In Figure 1 $G1$ shows the *N-fold* convolution when severities are independent and identically distributed. $G2$ denotes the *N-fold* convolution when severities are dependent on a discrete common cause random vector C.
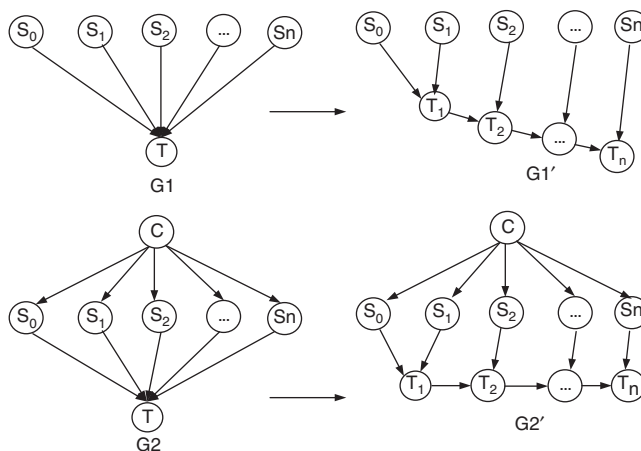

Figure 1. Bayesian Networks models of *N-fold* convolution of i.i.d. severity variables ($G1$) and of common cause version ($G2$) with accompanying binary factorised versions ($G1'$ and $G2'$).

BF ensures that, in the transformed BN, each variable's NPT expression involves a maximum of two continuous parent variables in the transformed BN. This produces a maximal discretised NPT of size m^3.

Theoretical equivalence of $G1$ and $G1'$ with the resulting BN models $G2$ and $G2'$ is demonstrated in Neil *et al.* (2012).

4. BFE

To solve the *N-fold* convolution problem using off-the-shelf BN technology is not possible because we cannot compute $G1$ and $G2$ effectively from the conditional dependency structures defined in Figure 1. This is because, even with BF, either the model size is prohibitively large (in the case of $G1$) or the JT cluster sizes would be exponential in size (as with $G2$). Therefore, the original contribution of this paper is to produce an iterative factorised approach to the computation that scales up to arbitrary sized models. This approach is called *Bayesian Factorisation and Elimination*. This algorithm performs convolution on the hybrid models required to aggregate risk in the presence (or absence) of causal dependencies. This algorithm exploits a number of advances from the field of BNs already described in section 3. We refer to these advances as the BN engine and they are shown in the overall algorithm architecture in Figure 2.

The BFE algorithm contains three separate steps, each performing specific optimisations:

1. Log-based aggregation (LBA): This algorithm computes equation (4), the *n-fold* convolution, in a log-based pattern that can be more efficient than aggregation by straight summation.
2. Variable elimination (VE): Variables are iteratively eliminated during LBA process, by which we can achieve greater computation efficiency for calculating arbitrary constant *n-fold* convolutions.
3. Compound density factorisation (CDF): The compound sum equation (3) can be factorised by this algorithm in order to reduce large NPTs into smaller ones. CDF is similar to BF except that in CDF we introduce one more intermediate variable (a Boolean node) for weighting the compound density combination at each step in the aggregation process.

4.1. LBA

In equation (3) each T_i, $i = 1, ..., n$ is the sum of its parent variables T_{i-1} and S_i, and the aggregation process simply involves repeated summations of the same variable S_i. As BF proceeds intermediate
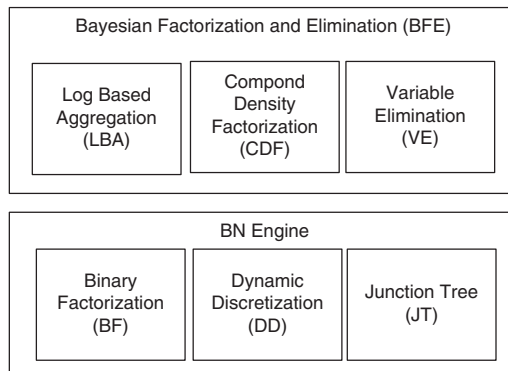

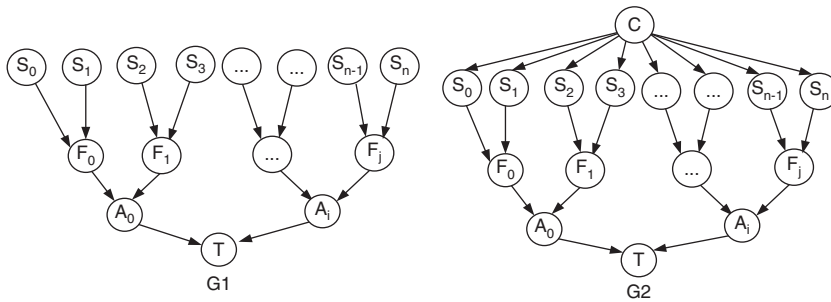**Figure 2.** Architecture of Bayesian Networks algorithms.

304

Figure 3. $G1$ and $G2$ Bayesian Networks binary factorised for aggregation.

variables F_j are created to aggregate every two parents, creating a hierarchy until the total aggregate T is computed. An example, in the presence and absence of common cause vector is shown in Figure 3.

This approach to aggregation is computationally expensive since all the variables are entered and computed in the BN explicitly. LBA simply computes and subsequently reuses prior computed results recursively, so that in each subsequent step we can reuse results from previous steps, without having to create the whole BN. The resulting process is $O(\log_2 n)$.

4.2. VE

The aim of VE is to remove nodes from a BN, G, that do not belong to a query set Q, containing only the variables of interest, by a process of marginalisation. Here we use VE to reduce the number of variables we handle but add additional steps to exploit repeated structure in the BF model. We do not need, therefore, to explicitly manipulate the whole BN, because we are not interested in setting arbitrary query variables or conditioning evidence. Instead, we iterate through the binary factored model, progressively creating subsets of the aggregation hierarchy that can be reused recursively, eliminating nodes and reusing parts as we go.

We first consider a full BF BN and use this to identify variables that can be eliminated and query sets necessary during VE. In the simple case for an *n-fold* convolution for independent i.i.d. severity variables, the graph $G1'$ in Figure 1 denotes the BF form of the computation of $T_n = \sum_{j=0}^{n} S_j$ after introducing the intermediate binary factored variables $\{T_1, T_2, \ldots, T_{n-1}\}$. The marginal distribution for T_n has the form

$$P(T_n) = \sum_{(S_0 \ldots, S_n, T_1, \ldots, T_{n-1})} P(S_0, S_1, \ldots, S_n, T_1, T_2, \ldots, T_{n-1}, T_n)$$

$$= \sum_{(S_0 \ldots, S_n, T_1, \ldots, T_{n-1})} P(T_n | T_{n-1}, S_n) P(T_{n-1} | T_{n-2}, S_{n-1}) \ldots P(T_1 | S_0, S_1) P(S_0) P(S_1) \ldots P(S_n) \quad (6)$$

Exploiting the conditional independence relations in Figure 1.

Notice that every pair of parent variables T_i and S_{i+1} is independent in this model and we can marginalise out each pair of T_i and S_{i+1} from the model separately. Equation (6) can be alternatively expressed as predefined "query blocks":

$$P(T_n) = \sum_{T_{n-1}, S_n} P(T_n | T_{n-1}, S_n) \left\{ \ldots \left\{ \sum_{T_1, S_2} P(T_2 | T_1, S_2) \left\{ \sum_{S_0, S_1} P(T_1 | S_0, S_1) P(S_0) P(S_1) \right\} P(S_2) \right\} \ldots \right\} P(S_n)$$
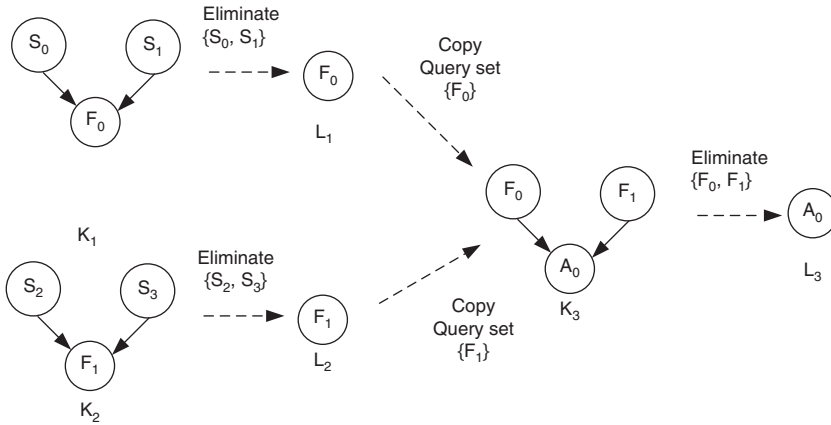
$$(7)$$

305

Figure 4. Variable elimination process applied to part of Bayesian Network *G*1.

So, using equation (7) we can recursively marginalise out, i.e., eliminate or prune, each pair of parents T_i and S_{i+1} from the model. For example, the elimination order in equation (7) could be: $\{S_0, S_1\}$, $\{T_1, S_2\}...\{T_{n-1}, S_n\}$. The marginal distribution of T_n, i.e., the final query set, is then yielded at the last elimination step.

In order to illustrate the recursive BN graph operations, required during VE, consider Figure 1 and BN *G*1. The first few steps involved are shown in Figure 4. At each stage we reuse the same graph structures and expressions for graphs $\{K_1, K_2, K_3\}$ and $\{L_1, L_2, L_3\}$. We can proceed through the BF BN, computing the marginal distributions for the query set, removing elimination sets and repeating the process until we exhaust the variable list.

However, in the case where common cause dependencies are present in the BN, as illustrated by *G*2 in Figure 1, additional care is needed during VE. Here the elimination set does not simply consist of leaf nodes that can be eliminated directly since we have a common parent node *C*, that we want to preserve in the query set at each step. To help highlight how the VE process operates in the presence of common cause variables consider BN *G*2' in Figure 1 and compute the posterior marginal distribution for T_2. The marginal distribution for T_2 has the form equation (8):

$$P(T_2) = \sum_{C,S_0,S_1,S_2,T_1} P(T_2 \,|\, T_1, S_2) P(T_1 \,|\, S_0, S_1) P(S_0 \,|\, C) P(S_1 \,|\, C) P(S_2 \,|\, C) P(C)$$

$$= \sum_{C,S_2,T_1} P(T_2 \,|\, T_1, S_2) P(S_2 \,|\, C) P(C) \left\{ \sum_{S_0,S_1} P(T_1 \,|\, S_0, S_1) P(S_0 \,|\, C) P(S_1 \,|\, C) \right\} \tag{8}$$

We first want to eliminate S_0 and S_1 by marginalising them:

$$P(T_1 \,|\, C) = \sum_{S_0,S_1} P(T_1 \,|\, S_0, S_1) P(S_0 \,|\, C) P(S_2 \,|\, C) \tag{9}$$

The marginal of T_2 can now be expressed along with *C*, T_1 and S_2 alone:

$$P(T_2) = \sum_{C,S_2,T_1} P(T_2 \,|\, T_1, S_2) P(S_2 \,|\, C) P(T_1 \,|\, C) P(C)$$
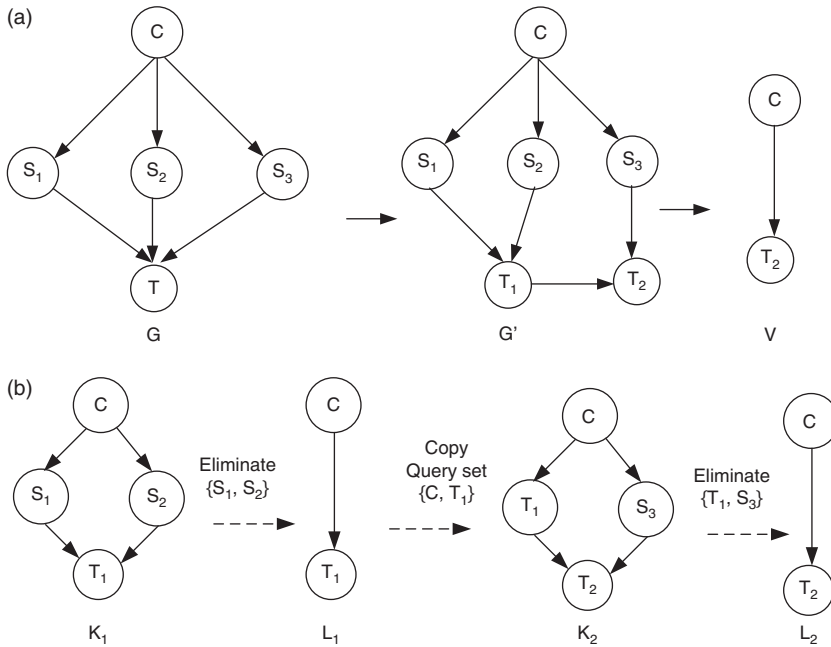
306

Figure 5. (a) Simple common cause model binary factorisation and variable elimination (VE) process. (b) VE process applied to part of Bayesian Network *G*.

Next we eliminate S_2 and T_1:

$$P(T_2 \,|\, C) = \sum_{T_1, S_2} P(T_2 \,|\, T_1, S_2) P(S_2 \,|\, C) P(T_1 \,|\, C) \tag{10}$$

In general, by VE, we obtain the conditional distribution for each variable T_{n-1} (the sum of n severity variables) with the form:

$$P(T_{n-1} \,|\, C) = \sum_{T_{n-2}, S_{n-1}} P(T_{n-1} \,|\, T_{n-2}, S_{n-1}) P(T_{n-2} \,|\, C) P(S_{n-1} \,|\, C) \tag{11}$$

Since equation (11) specifies the conditional distribution for variable $T_{n-1} \,|\, C$ and therefore the posterior marginal distribution for the target *n-fold* convolution T_{n-1} the aggregate total is obtained by marginalising out *C*.

In order to explain the VE algorithm, in terms of graph manipulation, in the common cause case we step through a *3-fold* convolution. Figure 5(a) depicts a *3-fold* convolution model, BF (from *G* to *G'*) and then subject to VE, resulting in reduced the BN *V*. The VE steps are shown in Figure 5(b), which, although operating on subsets of *G*, result in the same graph, i.e., $L_2 = V$.

To calculate the arbitrary *n-fold* convolution in the multiple common cause case it is essential to maintain the structure connecting the common causes in *G'* in every elimination step so that when variables are eliminated any dependencies on common cause variables are correctly maintained. Consequently, the elimination task involves generating the marginal for variable T_j conditional on the set $C = C_0, C_1, \ldots, C_m$. This more general case is shown in Figure 6, with multiple common cause
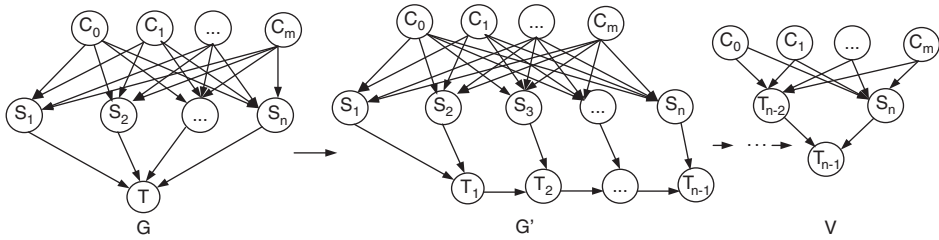
307

Figure 6. Multiple common cause model binary factorisation and variable elimination process (transformed graph name shown below each graph).

variables C_0, C_1,..., C_m, and dependent severity variables S_i. The scheme can be generalised to any configuration of common causes.

4.3. CDF

Recall the compound density expression for an *N-fold* convolution, as given in equation (3), where $T_j = \sum_{i=0}^{j} S_i, j = 0$ (*length of N*) is an *i-fold* convolution with S itself and $a_j = P(N = j)$ is the weighting assigned to the corresponding T_j. Unfortunately, the compound density expression for $P(T)$ is very space inefficient and to address this we need to factorise it. Given each component in the mixture is mutually exclusive, i.e., for a given value of N the aggregate total is equal to 1, and only one T_i, variable, this factorisation is straightforward. However, we cannot use a BF for equation (3), therefore we factorise the compound density expression into pairs of "block nodes" and combine each pair incrementally as shown below.

Equation (3) is factorised as shown in Figure 7, where additional Boolean variables E_j (with only two states *True* and *False*)[1] are introduced to assign weightings proportional to a_j, to each pair of block nodes, i.e., $\{T_0,T_1\}$, $\{F_0,T_2\}$,..., $\{F_{j-2}, T_j\}$. Factor variables F_j are created to calculate the weighted aggregate for each step, up to the length of the *N-fold* convolution L.

The NPT for E_{j-1} is defined by the following:

$$P(E_{j-1} = True) = \frac{a_0 + a_1 + \ldots + a_{j-1}}{a_0 + a_1 + \ldots + a_j} \tag{12}$$

The conditionally deterministic expression for variable F_{j-1} (called a partitioned node in BN parlance) is defined by

$$F_{j-1} = \begin{cases} F_{j-2} & \text{if } E_{j-1} = True \\ T_j & \text{if } E_{j-1} = False \end{cases} \tag{13}$$

Since T_0 and T_1 are mutually exclusive, the marginal distribution for variable F_0 is:

$$F_0 = P(E_0 = True)P(T_0) + P(E_0 = False)P(T_1) = a_0 P(T_0) + a_1 P(T_1)$$

which is identical to the first two terms in the original compound density expression (equation (3)). Similarly, the marginal for variable F_j becomes:

$$F_{j-1} = P(E_{j-1} = True)P(F_{j-2}) + P(E_{j-1} = False)P(T_j) \tag{14}$$

---

[1] "True" and "False" are used for convenience; any binary labelling would do equally well.
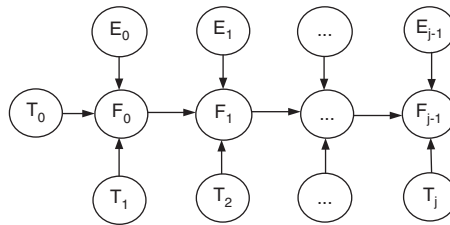
Figure 7. Compound density factorisation.

After applying the CDF method to equation (3) we have the marginal for F_{j-1}, as shown by (14), which yields the compound density, $P(T)$, for the *N-fold* convolution. Therefore, by using the CDF method we can compute the compound density (equation (3)) more efficiently.

The CDF method is a general way of factorising a compound density. It takes as input any *n-fold* convolution, regardless of the causal structure governing the severity variables. Note that the CDF method can be made more efficient by applying VE to remove leaf nodes. Likewise, we can execute the algorithm recursively reuse the same BN fragment $P(F \mid F, T, E)$.

4.4. The BFE convolution algorithm

The BFE convolution algorithm is formalised, as pseudo code, in Table 1.

5. Deconvolution using the BFE Algorithm

5.1. Deconvolution

Where we are interested in the posterior marginal distribution of the causal variables conditional on the convolution aggregated results we can perform deconvolution, in effect reversing the direction of inference involved in convolution. This is of interest in sensitivity analysis, where we might be interested in identifying which causal variables have the largest, differential, impact on some summary statistic of the aggregated total, such as the mean loss or the conditional value at risk, derived from $P(C \mid T > t_0)$.

One established solution for deconvolution involves inverse filtering using Fourier transforms, whereby the severity S is obtained by inverse transformation from its characteristic function. Alternative analytical estimation methods, i.e., maximum likelihood and numerical evaluation involving Fourier transforms or simulation-based sampling methods, can be attempted but none of them is known to have been applied to *N-fold* deconvolution in hybrid models containing discrete causal variables.

BN-based inference offers an alternative, natural, way of solving deconvolution because it offers both predictive (cause to consequence) reasoning and diagnostic (consequence to cause) reasoning. This process of backwards inference is called "back propagation", whereby evidence is entered into the BN on a consequence node and then the model is updated to determine the posterior probabilities of all parent and antecedent variables in the model. A "backwards" version of the BFE algorithm offers a solution for answering deconvolution problems in a general way without making any assumptions about the form of the density function of S. The approach again uses a discretised form for all continuous variables in the hybrid BN, thus ensuring that the frequency distribution N is identifiable.

Table 1. *Bayesian Factorisation and Elimination* convolution algorithm.

Input: S: Severity variable, N: Frequency variable, C: vector of common causes (optional)
Output: Compound density T
Main:
1. Compute the probability density function of N, with sample space Z by:
    $f_N(x) = P(N = x) = P(\{z_j \in Z : N(z) = x\}) = a_j, j = 0,1,...,length(Z)$
2. **for** $j = 0$ to (length of Z) **do**
3.   **for** $i = 0$ to z_j **do**
4.   Compute z_j-*fold* convolution $T_{z_j} = \sum_{i=0}^{z_j} S_i$ by **BF** and **LBA** algorithms
5.   Eliminate nodes (out of query set) by **VE** algorithm
6.   **end for**
7.   **While** $j \geq 2$ **do**
8.   Apply **CDF** algorithm to factorize (2.3) by probability density of N
            Compute $F_{j-1} = P(E_{j-1} = True)P(F_{j-2}) + P(E_{j-1} = False)P(T_{zj})$
9.   Eliminate nodes S_i, F_{j-2} and T_{zj} by **VE** algorithm
10.   **end while**
11.   **end for**
12. **return** $P(F_{j-1})$ {marginal distribution of T}

Example 1 Consider a simple BN with parent variable distributions $X \sim$ Normal ($\mu = 5$, $\sigma^2 = 5$), $Y \sim$ Normal ($\mu = 10$, $\sigma^2 = 10$) and likelihood function for a child variable $P(Z \mid X,Y) = P(Z = X + Y)$. Figure 8(a) shows the prior convolution effects of the back propagation calculation, as marginal distributions superimposed on the BN graph. The exact posterior marginal for Z is $Z \sim$ Normal ($\mu = 15$, $\sigma^2 = 15$). Our approximation produces a mean of 14.99 and variance 16.28.

If we set an observation $Z = z_0$ and perform inference we obtain the posterior marginal of X by Bayes' rule:

$$P(X \mid Z = z_0) = \frac{P(X, Z = z_0)}{P(Z = z_0)} = \frac{\sum_{Y} P(Z = z_0 \mid X, Y)P(X)P(Y)}{\sum_{X,Y} P(Z = z_0 \mid X, Y)P(X)P(Y)} \quad (15)$$

Where our likelihood $P(Z \mid X,Y)$ is a convolution function, equation (15) defines the deconvolution and yields the posterior marginal distribution of X given observation $Z = z_0$. In Figure 8(b), the observation is $Z = 30$ (which is approximated as a discrete bin of given width), and the posterior for X has updated to a marginal distribution with mean 9.97 and variance 3.477.

In the example shown in Figure 8 the parent variables X and Y are conditionally dependent given the observation $Z = z_0$. For *n-fold* convolution with or without common causes an observation on the T_i variables would also make each of the severity variables dependent and we can perform *n-fold* deconvolution using the DD and JT alone for small models containing non i.i.d severity variables with query block sizes of maximum cardinality four. For large models, containing i.i.d severity variables BFE provides a correct solution with minimal computational overhead.

We have already noted that during *N-fold* convolution the T_i variables are mutually exclusive, such that for a given $N = i$, if the variable T_i exists, then the other variables do not. This fact can be exploited during factorisation during the deconvolution processes.

Consider the common cause BN model shown in Figure 9. The fully specified model is shown in BN graph A. The posterior distributions for all nodes can be computed by way of the BFE convolution
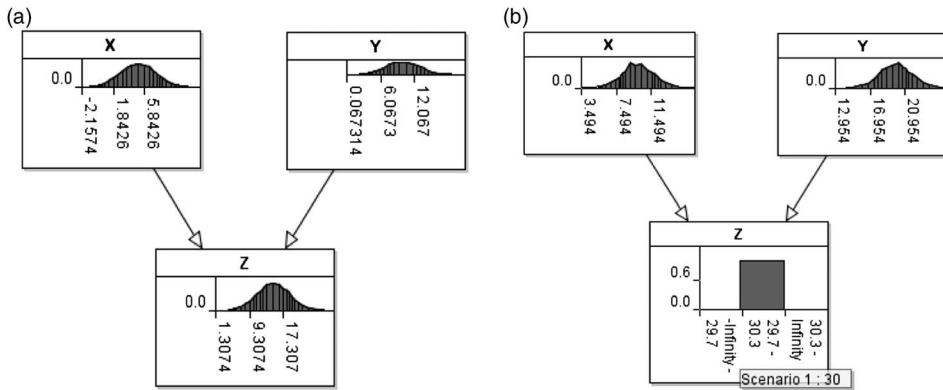
310

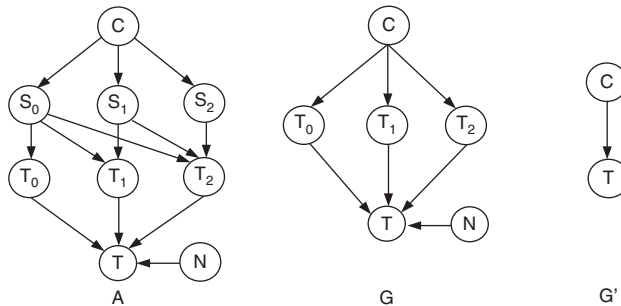**Figure 8.** (a) Convolution and (b) deconvolution.

Figure 9. Binary factored common cause *N-fold* Bayesian Network, *A*, reduced by applying the variable elimination algorithm to *G* and then *G'*.

algorithm and we can cache any distributions and parameters we might need during this process, for subsequent use during deconvolution. The BFE deconvolution algorithm then proceeds by eliminating all intermediate, frequency and severity variables until we get the reduced BN graph containing the final query set of interest.

Let us assume the model structure in BN *A* of Figure 9. Here frequency *N* is discretised into three finite states {1, 2, 3}, so there are three *n-fold* convolution variables $T_i = \sum_{j=0}^{i} S_j, i = 0, 1, 2$ each corresponding to the sum of one, two and three severity variables. *T* is the compound distribution defined by:

$$T = a_0 P(T_0) + a_1 P(T_1) + a_2 P(T_2), a_i = P(N = i), i = 0, 1, 2$$

Given evidence $T = t_0$ the deconvolution of *C* is achieved by:

$$P(C|T = t_0) = \frac{P(C, T = t_0)}{P(T = t_0)}$$

$$= \sum_{S_i, T_i, N} P(T = t_0 | \mathrm{pa}(T)) P(N) P(T_i | \mathrm{pa}(T_i)) P(S_i | C) P(C) \qquad (16)$$

where pa(*T*) denotes the parents of *T*. So, once the convolution model has eliminated all irrelevant variables, in this case S_i, T_{zj}, E_j, F_j we would be left with the query set, which here is $Q = \{C, T\}$.

311

5.2. Reconstructing the frequency variables during deconvolution

If we are also interested in including the frequency variable N in our query set we must be careful to cache variables E_j, F_{j-2} and T_{zj} during convolution. Recall that the prior distribution for N was decomposed into the E_j during CDF, therefore we need some way of updating this prior using the new posterior probabilities generated on the Boolean variables E_j during deconvolution. To perform deconvolution on N it is first necessarily to reconstruct N from the E_j variables that together composed the original N.

Reconstruction involves composing all Boolean variables E_j into the frequency variable N, in a way that the updating of E_j can directly result in generating a new posterior distribution of N. The model is established by connecting all E_j nodes to N, where the new NPT for N has the form of combining all its parents. However, it turns out this NPT is exponential (2^{j+1}) in size. To avoid the drawback we use an alternative, factorised, approach that can reconstruct the NPT incrementally.

As before, we reconstruct N using BF where the conditioning is conducted efficiently using incremental steps. Here the intermediate variables produced during BF N_k ($k = 0,...,j-1$) are created efficiently by ensuring their NPTs are of minimal size.

The routine for constructing the NPTs for N_k ($k = 0,...,j-1$) from the E_j's is:

1. Order parents E_j and E_{j-1} from higher index to lower index for N_k's NPT (since E_j is Boolean variable with only two states, one concatenating all E_{j-1}'s states and another state is single state that E_{j-1} does not contain. In this example E_1 should be placed on top of E_0 in the NPT table, as it is easier for comparing the common sets).

2. As we have already generated the NPT map of E_j, E_{j-1} and N_k. Next, we specify the NPT entry with unit value ("1") at $N_k = \tau$, when E_j and E_{j-1} has common sets τ (in this example, E_1 and E_0 have common sets τ = "0" and τ = "1").

3. Specify NPT entry with value ("1") at $N_k = \tau$, when E_j and E_{j-1} have no common sets and $E_j = \tau$ (E_j has one state τ that E_{j-1} does not contain, so under this case N_k only needs to be consistent with E_j as the changes on E_{j-1} would not affect the probability $P(N_k = \tau)$, in this example it is when $E_1 = \tau$ = "2").

4. Specify NPT entry with value ("0") at all other entries.

We repeat this routine for all N_k ($k = 0,...,j-1$) until we have exhausted all E_j's, producing a fully reconstructed N. Once we have built the reconstructed structure (N_k) for N, in fact the updates of E_j's probabilities are directly mapped to N_k, and so deconvolution of N is retrieved.

5.3. The BFE deconvolution algorithm with examples

The BFE deconvolution algorithm for *N-fold* deconvolution is formalised as pseudo code in Table 2.

Example 2 Consider a simplified example for deconvoluting N, suppose frequency distribution N is discretised as {0.1, 0.2, 0.3, 0.4} with discrete states {0, 1, 2, 3} and $S \sim$ Exponential(1). Figure 10(a) shows these incremental steps for example 4. In this example there are three parents (E_0, E_1, E_2) to N. The incremental composition steps of E_j have introduced two intermediate variables N_0 and N_1, and we expect the frequency N to be reconstructed at the end of the incremental step, which is variable N_1. Key to this process is how to build the NPT for each N_k.

312

Table 2. *Bayesian Factorisation and Elimination* deconvolution algorithm.

Input: S: Severity variable, N: Frequency variable, C: vector of common causes and $T = t_0$
Output: posterior marginal of query set members i.e. $P(C \mid T = t_0)$, $P(N \mid T = t_0)$
Main:
1. **do** convolution BFE algorithm to produce final query set
2. **if** N is in query set
3.   **reconstruct** N from E_j
4. **end if**
5. **set** evidence t_0 on T and perform inference
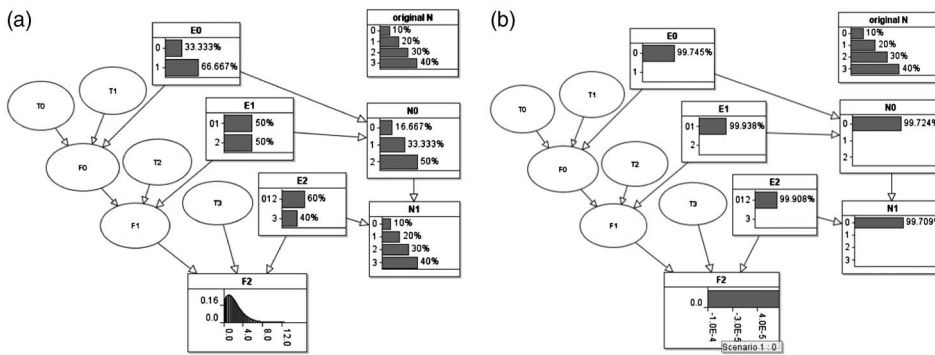6. **return** posterior marginal distributions for query set

Figure 10. (a) Reconstruct N and (b) deconvoluting N.

Table 3. The node probability table of N_0.

E_1	"01"		"2"	
E_0	"0"	"1"	"0"	"1"
$N_0 = 0$	1.0	0.0	0.0	0.0
$N_0 = 1$	0.0	1.0	0.0	0.0
$N_0 = 2$	0.0	0.0	1.0	1.0

Table 3 illustrates the NPT of N_0, where it composes E_0 and E_1 successively, in such a way that each N_k contains all and only its parents' discrete states. So N_0 has the discrete distribution on "0", "1" and "2".

Figure 10(b) shows the deconvolution of N by our reconstruction process. The reconstructed prior distribution of N_1 is identical to node "original N" (shown in Figure 10(a)) as we expected. After setting an observation value "0" at the compound sum variable F_2 we have queried that the posterior of N is 99.7% probability at state "0", since at state zero it has all possibility of generating a zero compound sum at F_2.

The reconstruction theme is applicable to cases that N has discrete parent cause variables as well, where E_j's NPTs are generated directly from N's parents, and the deconvolution is performed by BFE deconvolution algorithm. Experiment 3 in section 6 considers deconvoluting common cause variables where the model has this form.

6. Experiments

We report on a number of experiments using the BFE algorithm in order to determine whether it can be applied to a spectrum of risk aggregation problem archetypes. Where possible the results are compared to analytical results, FFT, Panjer's approach and MC simulation. The following experiments, with accompanying rationale, were carried out:

1. *Experiment 1*: *Convolution with multimodal (mixtures of) severity distribution*. We believe this to be a particularly difficult case for those methods that are more reliant on particular analytical assumptions. Practically, multimodal distributions are of interest in cases where we might have extreme outcomes, such as sharp regime shifts in asset valuations.

2. *Experiment 2*: *Convolution with discrete common causes variables*. This is the key experiment in the paper since these causes will be co-dependent and the severity distribution will depend on their values (and hence will be a conditional mixture).

3. *Experiment 3*: *Deconvolution with discrete common causes*. This is the inverse of experiment 2 where we seek to estimate the posterior marginal for the common causes conditioned on some observed total aggregated value.

The computing environment settings for the experiments are as follows. Operation system: Windows XP Professional, Intel i5 @ 3.30 GHz, 4.0 GB RAM. AgenaRisk was used to implement the BFE algorithm, which was written in java, where typically the DD settings were for 65 iterations for severity variables and 25 iterations for the frequency variable. A sample size of $2.0E + 5$ was used as the settings in R (2013) for the MC simulation.

6.1. Experiment 1: convolution with multimodal severity distribution

Here we set the event frequency as $N \sim \text{Poisson}(50)$ but the severity distribution is a mixture distribution, $S \sim f_S$:

$$f_S = (0.2)\text{Gamma}(5,\ 1.5) + (0.3)\text{Normal}(25,\ 2) + (0.4)\text{Normal}(50,\ 3) + (0.1)\text{Gamma}(100,\ 2)$$

In a hybrid BN a mixture distribution is modelled by conditioning the severity variable on one or more partitioning discrete variables C. Assuming that severity variables S_j are i.i.d. we can calculate the compound density using BFE.

The characteristic function of a mixture distribution is inconvenient to define (with continuous and discrete components). The analytical and programming effort needed to solve each multimodal severity distribution for Panjer is high, so here we compare with MC only. (Table 4).

The corresponding marginal distribution for the query node set $\{T, N, S\}$ is shown in Figure 11.

6.2. Experiment 2: convolution with discrete common causes variables

Loss distributions from operational risk can vary in different circumstances, e.g., exhibiting co-dependences among causes. Suppose in some cases that losses are caused by daily operations and these losses are drawn from a mixture of truncated Normal distributions, whereas extreme or some unexpected losses are distributed in a more severe distribution. We model this behaviour by a hierarchical common cause combination $C_0, \ldots, C_4$.

The severity variable S is conditioning on common cause variable, C_0, C_1, C_2. And these common cause variables are conditioned on higher common causes C_3 and C_4. Severity NPT is shown in Table 5. The frequency distribution of losses is modelled as $N \sim \text{Poisson}(50)$.

314

Table 4. Results of convolution with multimodal severity distribution.

Algorithm	Mean	s.d.	95th percentile	99th percentile	Analysis Effort
MC	2,444.8	516.7	3,340.0	3,787.7	Low
BFE	2,441.1	523.3	3,341.5	3,783.1	Low

MC, Monte Carlo; BFE, *Bayesian Factorisation and Elimination*.
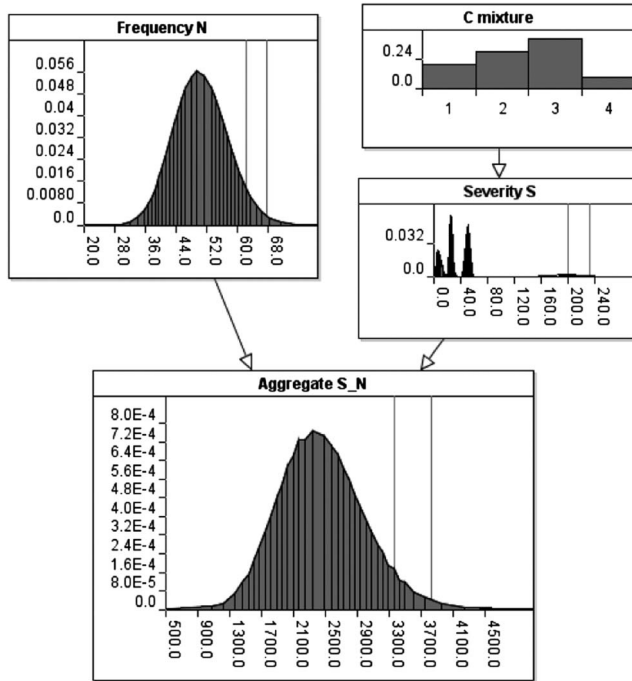

Figure 11. Marginal distributions for overlaid on Bayesian Network graph containing query nodes for Experiment 1.

Table 5. Severity node probability table.

C_0	High				Low			
C_1	High		Low		High		Low	
C_2	High	Low	High	Low	High	Low	High	Low
Expression	Normal (1,2)	Normal (2,3)	Normal (3,4)	Normal (4,5)	Normal (100,110)	Normal (110,120)	Normal (120,130)	Normal (130,140)

In Figure 12(a) the model severities with dependencies by common cause variables $C_0,\ldots, C_4$ is introduced. Figure 12(b) depicts a 16-*fold* convolution of dependent severities using the VE method. For any given frequency distribution N we can apply the BFE convolution algorithm to calculate the common cause *N-fold* convolution.
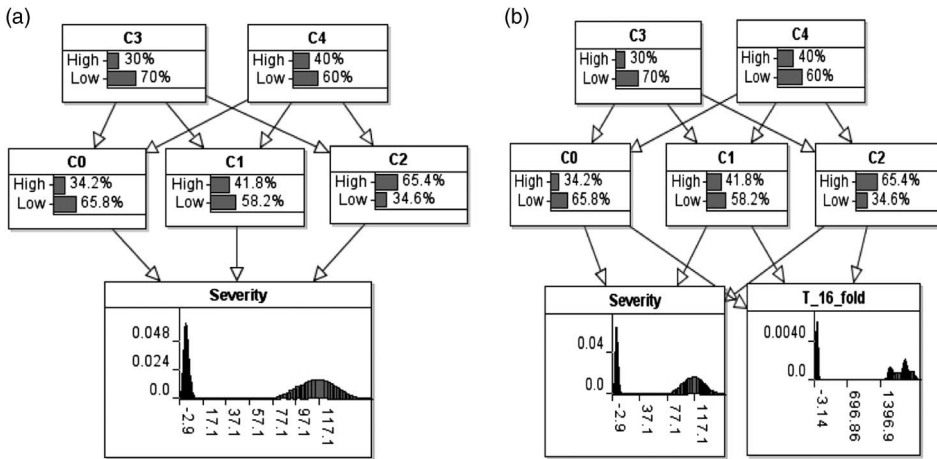
Figure 12. (a) Common cause-dependent severity; (b) 16-*fold* convolution of dependent severity.
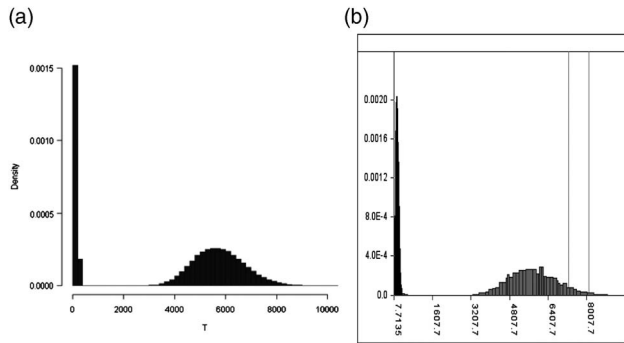

Figure 13. Compound densities (a) Monte Carlo; (b) *Bayesian Factorisation and Elimination*.

Table 6. Common cause *N-fold* convolution density.

Algorithm	Mean	Median	s.d.	95th percentile	99th percentile
MC	3,831	5,017	2,784	7,215	8,023
BFE	3,871	5,052	3,255	7,267	8,115

MC, Monte Carlo; BFE, *Bayesian Factorisation and Elimination*.

Figure 13 illustrates the output compound densities for the compared algorithms. Table 6 shows that results for the two approaches are almost identical on summary statistics except the small difference on standard deviation. BFE has offered a unified approach to construct and compute such a model conveniently.

6.3. Experiment 3: deconvolution with discrete common causes variables

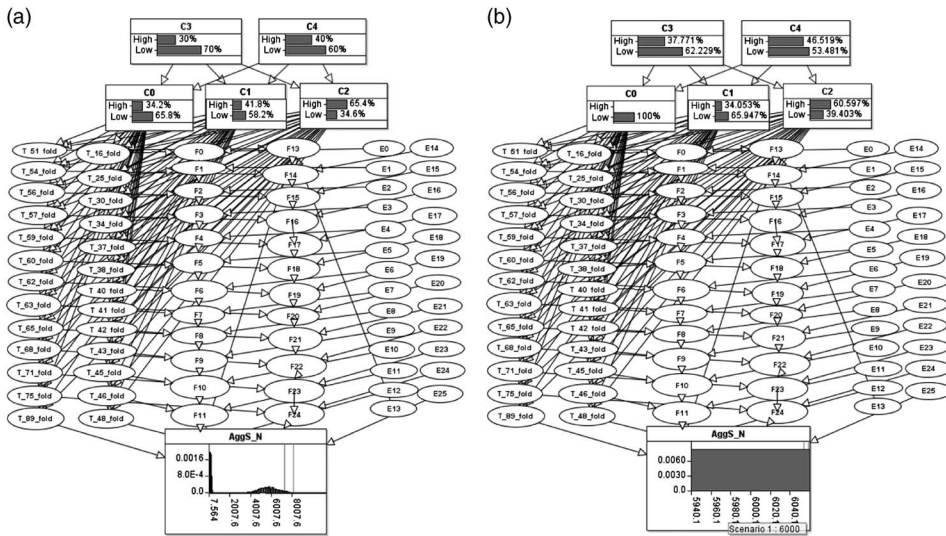We reuse the convolution model from experiment 2 as the input model for deconvolution (Figure 14).

316

Figure 14. (a) Common cause *N-fold* convolution using *Bayesian Factorisation and Elimination* algorithm. (b) *N-fold* model deconvolution using BFE algorithm. (The intermediate variables in this example are shown for reference despite them being eliminated during the convolution process).

Figure 14(b) sets an observation on total aggregation node *AggS_N*. After performing deconvolution we queried the posterior marginal of common causes and diagnose that the most likely common cause is C_0, which is in its "Low" state with certainty. This is easily explained by the fact that from the severity NPT, shown in Table 5, it is only when state of C_0 is "Low" that a value of 6,000 can be at all probable. This deconvolution is currently only supported by BEF since the information cannot be back retrieved by other approaches.

Deconvolution is obviously useful in carrying out a sensitivity of the model results, allowing the analyst to quickly check model assumptions and identify which causal factors have the largest consequential effect on the result. This is difficult to do manually or informally in the presence of non-linear interactions. Also, without "backwards" deconvolution we can only compute such sensitivities "forwards" one casual variable at a time and this is computationally much more expensive. For example, the forwards calculation of *T* from ten Boolean common cause variables would require 2^{10} calculations versus 40 in the backwards case (assuming *T* was discretised into 40 states).

7. Conclusion and Future Work

This paper has reviewed historical, popular, methods for performing risk aggregation and compared them with a new method called *Bayesian Factorisation and Elimination*. The method exploits a number of advances from the field of Bayesian Networks, covering methods to approximate statistical and conditionally deterministic functions and to factorise multivariate distributions for efficient computation. Our objective for BFE was for it to perform aggregation for classes of problems that the existing methods cannot solve (namely hybrid situations involving common causes) while

performing at least as well on conventional aggregation problems. Our experiments show that our objectives were achieved. For more difficult hybrid problems the experimental results show that BFE provides a more general solution that is not possible with the previous methods.

In addition, the BFE approach can be easily extended to perform deconvolution for the purposes of stress testing and sensitivity analysis in a way that competing methods cannot currently offer. The BFE deconvolution method reported here provides a low-resolution result, which is likely good enough for the purposes of model checking and sensitivity analysis. However, we are investigating an alternative high-resolution approach whereby variables are discretised efficiently during the deconvolution process, thus providing more accurate posterior results.

Ongoing and future research is also focused on more complex situations involving both copulas and common cause variables. The challenge here is to decompose these models into lower-dimensional joint distributions, where complexity can be further reduced by factorisation. One final area of interest includes optimisation of the results such that we might choose set of actions in the model that maximise returns for minimum risk: we see deconvolution playing a strong role here.

Acknowledgements

References

AgenaRisk. (2014). AgenaRisk. Available online at the address http://www.agenarisk.com/ (accessed 23 December 2014).

Arbenz, P. & Canestraro, D. (2012). Estimating copulas for insurance from scarce observations, expert opinion and prior information: a Bayesian approach. *ASTIN Bulletin*, **42**(1), 271–290.

Black, F. & Litterman, R.B. (1991). Asset allocation: combining investor views with market equilibrium. *The Journal of Fixed Income*, **1**(2), 7–18.

Brechmann, E.C. (2014). Hierarchical Kendall copulas: properties and inference. *Canadian Journal of Statistics*, **42**(1), 78–108.

Bruneton, J.-P. (2011). Copula-based hierarchical aggregation of correlated risks. The behaviour of the diversification benefit in Gaussian and lognormal trees. arXiv:1111.1113 [q-Fin]. Retrieved from http://arxiv.org/abs/1111.1113 (accessed 20 February 2014).

Cooper, G.F. & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, **9**(4), 309–347.

Cowell, R.G., Verrall, R.J. & Yoon, Y.K. (2007). Modeling operational risk with Bayesian networks. *Journal of Risk and Insurance*, **74**(4), 795–827.

Embrechts, P. (2009). Copulas: a personal view. *Journal of Risk and Insurance*, **76**(3), 639–650.

Fishman, G. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. New York: Springer.

Fenton, N. & Neil, M. (2012). *Risk Assessment and Decision Analysis with Bayesian Networks*. Boca Raton, Florida: CRC Press.

Heckman, P.E. & Meyers, G.G. (1983). The calculation of aggregate loss distributions (pp. 22–61). Presented at the Proceedings of the Casualty Actuarial Society. Retrieved from http://casact. net/pubs/proceed/proceed83/83022.pdf (accessed 23 December 2013).

IMF. (2009). *IMF global financial stability report – navigating the financial challenges ahead*. Washington, DC.

318

Jensen, F.V. & Nielsen, T.D. (2009). *Bayesian Networks and Decision Graphs*. New York: Springer.

Laeven, L. & Valencia, F. (2008). *Systemic Banking Crises: A New Database (SSRN Scholarly Paper No. 1278435)*. Social Science Research Network, Rochester, NY.

Lauritzen, S.L. (1996). *Graphical Models*. Oxford University Press.

Marquez, D., Neil, M. & Fenton, N. (2010). Improved reliability modeling using Bayesian networks and dynamic discretization. *Reliability Engineering & System Safety*, **95**(4), 412–425.

McNeil, A.J., Frey, R. & Embrechts, P. (2010). *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton, New Jersey: Princeton University Press.

Meucci, A. (2008). *Fully Flexible Views: Theory and Practice (SSRN Scholarly Paper No. 1213325)*. Social Science Research Network, Rochester, NY.

Neil, M., Chen, X. & Fenton, N. (2012). Optimizing the calculation of conditional probability tables in hybrid Bayesian networks using binary factorization. *IEEE Transactions on Knowledge and Data Engineering*, **24**(7), 1306–1312.

Neil, M. & Fenton, N. (2008). Using Bayesian networks to model the operational risk to information technology infrastructure in financial institutions. *Journal of Financial Transformation*, **22**, 131–138.

Neil, M., Tailor, M. & Marquez, D. (2007). Inference in hybrid Bayesian networks using dynamic discretization. *Statistics and Computing*, **17**, 219–233.

Nelsen, R.B. (2007). *An Introduction to Copulas*. New York: Springer.

Panjer, H.H. (1981). Recursive evaluation of a family of compound distributions. *ASTIN Bulletin*, **1**(12), 22–26.

Pearl, J. (1993). [Bayesian analysis in expert systems]: comment: graphical models, causality and intervention. *Statistical Science*, **8**(3), 266–269.

Politou, D. & Giudici, P. (2009). Modelling operational risk losses with graphical models and copula functions. *Methodology and Computing in Applied Probability*, **11**(1), 65–93.

R. (2013). The R Project for Statistical Computing. Available online at the address http://www.r-project.org/ (accessed 23 December 2013).

Rebonato, R. (2010). *Coherent Stress Testing: A Bayesian Approach to the Analysis of Financial Stress*. West Sussex, UK: John Wiley & Sons.

Spiegelhalter, D.J. & Lauritzen, S.L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, **20**(5), 579–605.