Computing Sobol Indices in Probabilistic Graphical Models

Rafael Ballester-Ripoll and Manuele Leonelli

School of Human Sciences & Technology, IE University, Madrid, Spain

Abstract

We show how to apply Sobol's method of global sensitivity analysis to measure the influence exerted by a set of nodes' evidence on a quantity of interest expressed by a Bayesian network. Our method exploits the network structure so as to transform the problem of Sobol index estimation into that of marginalization inference and, unlike Monte Carlo based estimators for variance-based sensitivity analysis, it gives exact results when exact inference is used. Moreover, the method supports the case of correlated inputs and it is efficient as long as eliminating the inputs' ancestors is computationally affordable. The proposed algorithms are inspired by the field of tensor networks and generalize earlier tensor sensitivity techniques from the acyclic to the cyclic case. We demonstrate our method on three medium to large Bayesian networks in the areas of structural reliability and project risk management.

Keywords:
Global sensitivity analysis, Bayesian networks, Sobol indices, Uncertainty quantification, Tensor networks

1. Introduction

Assessing the effect of the input values of a mathematical model on its outputs is a fundamental but often overlooked step of any real-world analysis. The set of techniques to perform such a critical task is usually known as *sensitivity analysis*. There is an increasing recognition of the importance of performing sensitivity analysis after the definition of a mathematical model and there are now a wide array of recent reviews of the methods available (e.g. Borgonovo and Plischke, 2016; Borgonovo, 2017; Razavi et al., 2021; Saltelli et al., 2019).

Probabilistic graphical models and specifically Bayesian networks (BNs) (e.g. Darwiche, 2009; Koller et al., 2009) are a class of models that are widely used for risk assessment of complex operational systems in a variety of domains (e.g. Bielza and Larrañaga, 2014; Cai et al., 2018; Drury et al., 2017; McLachlan et al., 2020). The main reason of their success is that they provide an efficient as well as intuitive framework to represent the relationship existing between a vector of variables of interest using a simple graph. Their use to assess

the reliability of engineering systems is becoming increasingly popular. BNs have been applied in areas such as shipping (Zhou et al., 2022; Yu et al., 2021; Goerlandt and Islam, 2021; Ung, 2021), hydraulic fracturing (Zio et al., 2022), constructions (Ji et al., 2022), oil and gas systems (Dimaio et al., 2021; Liu et al., 2021; Zhang and Weng, 2020), railway and aviation transport (Huang et al., 2020; Zhang and Mahadevan, 2021), turbines (Quintanar-Gago et al., 2021) and space launches (Pan et al., 2021), among others. Sensitivity analysis is a critical step in all of the above-cited works to assess the importance of various risk factors and to evaluate the overall safety of the system under study. However, such a critical step is usually carried out by using either some ad-hoc procedures or some of the local methods reviewed next.

Standard sensitivity analysis in BNs can be broken down into two main steps. First, some parameters of the model are varied and the effect of these variations on output probabilities of interest are investigated. For this purpose, a simple mathematical function, usually termed *sensitivity function*, describes an output probability of interest as a function of the BN parameters (Castillo et al., 1997; Coupé and Van Der Gaag, 2002). Second, once parameter variations are identified, the effect of these is summarized by a distance or divergence measure between the original and the varied distributions underlying the BN, most commonly the Chan-Darwiche distance (Chan and Darwiche, 2005) or the well-known Kullback-Leibler divergence.

As noticed by Rohmer (2020), all the above-mentioned methods are local since they most often measure the effect of a parameter variation on an output of interest, while the other parameters are kept fixed. Multi-way sensitivity analyses, where more than one parameter is varied contemporaneously, have been proposed (e.g. Bolt and Renooij, 2014; Chan and Darwiche, 2004; Kjærulff and van der Gaag, 2000; Leonelli et al., 2017; Leonelli and Riccomagno, 2019), but these quickly become computationally intractable.

Outside of the realm of BNs, the most common approach is the so-called *global sensitivity analysis* (Saltelli et al., 2000, 2008) which provides a "global", instead of local, representation of how different factors jointly influence some function of the model's output. Here we bridge this gap by introducing global sensitivity methods for probabilistic graphical models entailing the computation of Sobol indices (Sobol, 1990) for variables associated to vertices of a network. To our knowledge, the only other attempt in introducing global sensitivity methods for BNs was proposed by Li and Mahadevan (2017). However, their approach suffers of a number of drawbacks: i. it only computes Sobol indices for one vertex of the network; ii. it is computationally very expensive and requires a complex Monte Carlo simulation; iii. its implementation is not freely available.

The development of more accurate and faster methods for the computation of Sobol indices is a very active area of research, which has far-reaching consequences for the reliability of real-world systems. Recent contributions include the use of lambda surrogate models (Zhu and Sudret, 2021), polynomial chaos expansions (Mara and Becker, 2021), Latin hypercube designs (Damblin and Ghione, 2021), randomized quasi-Monte Carlo methods (Ökten and Liu, 2021) and random forests (Antoniadis et al., 2021). In practice, the most

commonly used technique remains Monte Carlo approximated methods, which are still the subject of ongoing research (e.g. Azzini et al., 2021). Here, we take a different approach and develop *exact* methods for the computation of Sobol indices based on *tensor networks* (TNs; see e.g. Ye and Lim, 2018).

In more details, our method takes advantage of the similarities between TNs and probabilistic graphical models and is part of an ongoing effort to foster interactions between these two communities (Robeva and Seigal, 2018). Like probabilistic networks, TNs have long had an important role in knowledge discovery and mathematical modeling, and are experiencing a surge in popularity along with the latest trends in machine learning. Although topologies admitting cycles are much more expressive, much of the extant TN literature focuses on learning acyclic structures from observable multidimensional arrays (tensors). Further, the Sobol indices of an acyclic tensor networks can be computed efficiently, especially in the particular case of the *linear tensor network* or *tensor train* (Rai, 2014; Zhang et al., 2015; Ballester-Ripoll et al., 2018, 2019). Our contribution is *to make exact Sobol sensitivity analysis possible in the case where the function of interest is given by a BN*; note that such networks can only be cast as tensor networks by introducing cycles (moralization). With this work we develop the first theoretically-grounded method to perform the critical step of global sensitivity analysis in BNs, which, as noted, are nowadays one of the most commonly used models in the field of reliability. Furthermore, our implementation of the methods is freely available in Python[1], contributing to the recent effort of promoting global sensitivity analysis (Douglas-Smith et al., 2020).

Importantly, by taking advantage of the underlying structure of the BN, the methods proposed here can formally account for correlated inputs, although some restrictions need to be imposed. Global sensitivity methods usually ignore input correlations which can greatly falsify the outcome of the sensitivity analysis (Do and Razavi, 2020), although some recent proposals attempt to embed correlated inputs (e.g. Chastaing et al., 2012; Iooss and Prieur, 2019; Sheikholeslami et al., 2020).

1.1. Notation

We use the following throughout the paper:

- The capital Y represents random variables, y their realizations and $\mathbb{Y}$ their sample spaces.
- Bold letters for vectors, e.g. $\boldsymbol{y}$, $\boldsymbol{Y}$, and blackboard bold letters for sets, e.g. $\mathbb{A}$, $\mathbb{B}$.
- $[n]$ denotes $\{1, \ldots, n\}$ and $\mathcal{P}$ the power set.
- For a subset $\mathbb{A} \subset [n]$, we denote $\boldsymbol{Y}_{\mathbb{A}} = (Y_i)_{i \in \mathbb{A}}$ and $\mathbb{Y}_{\mathbb{A}} = \times_{i \in \mathbb{A}} \mathbb{Y}_i$.

[1] Available at `https://github.com/rballester/bnsobol`.

- Calligraphic letters are used for graphical models: $\mathcal{N}, \mathcal{B}$ and $\mathcal{M}$ for Bayesian networks, Markov random fields and tensor networks, respectively.
- If $\mathcal{M}$ is a graphical model and $\mathbb{A} \subset [n]$, $\mathcal{M}(\boldsymbol{y}_{\mathbb{A}})$ is the model that results after setting $\boldsymbol{Y}_{\mathbb{A}} = \boldsymbol{y}_{\mathbb{A}}$.
- A subscript, as in $\mathcal{M}_{\mathbb{A}}$, denotes variable elimination, i.e. removing $\boldsymbol{Y}_{\mathbb{A}}$ from a model $\mathcal{M}$.
- The backslash ($\setminus$) denotes the absence of an element in a set. We use it whenever the set in question is clear from the context; for example, for a network $\mathcal{M}$ on variables $[n]$, $\mathcal{M}_{\setminus i}$ denotes removing $\boldsymbol{Y}_{[n]\setminus\{i\}}$.

2. Sobol Indices

The method of Sobol (1990) is one of the most powerful frameworks for global sensitivity analysis of a general function $f : \Omega \subseteq \mathbb{R}^k \to \mathbb{R}$, whose domain is usually restricted to a k-dimensional rectangle $\Omega = \times_{i\in[k]}\Omega_i$. In our case, as formalized in Section 3, Ω is assumed to be a discrete Cartesian product.

There are multiple kinds of Sobol indices, each with different nuances of interpretation; next, we review the two most important ones.

2.1. Variance Components

The *variance component* of the variable Y_i is defined as

$$S_i := \frac{\mathrm{Var}_i\left[\mathrm{E}_{\setminus i}[f]\right]}{\mathrm{Var}[f]}, \tag{1}$$

where $\mathrm{E}_{\setminus i}$ is the expectation with respect to $\boldsymbol{Y}_{\setminus i}$ and Var_i is the variance with respect to Y_i. This measures the *additive effect* that is due to Y_i or, in plain words, how the average model output changes as we vary y_i.

2.2. Total Indices

The *total index* of the variable Y_i swaps the roles of the expectation and the variance in Equation (1):

$$S_i^T := \frac{\mathrm{E}_{\setminus i}\left[\mathrm{Var}_i[f]\right]}{\mathrm{Var}[f]}. \tag{2}$$

The S_i^T measures the overall effect due to Y_i or, in plain words, the average variability that results from changing y_i while leaving all other parameters fixed. This effect includes the additive effect, which means that $S_i \leq S_i^T$ under the assumption that the input variables are independent of each other. We have $S_i = S_i^T$ if and only if Y_i can be separated from f as an additive component, that is, if the original function can be decomposed

as $f(\boldsymbol{y}) = g(y_i) + h(\boldsymbol{y}_{\setminus i})$. Among other uses, total indices are a popular tool for model simplification: a small S_i^T guarantees that variable Y_i can be safely *frozen* to any value of its domain and removed from the model.

2.3. Applications in Reliability Engineering

As they are able to capture global and non-linear effects, Sobol indices are highly expressive and have become a gold-standard in sensitivity analysis of multivariate functions in engineering and simulation (Saltelli et al., 2008; Sudret, 2008; Marrel et al., 2009; Iooss and Lemaître, 2015). Applications include model screening (i.e. removal of non-important variables), interpretation, factor prioritization, measuring uncertainty propagation, and identification of inter-variable interaction (similarly to multi-way ANOVA analysis), among others. In addition, a generalization of Sobol's method has been proposed that is particularly suitable to detect extreme values in the function of interest (Owen et al., 2014). Other generalizations that yield relevant metrics of interest include the Shapley values (Owen, 2014), the mean dimension (Liu and Owen, 2006), etc.

2.4. Algorithmic Challenges

Since both the expectation and the variance in Equations (1) and (2) are multidimensional integrals (or multi-index summations, in the discrete case), they are notoriously difficult to compute. Estimation for black-box functions is often undertaken via Monte Carlo sampling routines, which, due to the Central Limit Theorem, have a slow convergence rate of $O(\sqrt{n})$ in general where n is the number of samples taken. Even when a Quasi-Monte Carlo sampling plan is used, for example via Sobol or Saltelli sequences (Saltelli et al., 2008), large sampling budgets (say, with n between 10^4 and 10^8) are often needed in practice for each and every Sobol index. Alternatively, more efficient algorithms exist when f is represented as a *surrogate model*, which is the traditional sensitivity analysis approach to approximate expensive function evaluations. This includes *polynomial chaos expansions* (Sudret, 2008) and *Gaussian processes* (Marrel et al., 2009), among others; unfortunately, they are often poorly suited to approximate BNs, which are the target of this paper. For example, although BNs can naturally model discrete variables, polynomial chaos expansion and Gaussian processes assume continuous variables. Also, BNs model probability mass (or density) functions, which are non-negative functions and must add to 1 - but neither of the previously mentioned methods can take into account these constraints.

3. Bayesian Networks and Graphical Models

BNs are one of the most common statistical methods to investigate the dependence structure for a random vector of interest and to efficiently answer inferential queries. Next we review the theory of BNs and discuss their connection to more general probabilistic graphical models.

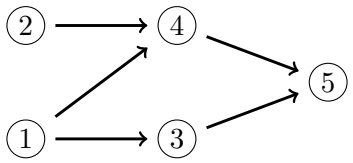


Figure 1: A simple DAG of a BN with vertex set $\{1, 2, 3, 4, 5\}$ and edge set $\{(1, 3), (1, 4), (2, 4), (3, 5), (4, 5)\}$.

3.1. Bayesian Networks

Consider a random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ of interest, where for the purposes of this paper we assume Y_i to be taking values in a discrete space $\mathbb{Y}_i$, $i \in [n]$. A BN gives a graphical representation of the relationship between the variables Y_i by means of a directed acyclic graph (DAG) (see e.g. Trudeau, 2013, for a review) as well as a factorization of the overall probability distribution $\Pr(\boldsymbol{Y} = \boldsymbol{y})$ in terms of simpler (conditional) distributions $\Pr(Y_i = y_i | \boldsymbol{Y}_{\mathbb{P}_i} = \boldsymbol{y}_{\mathbb{P}_i})$, where $\boldsymbol{Y}_{\mathbb{P}_i}$ includes the *parents* of the variable Y_i in the DAG of the BN. More formally, the overall factorization of the probability distribution induced by the BN can be written as

$$\Pr(\boldsymbol{Y} = \boldsymbol{y}) = \prod_{i=1}^{n} \Pr(Y_i = y_i | \boldsymbol{y}_{\mathbb{P}_i} = \boldsymbol{y}_{\mathbb{P}_i}). \tag{3}$$

Figure 1 gives an illustration of a DAG over five discrete variables $(Y_1, \ldots, Y_5)$. The nodes represent the index of the associated variable and the arrows indicate dependence, meaning that the conditional distribution of a node depends on its parents. The associated factorization of this BN is written as

$$\begin{aligned}\Pr(\boldsymbol{Y} = \boldsymbol{y}) = \Pr(Y_5 = y_5 | Y_4 = y_4, Y_3 = y_3) \Pr(Y_4 = y_4 | Y_2 = y_2, Y_1 = y_1) \\ \Pr(Y_3 = y_3 | Y_1 = y_1) \Pr(Y_2 = y_2) \Pr(Y_1 = y_1)\end{aligned}$$

The DAG associated to a BN provides an intuitive overview of the relationships existing between variables of interest. However, it does also provide a framework to assess if any generic conditional independence holds for a specific subset of the variables via the so-called *d-separation* criterion (see e.g. Darwiche, 2009). Furthermore, the DAG provides a framework for efficient propagation of probabilities and evidence for the computation of any (conditional) probability involving a specific subset of the variables.

Both of the above tasks usually require a manipulation of the original DAG, called *moralization*. In a nutshell, moralization adds an undirected edge between any two parents of a vertex which are not joined by an edge (hence the word moralize). Figure 2a reports the moralized version of the DAG in Figure 1. The *skeleton* of a DAG consists of the moralized DAG where all directed edges are replaced by undirected ones. The skeleton of the DAG in Figure 1 is given in Figure 2b. Such graphs enable for fast probability computations by taking advantage of an alternative factorization of $\Pr(\boldsymbol{Y} = \boldsymbol{y})$ to the one in Equation (3) in terms of cliques and separators (see e.g. Darwiche, 2009, for details).

(a) Moralized DAG (b) Skeleton

Figure 2: Illustration of the moralization process for the BN in Figure 1.

3.2. Markov Random Fields

Starting from a BN, the moralization process returns an undirected graph which provides an efficient framework to carry out a variety of tasks including d-separation, probability propagation, etc. However, in many applications it is customary to define a probabilistic graphical model based on undirected graphs directly (see e.g. Guo et al., 2015; Ma et al., 2019; L. Salemi et al., 2019). Such models are usually called *Markov Random Fields* (MRFs). Here we follow Robeva and Seigal (2018) and define them via hypergraphs (see e.g. Bretto, 2013).

Definition 1. *An MRF $\mathcal{B}$ with respect to an hypergraph with vertex set $[n]$ and hyperedges $\mathbb{C}_\mathcal{B}$ is defined by*

$$\begin{aligned}\phi_\mathcal{B} : \mathbb{Y} &\to \mathbb{R}_+ \\ \phi_\mathcal{B}(\boldsymbol{y}) &\to \frac{1}{z} \prod_{C \in \mathcal{P}([n])} \phi_\mathcal{B}(\boldsymbol{y}_C),\end{aligned}$$

where $\phi_\mathcal{B}(\boldsymbol{y}_C) = 1$ if $C \notin \mathbb{C}_\mathcal{B}$ and otherwise is a generic function from $\mathbb{Y}_\mathbb{C}$ to $\mathbb{R}_+$. The functions $\phi_\mathcal{B}(\boldsymbol{y}_\mathbb{C})$ are called potentials *and z is a normalizing constant to impose that $\sum_{\boldsymbol{y}\in\mathbb{Y}} \phi_\mathcal{B}(\boldsymbol{y}) = 1$.*

The previous definition is very general but is needed when working with TNs. More often, MRFs are defined via a simple undirected graph, and the function ϕ factorizes over the cliques of the graph (for this reason we denoted the hyperedges as $\mathbb{C}_\mathcal{B}$). Conversely, Definition 1 allows the factorization to work over all subsets of the vertex set. Notice that $\phi_\mathcal{B}(\boldsymbol{y})$ is actually a probability distribution since it is non-negative and sums to one.

As an illustration, consider the undirected graph in Figure 2b, whose cliques are $\{1, 2, 4\}$, $\{1, 2, 4\}$, $\{3, 4, 5\}$. An MRF over this undirected graph can be defined by constructing an hypergraph with the same vertex set and as hyperedges its cliques. Its probability distribution can be written as

$$\Pr(\boldsymbol{y}) = \phi_\mathcal{B}(\boldsymbol{y}) = \frac{1}{z}\phi_\mathcal{B}(y_1, y_2, y_4)\phi_\mathcal{B}(y_1, y_3, y_4)\phi_\mathcal{B}(y_3, y_4, y_5).$$

Notice that very BN can be cast as an MRF via the moralization process by setting as hyperedges of the MRF the cliques of the skeleton of the BN. Our new global sensitivity

analysis methods take advantage of the transformation of a BN to its skeleton (or to its equivalent MRF) to perform efficient probability computations. In particular, an operation required for probability propagation and that our methods will heavily use is that of *marginalization*. Given an MRF $\mathcal{B}$ over the vector $\boldsymbol{Y}$ and a set $\mathbb{A} \subset [n]$ we want to compute the probability $\Pr(\boldsymbol{y}_{\mathbb{A}})$ which can be derived as

$$\Pr(\boldsymbol{y}_{\mathbb{A}}) = \sum_{\boldsymbol{y}_{\setminus\mathbb{A}} \in \mathbb{Y}_{\setminus\mathbb{A}}} \Pr(\boldsymbol{y}_{\mathbb{A}}, \boldsymbol{y}_{\setminus\mathbb{A}}) = \frac{1}{z} \sum_{\boldsymbol{y}_{\setminus\mathbb{A}} \in \mathbb{Y}_{\setminus\mathbb{A}}} \prod_{C \in \mathbb{C}_{\mathcal{B}}} \phi_{\mathcal{B}}(\boldsymbol{y}_C).$$

Since the normalizing constant z is often hard to compute, we will simply work with potentials and non-normalized probabilities and call:

$$\mathcal{B}_{\setminus\mathbb{A}}(\boldsymbol{y}_{\mathbb{A}}) = \sum_{\boldsymbol{y}_{\setminus\mathbb{A}} \in \mathbb{Y}_{\setminus\mathbb{A}}} \prod_{C \in \mathbb{C}_{\mathcal{B}}} \phi_{\mathcal{B}}(\boldsymbol{y}_C),$$

so that $\Pr(\boldsymbol{y}_{\mathbb{A}}) = \frac{1}{z}\mathcal{B}_{\setminus\mathbb{A}}(\boldsymbol{y}_{\mathbb{A}})$.

3.3. Tensor Networks

Since MRFs encode probabilities in a non-normalized way, they are still subject to the non-negativity constrain. *Tensor networks* drop this condition and thus become a graphical representation for general functions. Therefore, a TN $\mathcal{M}$ is defined as an MRF $\mathcal{B}$, but whose potentials $\phi_{\mathcal{M}}$ are generic functions to $\mathbb{R}$. For instance, the problem of tensor decomposition seeks to find a low-parametric network that encodes, either exactly or approximately, a given real or complex-valued input tensor.

In this paper we do not aim to learn a TN out of data. Instead, we turn the input BN $\mathcal{N}$, which may have been learned from data, into its equivalent MRF, and then derive a TN $\mathcal{M}$ from $\mathcal{B}$ by simply dropping the non-negativity constraints from $\mathcal{B}$. Such a TN is said to be *derived* from $\mathcal{B}$. Given this TN $\mathcal{M}$ we then manipulate it in order to extract relevant sensitivity indices for the original BN $\mathcal{N}$: see Section 4 below.

4. Proposed Method for Global Sensitivity Analysis in BNs

4.1. Problem Setup

As customary in sensitivity analysis for BNs (see e.g. Gómez-Villegas et al., 2013), we assume a partition of the vertices into an output node $\mathbb{O}$, evidential nodes $\mathbb{E}$ and chance nodes $\mathbb{U}$, i.e. $[n] = \mathbb{O} \cup \mathbb{E} \cup \mathbb{U}$. In more details:

- $Y_{\mathbb{O}}$ is an *output variable* whose expected value is a quantity of interest.
- $\boldsymbol{Y}_{\mathbb{E}}$ is a vector of *evidential variables* whose influence on $Y_{\mathbb{O}}$ we wish to quantify.
- $\boldsymbol{Y}_{\mathbb{U}}$ is a vector of *chance variables* which are unobserved and cannot be fixed.

Formally, given such a network, we wish to study the following function of interest:

$$\begin{aligned} f : \mathbb{Y}_{\mathbb{E}} &\to \mathbb{R} \\ \boldsymbol{y}_{\mathbb{E}} &\mapsto \mathrm{E}_{\mathbb{U}}[Y_{\mathbb{O}} | \boldsymbol{Y}_{\mathbb{E}} = \boldsymbol{y}_{\mathbb{E}}], \end{aligned} \tag{4}$$

where $\mathrm{E}_{\mathbb{U}}$ refers to the expectation with respect to $\boldsymbol{Y}_{\mathbb{U}}$. Without loss of generality we assume there is a single output variable $Y_{\mathbb{O}}$ (see the remark below).

Specifically, our goal is to assess the *global effect* of observed evidence $\boldsymbol{Y}_{\mathbb{E}} = \boldsymbol{y}_{\mathbb{E}}$ on the quantity of interest $f(\boldsymbol{y}_{\mathbb{E}})$ from Equation (4). In Sobol's variance-based sense, the effect of an evidential variable Y_i is captured by two indices S_i and S_i^T for every $i \in \mathbb{E}$. Ours is the first method to address the case in which f is defined by a probabilistic graphical model as a function of evidence in several of its nodes. Note that explicitly evaluating $f(\boldsymbol{y}_{\mathbb{E}})$ requires inference in a BN. This operation, which bears significant computational cost for one sample $\boldsymbol{y}_{\mathbb{E}}$, becomes quickly impractical for thousands or millions of samples, as is typically required by Monte Carlo based methods. Building a surrogate model is rarely viable and accurate enough for general BNs, and the case of correlated input variables is even more challenging. Our algorithms avoid the explicit evaluation of f and, instead, exploit the network's structure to efficiently obtain all variances required in Equations (1) and (2).

One must also account for the way evidential variables are distributed: they follow a joint probability mass function $\Pr(\boldsymbol{y}_{\mathbb{E}}) \propto \mathcal{B}_{\setminus\mathbb{E}}(\boldsymbol{y}_{\mathbb{E}})$ for $\boldsymbol{y}_{\mathbb{E}} \in \mathbb{Y}_{\mathbb{E}}$. If all vertices $\mathbb{E}$ are root nodes (i.e. they have no parents), Pr is separable, which is an important particular case in the sensitivity analysis literature; in fact, many applications of the method of Sobol assume uncorrelated inputs. For example, when input values can be set at will (as in an experimental design), it is customary to assume independent, uniformly distributed inputs in a rectangle $\Omega \subset \mathbb{R}^{|\mathbb{E}|}$. The method we propose supports correlated inputs as well by means of computing $\mathcal{B}_{\setminus\mathbb{E}}$, which we do via marginalization. While marginalizing the offspring of $\mathbb{E}$ is straightforward (no operations are required), marginalizing its ancestors is more computationally intensive. Therefore, our method is most useful when the evidential nodes are located in a relatively upstream position in the graph.

Remark. Note that our formulation is flexible enough to allow for arbitrary output functions to be considered, including those that represent the *utility* of a set of nodes. For example, suppose we initially have a network with nodes $\mathbb{E} \cup \mathbb{U}$ and a utility function $g : \mathbb{Y}_{\mathbb{D}} \to \mathbb{Y}_{\mathbb{O}}$ where $\mathbb{D} \subseteq \mathbb{E} \cup \mathbb{U}$ is a subset of the network's variables, and assume we want to study g's output as a function of variables $\mathbb{E}$. To reduce this task to Equation (4), we encode g into a new node $\mathbb{O}$ that we add to the graph as in Luque and Díez (2010). The domain of $Y_{\mathbb{O}}$ is $\mathbb{Y}_{\mathbb{O}}$, its parents are $\boldsymbol{Y}_{\mathbb{D}}$, and its conditional probability table Pr is built so as to reflect g:

$$\Pr(y_{\mathbb{O}} | \boldsymbol{y}_{\mathbb{D}}) = \begin{cases} 1 \text{ if } y_{\mathbb{O}} = g(\boldsymbol{y}_{\mathbb{D}}) \\ 0 \text{ otherwise.} \end{cases}$$

This function encoding strategy is demonstrated in one of our experiments in Section 5.

4.2. Network Arithmetics

The proposed algorithms rely on two new operations between TNs corresponding to the squaring of a TN and to the quotient between two TNs over the same vertices. Such operations are implemented by manipulating the original TNs into new ones, a procedure that we report in full detail in Appendix A. The proofs of the results of this section are collated in Appendix B.

Definition 2. *Given a TN $\mathcal{M}$ with vertex set $[n]$ and $\mathbb{A} \subseteq [n]$, the* square *of $\mathcal{M}$ with respect to $\mathbb{A}$ is a TN $\mathcal{M}^2$ such that $(\mathcal{M}^2)_{\setminus\mathbb{A}}(\boldsymbol{y}_\mathbb{A}) = (\mathcal{M}_{\setminus\mathbb{A}}(\boldsymbol{y}_\mathbb{A}))^2$ for any $\boldsymbol{y}_\mathbb{A} \in \mathbb{Y}_\mathbb{A}$. Since the squaring commutes with the marginalization, we simply write $\mathcal{M}^2_{\setminus\mathbb{A}}$.*

Definition 3. *Let $\mathcal{M}$ and $\mathcal{M}'$ be two TNs for $\boldsymbol{Y}$ whose hypergraphs have the same vertex set $[n]$. The* quotient *of $\mathcal{M}$ over $\mathcal{M}'$ denoted as $\mathcal{M}/\mathcal{M}'$ is such that $(\mathcal{M}/\mathcal{M}')(\boldsymbol{y}) = \mathcal{M}(\boldsymbol{y})/\mathcal{M}'(\boldsymbol{y})$ for any $\boldsymbol{y} \in \mathbb{Y}$.*

The following proposition formalizes how the operation of quotient commutes with those of marginalization and squaring. This commuting will be heavily used in our algorithms for the computation of the Sobol indices.

Proposition 1. *Let $\mathbb{A} \subset [n]$. Then, $(\mathcal{M}/\mathcal{M}'_\mathbb{A})_\mathbb{A} = \mathcal{M}_\mathbb{A}/\mathcal{M}'_\mathbb{A}$. In addition, $(\mathcal{M}/\mathcal{M}'_\mathbb{A})^2_\mathbb{A} = (\mathcal{M}_\mathbb{A})^2/(\mathcal{M}'_\mathbb{A})^2$, where the square is w.r.t. $\setminus\mathbb{A}$.*

4.3. Representing the Target Function

The quotient operation is also useful to build a TN representation of the function of interest f in a sensitivity analysis, as shown by the following proposition. Recall that our starting point is BN whose vertex set is decomposed into evidential, chance and output nodes, i.e. $\mathbb{E} \cup \mathbb{U} \cup \mathbb{O} = [n]$. The MRF $\mathcal{B}$ derived from such a BN inherits the same vertex decomposition as well as the TN $\mathcal{M}$ which loses the non-negativity condition. If furthermore we set $\phi(y_\mathbb{O}) = y_\mathbb{O}$ we call $\mathcal{M}$ the *sensitivity TN* derived from $\mathcal{B}$. Loosely speaking, the sensitivity TN is simply the TN derived from a BN $\mathcal{B}$ where the potential associated to the output node is fixed to one possible value of interest $y_\mathbb{O}$.

Proposition 2. *Let $\mathcal{M}$ be the sensitivity TN derived from a MRF $\mathcal{B}$. The function of interest f (Equation 4) equals $(\mathcal{M}/\mathcal{B}_{\mathbb{U}\cup\mathbb{O}})_{\mathbb{U}\cup\mathbb{O}} = \mathcal{M}_{\mathbb{U}\cup\mathbb{O}}/\mathcal{B}_{\mathbb{U}\cup\mathbb{O}}$.*

The expectation of a TN further entertains a representation in terms of quotients.

Proposition 3. *Let $\mathcal{M}$ be the sensitivity TN derived from a MRF $\mathcal{B}$. For any partition $\{\mathbb{A}, \mathbb{B}, \mathbb{D}\}$ of $[n]$ we have $\mathrm{E}_\mathbb{A}[\mathcal{M}_\mathbb{B}] = (\mathcal{M}_\mathbb{B} \cdot \mathcal{B}_\mathbb{B}/\mathcal{B}_{\mathbb{A}\cup\mathbb{B}})_\mathbb{A}$.*

The above result is needed to show that the expectation of the function of interest in sensitivity analysis is simply the sensitivity TN where all variables are marginalized.

Proposition 4. $\mathrm{E}[f] = \mathcal{M}_{[n]}$.

Having introduced these fundamental results, we are now ready to compute the Sobol indices as defined in Section 2.

4.4. Computing the Global Variance of f

Both types of Sobol indices (Equations 1 and 2) require computing $\mathrm{Var}[f]$ in the denominator. We do so via the formula $\mathrm{Var}[f] = \mathrm{E}[f^2] - \mathrm{E}[f]^2$ and deriving each of these terms as follows:

- First term: $\mathrm{E}[f^2] = (\mathcal{M}^2/\mathcal{B}_{\setminus\mathbb{E}})_{\mathbb{E}\cup\setminus\mathbb{E}}$, where the square of $\mathcal{M}$ is taken w.r.t. $\mathbb{E}$. See Appendix B.5 for the proof.
- Second term: $\mathrm{E}[f]^2 = (\mathcal{M}_{[n]})^2$ (using Prop. 4).

4.5. Computing the Variance Components

We obtain the index S_i via Equation (1), where the numerator is found via the identity

$$\mathrm{Var}_i[\mathrm{E}_{\setminus i}[f]] = \mathrm{E}_i[\mathrm{E}^2_{\setminus i}[f]] - \mathrm{E}_i[\mathrm{E}_{\setminus i}[f]]^2 \tag{5}$$

and by observing that:

- The first term is equal to $\mathrm{E}_i[\mathrm{E}^2_{\setminus i}[f]] = (\mathcal{M}^2_{\setminus i}/\mathcal{B}_{\setminus i})_i$. See Appendix B.6 for the proof.
- For the second term, $\mathrm{E}_i[\mathrm{E}_{\setminus i}[f]]^2 = \mathrm{E}[f]^2 = (\mathcal{M}_{[n]})^2$ (using Prop. 4).

The denominator is simply the model's global variance $\mathrm{Var}[f]$ (Section 4.4).

4.6. Computing the Total Indices

To obtain the index S_i^T, we use the following alternative definition:

$$S_i^T = 1 - \frac{\mathrm{Var}_{\setminus i}\left[\mathrm{E}_i[f]\right]}{\mathrm{Var}[f]}, \tag{6}$$

which is equivalent (Saltelli et al., 2008) to the one given in Equation (2). To obtain the numerator, we use the identity in Equation (5) and observe that:

- For the first term, analogously to Section 4.5, substituting i for $\setminus i$ and vice versa we arrive at $\mathrm{E}_{\setminus i}[\mathrm{E}^2_i[f]] = (\mathcal{M}^2_i/\mathcal{B}_{\setminus\mathbb{E}\cup\{i\}})_{\setminus i}$, where the square is taken w.r.t. $\mathbb{E}\setminus i$.
- For the second term, again $\mathrm{E}_{\setminus i}[\mathrm{E}_i[f]]^2 = \mathrm{E}[f]^2 = (\mathcal{M}_{[n]})^2$ (using Prop. 4).

4.7. Summarized Algorithm

We put together all steps in Algorithm 1 so as to minimize the cost of intermediate calculations. See Appendix A for the full details on how we construct the square and quotient networks (Propositions 5 and 7).

Although in this paper we only demonstrate first-order Sobol indices, higher-order versions in the literature are defined using the same formulas (Equations 1 and 2) by simply replacing variable singletons by the appropriate tuple of variables. Therefore, they are readily supported by our method.

Algorithm 1: Compute first-order Sobol indices for a given BN

Input: Bayesian network $\mathcal{N}$ with variables $\mathbb{O} \cup \mathbb{U} \cup \mathbb{E}$, variable of interest $i \in \mathbb{E}$, domain $\mathbb{Y}_{\mathbb{O}}$ of the output variable $Y_{\mathbb{O}}$

Result: Variance component S_i and total index S_i^T for variable Y_i

▶ Derive the equivalent MRF $\mathcal{B}$ to $\mathcal{N}$

▶ Construct the sensitivity TN $\mathcal{M}$ derived from $\mathcal{B}$

▶ Find $\mathcal{J} := \mathcal{B}_{\setminus \mathbb{E}}$ via marginalization

▶ Compute $\mathcal{M}_i, \mathcal{J}_i, \mathcal{M}_{\setminus i}, \mathcal{J}_{\setminus i}, \mathcal{M}_{[n]}$ via marginalization

▶ Compute $\mathcal{M}^2, \mathcal{M}_i^2$, and $\mathcal{M}_{\setminus i}^2$ via network squaring (Prop. 5)

▶ Compute $\mathcal{M}^2/\mathcal{J}, \mathcal{M}_i^2/\mathcal{J}_i$, and $\mathcal{M}_{\setminus i}^2/\mathcal{J}_{\setminus i}$ via network quotient (Prop. 7)

▶ $V := (\mathcal{M}^2/\mathcal{J})_{\mathbb{E} \cup \setminus \mathbb{E}} - (\mathcal{M}_{[n]})^2$

▶ $S_i := ((\mathcal{M}_{\setminus i}^2/\mathcal{J}_{\setminus i})_i - (\mathcal{M}_{[n]})^2)/V$

▶ $S_i^T := 1 - \left((\mathcal{M}_i^2/\mathcal{J}_i)_{\setminus i} - (\mathcal{M}_{[n]})^2\right)/V$

return S_i, S_i^T

5. Illustrations

In order to assess the validity of our algorithms, we now consider two networks based on simulated data and one real-world network. In all cases, we use exact marginalization via the *minimal weight* heuristic for variable ordering (Kjærulff, 1990): at each step, we marginalize the node such that the product of its neighbors' cardinalities is minimal.

All experiments were run on a 4-core i5-6600 3.3GHz Intel workstation with 16GB RAM.

5.1. Concrete Resistance

As a first example, we consider a 24-nodes BN due to Castillo and Kjærulff (2003) designed to assess the damage of reinforced concrete structures of buildings. There is one output variable (the damage of a reinforced concrete beam), 16 observable variables influencing the damage of reinforced concrete structures and seven intermediate unobservable variables that define some partial states of the structure. The list of variables and their definition is reported in Table 1 and their cause-effect relationships are summarized by the BN in Figure 3. The original variables in Castillo and Kjærulff (2003) are continuous and their model is defined as a Gaussian BN (e.g. Darwiche, 2009). From their definition we derived a discrete BN as follows: (i) we simulate 1'000'000 observations from their Gaussian BN; (ii) each variable is independently discretized into three categories (low, medium, high) using the equal frequency method (Nojavan et al., 2017); (iii) retaining the same DAG as in Castillo and Kjærulff (2003), the conditional probabilities for the discrete data are learned. All these steps are carried out with the `bnlearn` package (Scutari, 2010).

The 16 evidential variables are the roots of the DAG (that is, without parents), for a total of $3^{16} \approx 4.31 \cdot 10^7$ grid points in the domain of the function of interest f. The resulting

Variable	Definition
O	Damage assessment
U_7	Cracking state
U_6	Cracking state in shear domain
U_5	Steel corrosion
U_4	Cracking state in flexure domain
U_3	Shrinkage cracking
U_2	Worst cracking in flexure domain
U_1	Corrosion state
E_{16}	Weakness of the beam
E_{15}	Deflection of the beam
E_{14}	Position of the worst shear crack
E_{13}	Breadth of the worst shear crack
E_{12}	Position of the worst flexure crack
E_{11}	Breadth of the worst flexure crack
E_{10}	Length of the worst flexure cracks
E_9	Cover
E_8	Structure age
E_7	Humidity
E_6	pH value in the air
E_5	Content of chlorine in the air
E_4	Number of shear cracks
E_3	Number of flexure cracks
E_2	Shrinkage
E_1	Corrosion

Table 1: Variable labels and names of the *Concrete* network.

Var. i	S_i	Time (S_i)	S_i^T	Time (S_i^T)
E1	0.05464	0.06325	0.06335	0.09594
E2	0.03081	0.05967	0.03569	0.10103
E3	0.06339	0.06109	0.07750	0.09831
E4	0.04156	0.05868	0.05470	0.09775
E5	0.04837	0.06028	0.06217	0.09880
E6	0.06281	0.06016	0.07860	0.09927
E7	0.01783	0.05837	0.02503	0.09913
E8	0.02344	0.05937	0.03393	0.09868
E9	0.02357	0.05922	0.03412	0.09612
E10	0.05574	0.06039	0.07461	0.09657
E11	0.05560	0.05826	0.07452	0.09817
E12	0.12943	0.05956	0.15069	0.09695
E13	0.14802	0.05955	0.17370	0.09784
E14	0.08480	0.05865	0.10512	0.10081
E15	0.07363	0.05868	0.07535	0.10183
E16	0.01338	0.05963	0.01388	0.09872

Table 2: *Concrete* network (Castillo and Kjærulff, 2003): calculated Sobol sensitivity indices along with the required computation times.

concrete resistances are given by output variable O, whose values we map to $\mathbb{R}$ as follows: low $\equiv 0$, medium $\equiv 1$, and high $\equiv 2$. Table 2 and Figure 3 summarize the global sensitivity indices. Overall, computing all 16 variance components and total indices took 1.12s and 1.79s, respectively. Each index requires four marginalizations in our implementation, for a total of $16 \cdot 2 \cdot 4 = 128$ such operations that took 2.24s globally.

The results suggest that the most critical variables for the output are position of the worst flexure crack (E_{12}), breadth of the worst shear crack (E_{13}) and position of the worst shear crack (E_{14}). Conversely, humidity (E_7) and weakness of the beam (E_6) have very little effect on the output: therefore if one needs to assess the damage of concrete structures, these two inputs could almost be disregarded. Notice that the significant differences between variance components and total indices indicate that much of those effects are due to variable interactions, not to single variables alone, therefore highlighting the need of global sensitivity analyses to uncover the actual relationships between input and output variables, which a local analysis would have not been able to retrieve.

5.1.1. Timing Comparison

Based on the average marginalization time of the *Concrete* model, a brute-force computation evaluating all entries of f would have taken more than 10^5 seconds. We conducted a comparison vs. Monte Carlo-based estimation of the total indices using the *SALib* sensitivity analysis library (Herman and Usher, 2017). Results are reported in Fig. 4, which shows that MC indices converge to ours well after 10^4 samples, which requires several orders of magnitude more time than the proposed method.

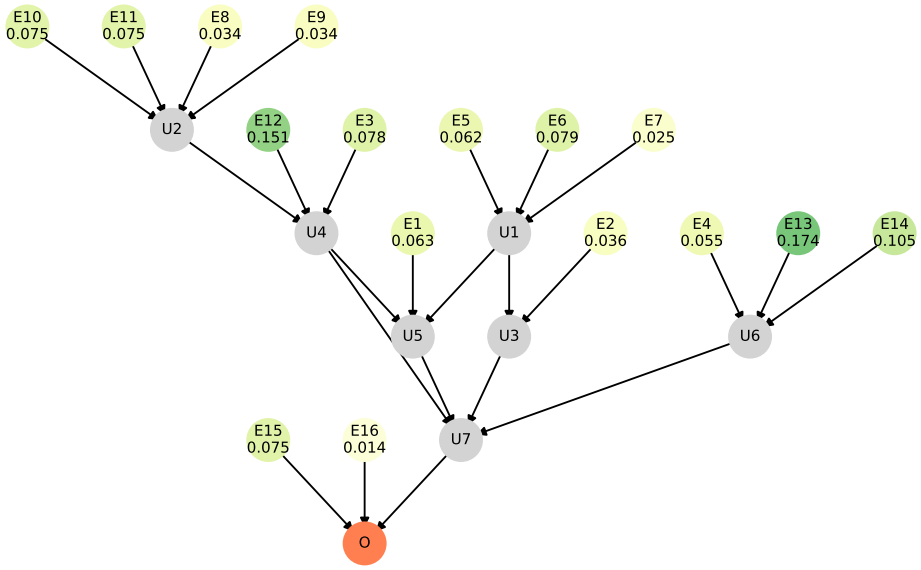


Figure 3: *Concrete* network (Castillo and Kjærulff, 2003): sensitivity analysis on output variable O. Chance nodes are shown in gray, while evidential nodes are color-coded according to their respective total indices (Sec. 2.2).

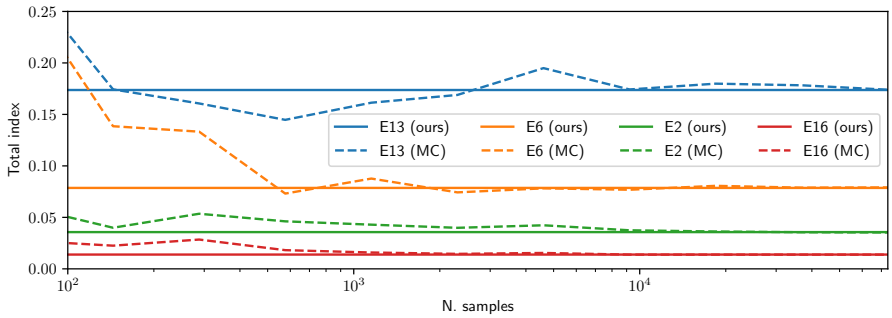


Figure 4: Four total indices of the *Concrete* network as computed by our method vs. Monte Carlo estimation over different numbers of samples. Computing all 16 total indices took 1.24s with our method and 806.17s at 73'278 samples for the MC method.

5.2. Synthetic Network with Dependent Inputs

As a second simulated example, we created a 40-nodes Gaussian Bayesian network with three output nodes, 15 evidential nodes, and 22 chance nodes. Both the structure of the DAG (reported in Figure 5) and the conditional distributions of the nodes where chosen a priori so as to best illustrate the methodology. In this case, however, the evidential variables exhibit some dependence. A discrete BN is obtained using the same routine as in the previous example. The output is a new additional node $O(O_1, O_2, O_3) := O_1 + O_2 + O_3$,

where low $\equiv 0$, medium $\equiv 1$ and high $\equiv 2$. Since there are 15 evidential variables, the domain of the function of interest f contains $3^{15} \approx 1.43 \cdot 10^7$ grid points. Table 3 (right) and Figure 5 report the global sensitivity indices. Note that $S_i > S_i^T$ for several variables; this is a well-known consequence of the non-independence of the function's inputs. Overall, computing all 15 variance components and total indices took 5.88s and 11.48s, respectively. Although, the computation time is slightly larger than in the previous example, it is still feasible to compute the global sensitivity indices in the much more challenging case of dependent inputs.

To illustrate the consequences of overlooking dependence in the inputs, we created the same discrete BN with 15 inputs, but where the dependence between the evidential variables is deleted. The global sensitivity indices for the independent network are reported on the left side of Table 3. We can notice two things. First, $S_i^T \geq S_i$ for all variables, due to missed dependence between inputs. Second, evidential variables that are important for the output through the dependence with other inputs (for instance E_2 or E_5) have now a much smaller effect on the output and their relevance would be overlooked by considering the inputs independent.

	Independent inputs				Dependent inputs			
Var. i	S_i	Time (S_i)	S_i^T	Time (S_i^T)	S_i	Time (S_i)	S_i^T	Time (S_i^T)
E1	0.00812	0.38456	0.00832	0.66675	0.05757	0.36064	0.00437	0.63704
E10	0.00617	0.35793	0.00658	0.67984	0.00277	0.35715	0.00311	0.65014
E11	0.03099	0.35240	0.03118	0.65987	0.01198	0.35403	0.01237	0.64012
E12	0.06793	0.34724	0.06883	0.74234	0.04408	0.35532	0.02202	0.71773
E13	0.00471	0.35308	0.00497	0.71207	0.02735	0.35807	0.00482	0.72696
E14	0.00336	0.35799	0.00337	0.69925	0.00121	0.36493	0.00124	0.65699
E15	0.03169	0.36250	0.03193	0.60682	0.01366	0.35283	0.01395	0.61534
E2	0.04169	0.35959	0.04200	0.64520	0.15364	0.35414	0.01426	0.64304
E3	0.02196	0.35772	0.02223	0.64319	0.10733	0.35941	0.01452	0.63806
E4	0.05410	0.35593	0.05445	0.64833	0.13931	0.35352	0.03504	0.65013
E5	0.08062	0.35327	0.08140	0.58844	0.27517	0.35414	0.03071	0.60475
E6	0.34252	0.35654	0.34322	0.59323	0.37965	0.35600	0.17480	0.61408
E7	0.11156	0.35384	0.11220	0.64536	0.16543	0.41148	0.05473	0.64713
E8	0.08040	0.35663	0.08076	0.59313	0.13329	0.35635	0.02463	0.61735
E9	0.11108	0.35002	0.11224	0.64339	0.17694	0.35846	0.06924	0.64184

Table 3: *Synthetic* network (two variants are considered here): calculated Sobol sensitivity indices along with the required computation times.

5.3. Risk Assessment in Project Complexity

Our last example is a BN due to Qazi et al. (2016) representing critical risks specific to a construction project and adapted from an existing model proposed by Eybpoosh et al. (2011). Qazi et al. (2016) considers eight project complexity elements which can be chosen at the starting of the project as well as four project objectives: timeliness; cost; quality; and

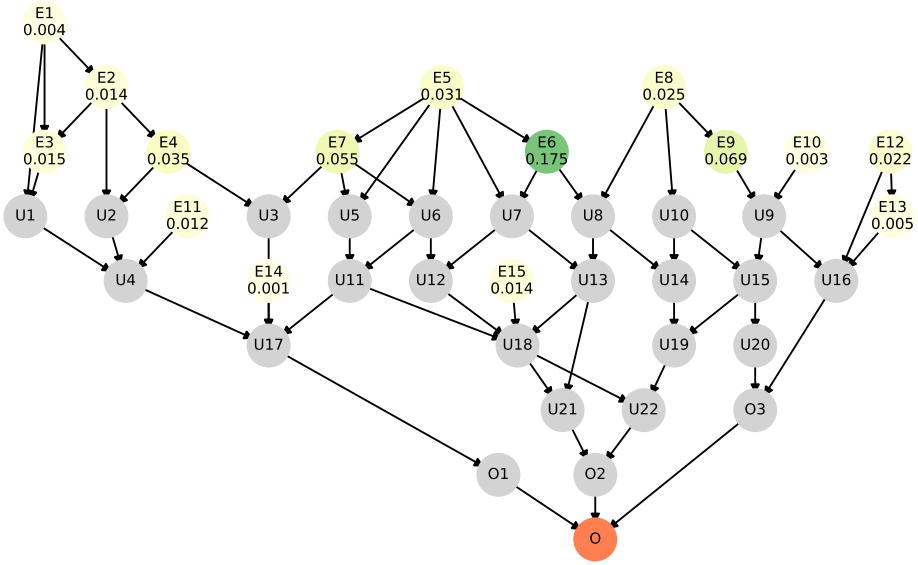


Figure 5: *Synthetic* network with dependent inputs: sensitivity analysis on the output quantity of interest $O(\mathrm{O}_1, \mathrm{O}_2, \mathrm{O}_3)$. Chance nodes are shown in gray, while evidential nodes are color-coded according to their respective total Sobol index (Sec. 2.2).

market share. All these objectives are henceforth presented as negative counterparts in order to align these to the notion of risks and are assumed to be equally important. There are 14 additional variables that are not observed at the start of the project (chance variables). All variables are defined as binary; see Table 4 for a full list with their descriptions. Although the eight project complexity elements are considered as decision variables in Qazi et al. (2016), giving an influence diagram representation of the problem (e.g. Darwiche, 2009), here they are considered as evidential random variables and are assigned a priori an equal probability to their outcomes.

Table 5 and Figure 6 report a summary of the results. Overall, computing all 8 variance components and total indices took 1.06s and 2.00s, respectively, thus highlighting the computational efficiency of our algorithms. Clearly, the lack of experience with the involved team (E_1) has the biggest impact on the possible difficulties involved with the project, with Sobol indices more than twice larger than for all other variables. Other complexity elements that are relevant for the output are (in order of importance) strict quality requirements (E_4), lack of experience with technology (E_3) and political instability (E_7). Conversely, the presence of multiple stakeholders and perspectives (E_6) has almost no effect on the success of the construction project, with Sobol indices which are much smaller than for all other variables.

Variable	Definition
E_1	Lack of experience with the involved team
E_2	Use of innovative technology
E_3	Lack of experience with technology
E_4	Strict quality requirements
E_5	Multiple contracts
E_6	Multiple stakeholder and variety of perspectives
E_7	Political instability
E_8	Susceptibility to natural disasters
U_1	Contractors' lack of experience
U_2	Suppliers default
U_3	Delays in design and regulatory approval
U_4	Contract related problems
U_5	Economic issues in country
U_6	Major design changes
U_7	Delays in obtaining raw material
U_8	Non-availability of local resources
U_9	Unexpected events
U_{10}	Increase in raw material prices
U_{11}	Change in project specifications
U_{12}	Conflicts with project stakeholders
U_{13}	Decrease in productivity
U_{14}	Delays/interruptions
O_1	Decrease in quality of work
O_2	Low market share/reputational issues
O_3	Time overrun
O_4	Cost overrun

Table 4: Variable labels and names of the *Project Complexity* network.

Var. i	S_i	Time (S_i)	S_i^T	Time (S_i^T)
E1	0.48218	0.12326	0.48513	0.22200
E2	0.00502	0.12252	0.02151	0.21638
E3	0.12845	0.12056	0.13723	0.21051
E4	0.16822	0.12094	0.18604	0.21015
E5	0.00193	0.11998	0.00466	0.21473
E6	0.00093	0.12192	0.00224	0.22305
E7	0.09408	0.12048	0.10671	0.22147
E8	0.08309	0.12162	0.09678	0.22114

Table 5: *Project Management* network (Qazi et al., 2016): calculated Sobol sensitivity indices along with the required computation times.

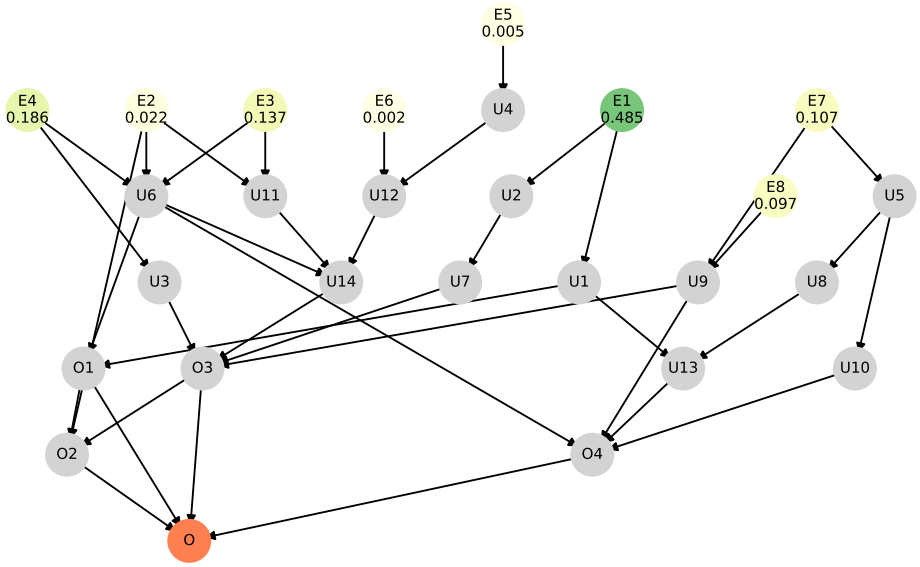


Figure 6: *Project Management* network (Qazi et al., 2016): sensitivity analysis on the output quantity of interest *utility*(O_1, O_2, O_3, O_4). Chance nodes are shown in gray, while evidential nodes are color-coded according to their respective total Sobol index (Sec. 2.2).

6. Conclusions

We have addressed efficient Sobol sensitivity analysis of BNs, for the general case when the output is the expected value of one of its nodes as a function of a set of other nodes. Our algorithms compute the two most important Sobol indices (variance components and total indices) by first transforming the input BN into a TN that captures the target function. Afterwards, via various manipulations of the TN, we compute each required variance using only a few inference queries (marginalization) over the modified TN. We showed this procedure to scale well: it is able to handle networks where both brute-force and Monte Carlo estimation would be impossible, as they would require millions of inference queries

over the BN. In addition, our method is exact, provided that exact inference is affordable on the modified TN.

The computation of the Sobol indices is of paramount importance for assessing the reliability of many engineering systems which, as already discussed, are often modeled via BNs. Such indices can guide research prioritization over the inputs of a system with the aim of decreasing the variability of the output, as illustrated by our examples. Thanks to this work, the computation of such indices for networks of medium-to-large size is now feasible and our implementation makes the methods available to practitioners. We plan to translate the code also in other languages, for example R, and to include it in the `bnmonitor` package for sensitivity and robustness analysis of BNs (Leonelli et al., 2021).

Our method is a generalization of previous sensitivity analysis algorithms for acyclic tensor networks such as the Tucker or tensor train decompositions. Our version works on cyclic networks, which arise naturally (due to moralization) when casting a general BN into a TN. The key enabling concepts are those of network squaring and quotient, thanks to which we can manipulate the function of interest so as to produce networks whose marginalization yields the desired statistics.

Although our methods directly apply to discrete BNs, they could also be used for networks where variables are continuous. One possibility is, as in Sections 5.1 and 5.2, to simulate from the continuous BN and to learn the discrete probabilities of the fixed network structure. Another possibility in the presence of continuous data is to discretize the data before learning the structure of the BN (e.g. Beuzen et al., 2018; Nojavan et al., 2017) and then using our algorithms on the discrete BN. As a last possibility, one could devise algorithms to compute approximated values of the Sobol indices directly on the continuous BN by evaluating the required expectations and variances at a discrete grid of points. Whether this approach is computationally feasible and performing well is the topic of current research.

As a venue of future research, we will explore the feasibility of computing more sophisticated variance-based sensitivity indices from BN-defined functions. These potentially include the mean dimension, the Shapley values, the dimension distribution, and other more recent metrics that are based on the method of Sobol.

References

A. Antoniadis, S. Lambert-Lacroix, and J.-M. Poggi. Random forests for global sensitivity analysis: A selective review. *Reliability Engineering & System Safety*, 206:107312, 2021.

I. Azzini, T. A. Mara, and R. Rosati. Comparison of two sets of Monte Carlo estimators of Sobol' indices. *Environmental Modelling & Software*, 144:105167, 2021.

R. Ballester-Ripoll, E. G. Paredes, and R. Pajarola. Tensor algorithms for advanced sensitivity metrics. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1172–1197, 2018.

R. Ballester-Ripoll, E. G. Paredes, and R. Pajarola. Sobol tensor trains for global sensitivity analysis. *Reliability Engineering & System Safety*, 183:311–322, 2019.

T. Beuzen, L. Marshall, and K. D. Splinter. A comparison of methods for discretizing continuous variables in Bayesian networks. *Environmental Modelling & Software*, 108: 61–66, 2018.

C. Bielza and P. Larrañaga. Bayesian networks in neuroscience: A survey. *Frontiers in Computational Neuroscience*, 8:131, 2014.

J. H. Bolt and S. Renooij. Local sensitivity of Bayesian networks to multiple simultaneous parameter shifts. In *European Workshop on Probabilistic Graphical Models*, pages 65–80. Springer, 2014.

E. Borgonovo. *Sensitivity analysis: An introduction for the management scientist*. Springer, 2017.

E. Borgonovo and E. Plischke. Sensitivity analysis: A review of recent advances. *European Journal of Operational Research*, 248(3):869–887, 2016.

A. Bretto. Hypergraph theory: An introduction. *Springer*, 2013.

B. Cai, X. Kong, Y. Liu, J. Lin, X. Yuan, H. Xu, and R. Ji. Application of Bayesian networks in reliability evaluation. *IEEE Transactions on Industrial Informatics*, 15(4): 2146–2157, 2018.

E. Castillo and U. Kjærulff. Sensitivity analysis in Gaussian Bayesian networks using a symbolic-numerical technique. *Reliability Engineering & System Safety*, 79(2):139–148, 2003.

E. Castillo, J. M. Gutiérrez, and A. S. Hadi. Sensitivity analysis in discrete Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 27(4):412–423, 1997.

H. Chan and A. Darwiche. Sensitivity analysis in Bayesian networks: From single to multiple parameters. In *Proocedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 317–325, 2004.

H. Chan and A. Darwiche. A distance measure for bounding probabilistic belief change. *International Journal of Approximate Reasoning*, 38:149–174, 2005.

G. Chastaing, F. Gamboa, and C. Prieur. Generalized Hoeffding-Sobol decomposition for dependent variables - application to sensitivity analysis. *Electronic Journal of Statistics*, 6:2420–2448, 2012.

V. M. H. Coupé and L. C. Van Der Gaag. Properties of sensitivity analysis of Bayesian belief networks. *Annals of Mathematics and Artificial Intelligence*, 36(4):323–356, 2002.

G. Damblin and A. Ghione. Adaptive use of replicated Latin Hypercube Designs for computing Sobol' sensitivity indices. *Reliability Engineering & System Safety*, 212:107507, 2021.

A. Darwiche. *Modeling and reasoning with Bayesian networks*. Cambridge University Press, 2009.

F. Dimaio, O. Scapinello, E. Zio, C. Ciarapica, S. Cincotta, A. Crivellari, L. Decarli, and L. Larosa. Accounting for safety barriers degradation in the risk assessment of oil and gas systems by multistate Bayesian networks. *Reliability Engineering & System Safety*, 216:107943, 2021.

N. C. Do and S. Razavi. Correlation effects? A major but often neglected component in sensitivity and uncertainty analysis. *Water Resources Research*, 56(3):e2019WR025436, 2020.

D. Douglas-Smith, T. Iwanaga, B. F. Croke, and A. J. Jakeman. Certain trends in uncertainty and sensitivity analysis: An overview of software tools and techniques. *Environmental Modelling & Software*, 124:104588, 2020.

B. Drury, J. Valverde-Rebaza, M. F. Moura, and A. de Andrade Lopes. A survey of the applications of Bayesian networks in agriculture. *Engineering Applications of Artificial Intelligence*, 65:29–42, 2017.

M. Eybpoosh, I. Dikmen, and M. Talat Birgonul. Identification of risk paths in international construction projects using structural equation modeling. *Journal of Construction Engineering and Management*, 137(12):1164–1175, 2011.

F. Goerlandt and S. Islam. A Bayesian Network risk model for estimating coastal maritime transportation delays following an earthquake in British Columbia. *Reliability Engineering & System Safety*, 214:107708, 2021.

M. A. Gómez-Villegas, P. Main, and R. Susi. The effect of block parameter perturbations in Gaussian Bayesian networks: Sensitivity and robustness. *Information Sciences*, 222: 439–458, 2013.

J. Guo, J. Cheng, E. Levina, G. Michailidis, and J. Zhu. Estimating heterogeneous graphical models for discrete data with an application to roll call voting. *The Annals of Applied Statistics*, 9(2):821, 2015.

J. Herman and W. Usher. SALib: An open-source python library for sensitivity analysis. *The Journal of Open Source Software*, 2(9), jan 2017.

W. Huang, Y. Zhang, X. Kou, D. Yin, R. Mi, and L. Li. Railway dangerous goods transportation system risk analysis: An interpretive structural modeling and Bayesian network combining approach. *Reliability Engineering & System Safety*, 204:107220, 2020.

B. Iooss and P. Lemaître. A review on global sensitivity analysis methods. In *Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications*, pages 101–122, Boston, MA, 2015. Springer US. ISBN 978-1-4899-7547-8.

B. Iooss and C. Prieur. Shapley effects for sensitivity analysis with correlated inputs: comparisons with Sobol' indices, numerical estimation and applications. *International Journal for Uncertainty Quantification*, 9(5):493–514, 2019.

C. Ji, X. Su, Z. Qin, and A. Nawaz. Probability analysis of construction risk based on noisy-or gate Bayesian networks. *Reliability Engineering & System Safety*, 217:107974, 2022.

U. Kjærulff. Triangulation of graphs - algorithms giving small total state space. Technical Report R 90-09, Department of Mathematics and Computer Science, Strandvejen, DK 9000 Aalborg, Denmark, 1990.

U. Kjærulff and L. C. van der Gaag. Making sensitivity analysis computationally efficient. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 317–325, 2000.

D. Koller, N. Friedman, and F. Bach. *Probabilistic graphical models: Principles and techniques*. MIT press, 2009.

J. H. P. Kwisthout, H. L. Bodlaender, and L. C. van der Gaag. The necessity of bounded treewidth for efficient inference in Bayesian networks. In *Proceedings of the 19th European Conference on Artificial Intelligence*, pages 237–242, 2010.

P. L. Salemi, E. Song, B. L. Nelson, and J. Staum. Gaussian Markov random fields for discrete optimization via simulation: Framework and algorithms. *Operations Research*, 67(1):250–266, 2019.

N. Lee and A. Cichocki. Fundamental tensor operations for large-scale data analysis using tensor network formats. *Multidimensional Systems and Signal Processing*, pages 1–40, 2017.

M. Leonelli and E. Riccomagno. A geometric characterisation of sensitivity analysis in monomial models. *arXiv:1901.02058*, 2019.

M. Leonelli, C. Görgen, and J. Q. Smith. Sensitivity analysis in multilinear probabilistic models. *Information Sciences*, 411:84–97, 2017.

M. Leonelli, R. Ramanathan, and R. L. Wilkerson. Sensitivity and robustness analysis in Bayesian networks with the bnmonitor R package. *arXiv:2107.11785*, 2021.

C. Li and S. Mahadevan. Sensitivity analysis of a Bayesian network. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part J: Mechanical Engineering*, 4(1): 011003, 2017.

C. Liu, Y. Wang, X. Li, Y. Li, F. Khan, and B. Cai. Quantitative assessment of leakage orifices within gas pipelines using a Bayesian network. *Reliability Engineering & System Safety*, 209:107438, 2021.

R. Liu and A. B. Owen. Estimating mean dimensionality of analysis of variance decompositions. *Journal of the American Statistical Association*, 101(474):712–721, 2006.

M. Luque and F. J. Díez. Variable elimination for influence diagrams with super value nodes. *International Journal of Approximate Reasoning*, 51(6):615–631, 2010.

Y.-P. Ma, I. Sudakov, C. Strong, and K. M. Golden. Ising model for melt ponds on arctic sea ice. *New Journal of Physics*, 21(6):063029, 2019.

T. A. Mara and W. E. Becker. Polynomial chaos expansion for sensitivity analysis of model output with dependent inputs. *Reliability Engineering & System Safety*, 214:107795, 2021.

A. Marrel, B. Iooss, B. Laurent, and O. Roustant. Calculations of Sobol indices for the Gaussian process metamodel. *Reliability Engineering & System Safety*, 94:742–751, 2009.

S. McLachlan, K. Dube, G. A. Hitman, N. Fenton, and E. Kyrimi. Bayesian networks in healthcare: distribution by medical condition. *Artificial Intelligence in Medicine*, page 101912, 2020.

F. Nojavan, S. S. Qian, and C. A. Stow. Comparative analysis of discretization methods in Bayesian networks. *Environmental Modelling & Software*, 87:64–71, 2017.

G. Ökten and Y. Liu. Randomized quasi-monte carlo methods in global sensitivity analysis. *Reliability Engineering & System Safety*, 210:107520, 2021.

A. B. Owen. Sobol' indices and Shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):245–251, 2014.

A. B. Owen, J. Dick, and S. Chen. Higher order Sobol' indices. *Information and Inference: A Journal of the IMA*, 3(1):59–81, 03 2014.

X. Pan, D. Zuo, W. Zhang, L. Hu, H. Wang, and J. Jiang. Research on human error risk evaluation using extended Bayesian networks with hybrid data. *Reliability Engineering & System Safety*, 209:107336, 2021.

A. Qazi, J. Quigley, A. Dickson, and K. Kirytopoulos. Project complexity and risk management (ProCRiM): Towards modelling project complexity driven risk paths in construction projects. *International Journal of Project Management*, 34(7):1183–1198, 2016.

D. A. Quintanar-Gago, P. F. Nelson, A. Diaz-Sanchez, and M. S. Boldrick. Assessment of steam turbine blade failure and damage mechanisms using a Bayesian network. *Reliability Engineering & System Safety*, 207:107329, 2021.

P. Rai. *Sparse Low Rank Approximation of Multivariate Functions – Applications in Uncertainty Quantification*. Doctoral thesis, École Centrale Nantes, Nov. 2014. URL `https://tel.archives-ouvertes.fr/tel-01143694`.

S. Razavi, A. Jakeman, A. Saltelli, C. Prieur, B. Iooss, E. Borgonovo, E. Plischke, S. Lo Piano, T. Iwanaga, W. Becker, et al. The future of sensitivity analysis: An essential discipline for systems modeling and policy support. *Environmental Modelling & Software*, 137:104954, 2021.

N. Robertson and P. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.

E. Robeva and A. Seigal. Duality of graphical models and tensor networks. *Information and Inference: A Journal of the IMA*, 8(2):273–288, 2018.

J. Rohmer. Uncertainties in conditional probability tables of discrete Bayesian belief networks: A comprehensive review. *Engineering Applications of Artificial Intelligence*, 88: 103384, 2020.

A. Saltelli, S. Tarantola, and F. Campolongo. Sensitivity anaysis as an ingredient of modeling. *Statistical Science*, 15(4):377–395, 2000.

A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis: The Primer*. John Wiley & Sons, Ltd., 2008.

A. Saltelli, K. Aleksankina, W. Becker, P. Fennell, F. Ferretti, N. Holst, S. Li, and Q. Wu. Why so many published sensitivity analyses are false: A systematic review of sensitivity analysis practices. *Environmental Modelling & Software*, 114:29–39, 2019.

M. Scutari. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22, 2010.

R. Sheikholeslami, S. Gharari, S. M. Papalexiou, and M. P. Clark. Viscous: A variance-based sensitivity analysis using copulas for efficient identification of dominant hydrological processes. *Earth and Space Science Open Archive*, page 37, 2020.

I. M. Sobol. Sensitivity estimates for nonlinear mathematical models (in Russian). *Mathematical Models*, 2:112–118, 1990.

B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964 – 979, 2008.

R. J. Trudeau. *Introduction to Graph Theory*. Courier Corporation, 2013.

S. Ung. Navigation risk estimation using a modified Bayesian network modeling - a case study in Taiwan. *Reliability Engineering & System Safety*, 213:107777, 2021.

K. Ye and L.-H. Lim. Tensor network ranks. *arXiv:1801.02662*, 2018.

Q. Yu, Â. P. Teixeira, K. Liu, H. Rong, and C. G. Soares. An integrated dynamic ship risk model based on Bayesian networks and evidential reasoning. *Reliability Engineering & System Safety*, 216:107993, 2021.

X. Zhang and S. Mahadevan. Bayesian network modeling of accident investigation reports for aviation safety assessment. *Reliability Engineering & System Safety*, 209:107371, 2021.

Y. Zhang and W. Weng. Bayesian network model for buried gas pipeline failure analysis caused by corrosion and external interference. *Reliability Engineering & System Safety*, 203:107089, 2020.

Z. Zhang, X. Yang, I. V. Oseledets, G. E. Karniadakis, and L. Daniel. Enabling high-dimensional hierarchical uncertainty quantification by ANOVA and tensor-train decomposition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(1):63–76, 2015.

Y. Zhou, X. Li, and K. F. Yuen. Holistic risk assessment of container shipping service based on Bayesian network modelling. *Reliability Engineering & System Safety*, page 108305, 2022.

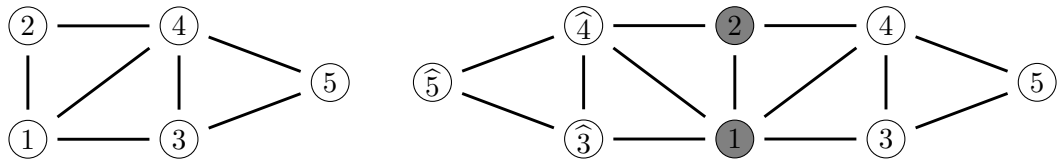
X. Zhu and B. Sudret. Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models. *Reliability Engineering & System Safety*, 214:107815, 2021.

E. Zio, M. Mustafayeva, and A. Montanaro. A Bayesian belief network model for the risk assessment and management of premature screen-out during hydraulic fracturing. *Reliability Engineering & System Safety*, 218:108094, 2022.

Appendix A. Graphical Representation of Squaring and Quotients of TNs

Proposition 5. *Let* $\mathbb{B} = [n] \setminus \mathbb{A}$ *and* $\mathcal{M}$ *be a TN . The square* $\mathcal{M}^2$ *w.r.t.* $\mathbb{A}$ *can be constructed as follows:*

- *the vertex set of* $\mathcal{M}^2$ *is* $\mathbb{A} \cup \mathbb{B} \cup \widehat{\mathbb{B}}$ *where* $\mathbb{A} \cap \widehat{\mathbb{B}} = \emptyset$, $\mathbb{B} \cap \widehat{\mathbb{B}} = \emptyset$ *and there exists a bijection* $\eta : \widehat{\mathbb{B}} \to \mathbb{B}$*;*



(a) A tensor network $\mathcal{M}$ (b) Squared network $\mathcal{M}^2$ with respect to $\{1, 2\}$

Figure A.7: Illustration of the squaring process.

- *the edge set of* $\mathcal{M}^2$ *is the union of the edge set of* $\mathcal{M}$ *with* $\mathbb{E}'$ *where* $\mathbb{E}'$ *includes for every* $\mathbb{D} \in \mathbb{C}$ *the hyperedge* $\{\mathbb{A} \cap \mathbb{D}\} \cup \eta(\mathbb{D} \setminus \mathbb{A})$*;*
- *if* $\mathbb{D} \subseteq \mathbb{A} \cup \mathbb{B}$ *then* $\phi_{\mathcal{M}^2}(\boldsymbol{y}_{\mathbb{D}}) = \phi_{\mathcal{M}}(\boldsymbol{y}_{\mathbb{D}})$*, otherwise* $\mathbb{D} \subseteq \mathbb{A} \cup \widehat{\mathbb{B}}$ *and* $\phi_{\mathcal{M}^2}(\boldsymbol{y}_{\mathbb{D}}) = \phi_{\mathcal{M}}(\boldsymbol{y}_{C \cap \eta(\widehat{\mathbb{B}})}, \boldsymbol{y}_{\mathbb{D} \cap \mathbb{A}})$.

Proof. For any $\boldsymbol{y}_{\mathbb{A}} \in \mathbb{Y}_{\mathbb{A}}$ we have

$$
\begin{aligned}
(\mathcal{M}_{\setminus \mathbb{A}}(\boldsymbol{y}_{\mathbb{A}}))^2 &= \left(\sum_{\boldsymbol{y}_{\mathbb{B}} \in \mathbb{Y}_{\mathbb{B}}} \prod_{\mathbb{D} \in \mathcal{P}(\mathbb{V}(\mathcal{M}))} \phi_{\mathcal{M}}(\boldsymbol{y}_{\mathbb{A} \cap \mathbb{D}}, \boldsymbol{y}_{\mathbb{B} \cap \mathbb{D}}) \right)^2 \\
&= \sum_{\boldsymbol{y}_{\mathbb{B}}, \boldsymbol{y}'_{\mathbb{B}} \in \mathbb{Y}_{\mathbb{B}}} \prod_{\mathbb{D} \in \mathcal{P}(\mathbb{V}(\mathcal{M}))} \phi_{\mathcal{M}}(\boldsymbol{y}_{\mathbb{A} \cap \mathbb{D}}, \boldsymbol{y}_{\mathbb{B} \cap \mathbb{D}}) \phi_{\mathcal{M}}(\boldsymbol{y}_{\mathbb{A} \cap \mathbb{D}}, \boldsymbol{y}'_{\mathbb{B} \cap \mathbb{D}}) \\
&= \sum_{\boldsymbol{y}_{\mathbb{B}}, \boldsymbol{y}'_{\mathbb{B}} \in \mathbb{Y}_{\mathbb{B}}} \prod_{\mathbb{D} \in \mathcal{P}(\mathbb{V}(\mathcal{M}))} \phi_{\mathcal{M}^2}(\boldsymbol{y}_{\mathbb{A} \cap \mathbb{D}}, \boldsymbol{y}_{\mathbb{B} \cap \mathbb{D}}) \phi_{\mathcal{M}^2}(\boldsymbol{y}_{\mathbb{A} \cap \mathbb{D}}, \boldsymbol{y}'_{\mathbb{B} \cap \eta(\widehat{\mathbb{D}})}) \\
&= \sum_{\boldsymbol{y}_{\mathbb{B} \cup \widehat{\mathbb{B}}} \in \mathbb{Y}_{\mathbb{B} \cup \widehat{\mathbb{B}}}} \prod_{\mathbb{D} \in \mathcal{P}(\mathbb{V}(\mathcal{M}^2))} \phi_{\mathcal{M}^2}(\boldsymbol{y}_{\mathbb{A} \cap \mathbb{D}}, \boldsymbol{y}_{(\mathbb{B} \cup \widehat{\mathbb{B}}) \cap \mathbb{D}}) \qquad \text{(A.1)}
\end{aligned}
$$

where the last equality follows by noticing that $\mathbb{E}(\mathcal{M}^2)$ does not include any element with vertices in both $\mathbb{B}$ and $\widehat{\mathbb{B}}$. The result easily follows by noticing that the rhs of Equation (A.1) is exactly $\mathcal{M}^2_{\setminus \mathbb{A}}(\boldsymbol{y}_{\mathbb{A}})$. □

Figure A.7 illustrates the squaring procedure of Proposition 5. A TN $\mathcal{M}$ has hypergraph with vertex set $[5]$ and hyperedges $\{1, 2, 4\}$, $\{1, 3, 4\}$ and $\{3, 4, 5\}$. This can be represented by the undirected graph in Figure A.7a. Suppose that we are interested in the squared TN $\mathcal{M}^2$ w.r.t. $\{1, 2\}$, which, using Proposition 5, can be constructed as the one in Figure A.7b. The idea is related to a known algorithm to compute the dot product between two tensor networks $\mathcal{M}$ and $\mathcal{M}'$ by contracting all nodes to yield a scalar $\langle \mathcal{M}, \mathcal{M}' \rangle$ (Lee and Cichocki, 2017). Our version sets $\mathcal{M} = \mathcal{M}'$ and skips the node contraction part.

The following proposition bounds the treewidth of $\mathcal{M}^2$ in terms of the original network $\mathcal{M}$'s treewidth. This is useful from the computational complexity point of view, as the inference problem is known to be more tractable when the treewidth is bounded Kwisthout et al. (2010).

Proposition 6. *Let $\mathcal{M}$ be a TN over variables $[n]$, and let $\mathcal{M}^2$ be its square over $\mathbb{A} \subseteq [n]$. Then,* $\text{treewidth}(\mathcal{M}^2) \leq 2 \cdot \text{treewidth}(\mathcal{M}) + 1$.

Proof. (sketch)

A junction tree $\mathcal{T}$ of $\mathcal{M}$ is a graph $(\mathbb{V}, E)$ satisfying these properties (Robertson and Seymour, 1986):

- $\mathcal{T}$ is a cluster graph of $\mathcal{M}$, i.e. each of its vertices is a subset of the original set of vertices: $v \subseteq [n] = \mathbb{A} \cup \mathbb{B}$ for all $v \in \mathbb{V}$.
- $\mathcal{T}$ includes all vertices of $\mathcal{M}$: $\bigcup_{v \in \mathbb{V}} v = \mathbb{A} \cup \mathbb{B}$.
- If (X_i, X_j) is an edge of $\mathcal{M}$, then there exists a $v \in \mathbb{V}$ such that $X_i, X_j \in v$.
- If two vertices v_i, v_j of $\mathbb{V}$ both contain a vertex X of $\mathcal{M}$, then all vertices v_k in the path from v_i to v_j also contain X.

The *treewidth* of $\mathcal{M}$ is defined as the smallest integer K such that there exists a junction tree $\mathcal{T}$ of $\mathcal{M}$ whose largest node contains $K+1$ vertices from $[n]$ (Robertson and Seymour, 1986).

We prove the proposition by noting that we can build a junction tree of $\mathcal{M}^2$, which we denote $\mathcal{T}^2 = (\mathbb{V}^2, E^2)$, by modifying $\mathcal{T}$ as follows. For every $v \in \mathbb{V}$ its corresponding vertex $v^2 \in \mathbb{V}^2$ is defined as $v^2 = \mathbb{A}_v \cup \mathbb{B}_v \cup \widehat{\mathbb{B}}_v$, where

$$\begin{cases} \mathbb{A}_v = \{X \in \mathbb{A} \,|\, X \in v\} \\ \mathbb{B}_v = \{X \in \mathbb{B} \,|\, X \in v\} \\ \widehat{\mathbb{B}}_v = \{X \in \widehat{\mathbb{B}} \,|\, \eta(X) \in v\}. \end{cases}$$

The edges are left untouched, i.e. $e = (v_i^2, v_j^2) \in E^2$ iff $(v_i, v_j) \in E$.

Since $\mathcal{M}^2$ closely mimics the topology of $\mathcal{M}$, it is straightforward to see that $\mathcal{T}^2$ is a junction tree of $\mathcal{M}^2$ as it satisfies all four required properties. To complete the proof, note that each vertex of $\mathcal{T}^2$ can only contain at most $2(K+1)$ elements of $\mathbb{A} \cup \mathbb{B} \cup \widehat{\mathbb{B}}$. Therefore, the treewidth of $\mathcal{M}^2$ is at most $2(K+1) - 1 = 2K + 1$. □

Proposition 7. *The quotient $\mathcal{M}/\mathcal{M}'$ is a TN such that:*

- *the vertex set of its hypergraph is the same as those of $\mathcal{M}$ and $\mathcal{M}'$*
- *the hyperedges of its hypergraph are the union of the hyperedges of $\mathcal{M}$ and $\mathcal{M}'$*
- *its potentials are $\phi_{\mathcal{M}/\mathcal{M}'}(\boldsymbol{y}) = \phi_{\mathcal{M}}(\boldsymbol{y})/\phi_{\mathcal{M}'}(\boldsymbol{y})$ for any $\boldsymbol{y} \in \mathbb{Y}$.*

Proof. For any $\boldsymbol{y} \in \mathbb{Y}$ we have

$$\frac{\mathcal{M}(\boldsymbol{y})}{\mathcal{M}'(\boldsymbol{y})} = \frac{\prod_{\mathbb{A}\in\mathcal{P}(\mathbb{V}(\mathcal{M}))}\phi_{\mathcal{M}}(\boldsymbol{y}_{\mathbb{A}})}{\prod_{\mathbb{A}\in\mathcal{P}(\mathbb{V}(\mathcal{M}'))}\phi_{\mathcal{M}'}(\boldsymbol{y}_{\mathbb{A}})} = \prod_{\mathbb{A}\in\mathcal{P}(\mathbb{V}(\mathcal{M}))}\frac{\phi_{\mathcal{M}}(\boldsymbol{y}_{\mathbb{A}})}{\phi_{\mathcal{M}'}(\boldsymbol{y}_{\mathbb{A}})}$$
$$= \prod_{\mathbb{A}\in\mathcal{P}(\mathbb{V}(\mathcal{M}/\mathcal{M}'))}\phi_{\mathcal{M}/\mathcal{M}'}(\boldsymbol{y}_{\mathbb{A}}) = (\mathcal{M}/\mathcal{M}')(\boldsymbol{y})$$

□

Appendix B. Proofs

Appendix B.1. Proof of Proposition 1

For the first statement and any $\boldsymbol{y}_{\setminus\mathbb{A}} = \mathbb{Y}_{\setminus\mathbb{A}}$,

$$(\mathcal{M}/\mathcal{M}'_{\mathbb{A}})_{\mathbb{A}}(\boldsymbol{y}_{\setminus\mathbb{A}}) = \sum_{\boldsymbol{y}_{\mathbb{A}}\in\mathbb{Y}_{\mathbb{A}}}(\mathcal{M}(\boldsymbol{y}_{\mathbb{A}},\boldsymbol{y}_{\setminus\mathbb{A}})/\mathcal{M}'_{\mathbb{A}}(\boldsymbol{y}_{\setminus\mathbb{A}}))$$
$$= \left(\sum_{\boldsymbol{y}_{\mathbb{A}}\in\mathbb{Y}_{\mathbb{A}}}\mathcal{M}(\boldsymbol{y}_{\mathbb{A}},\boldsymbol{y}_{\setminus\mathbb{A}})\right)/\mathcal{M}'_{\mathbb{A}}(\boldsymbol{y}_{\setminus\mathbb{A}}) = \mathcal{M}_{\mathbb{A}}(\boldsymbol{y}_{\setminus\mathbb{A}})/\mathcal{M}_{\mathbb{A}}(\boldsymbol{y}_{\setminus\mathbb{A}}).$$

The second statement follows immediately from the above and the definition of square network w.r.t. $\setminus\mathbb{A}$.

Appendix B.2. Proof of Proposition 2

For any $\boldsymbol{y}_{\mathbb{E}} \in \mathbb{Y}_{\mathbb{E}}$ we have

$$f(y) = \mathrm{E}_{\mathbb{U}\cup\mathbb{O}}[Y_{\mathbb{O}}|\boldsymbol{Y}_{\mathbb{E}} = \boldsymbol{y}_{\mathbb{E}}] = \sum_{\boldsymbol{y}_{\mathbb{O}\cup\mathbb{U}}\in\mathbb{Y}_{\mathbb{O}\cup\mathbb{U}}} y_{\mathbb{O}}\cdot\Pr(\boldsymbol{y}_{\mathbb{E}},\boldsymbol{y}_{\mathbb{U}\cup\mathbb{O}})/\Pr(\boldsymbol{y}_{\mathbb{E}})$$
$$= \sum_{\boldsymbol{y}_{\mathbb{O}\cup\mathbb{U}}\in\mathbb{Y}_{\mathbb{O}\cup\mathbb{U}}} y_{\mathbb{O}}\cdot\mathcal{B}(\boldsymbol{y}_{\mathbb{E}},\boldsymbol{y}_{\mathbb{U}\cup\mathbb{O}})/\mathcal{B}_{\mathbb{U}\cup\mathbb{O}}(\boldsymbol{y}_{\mathbb{E}})$$
$$= \sum_{\boldsymbol{y}_{\mathbb{O}\cup\mathbb{U}}\in\mathbb{Y}_{\mathbb{O}\cup\mathbb{U}}} \mathcal{M}(\boldsymbol{y}_{\mathbb{E}},\boldsymbol{y}_{\mathbb{U}\cup\mathbb{O}})/\mathcal{B}_{\mathbb{U}\cup\mathbb{O}}(\boldsymbol{y}_{\mathbb{E}}) = (\mathcal{M}/\mathcal{B}_{\mathbb{U}\cup\mathbb{O}})_{\mathbb{U}\cup\mathbb{O}}(\boldsymbol{y}_{\mathbb{E}}).$$

Appendix B.3. Proof of Proposition 3

Note that $\mathrm{E}_{\mathbb{A}}(\mathcal{M}(\boldsymbol{y}_{\setminus\mathbb{A}})) = \sum_{\boldsymbol{y}_{\mathbb{A}}\in\mathbb{Y}_{\mathbb{A}}}\mathcal{M}(\boldsymbol{y}_{\mathbb{A}},\boldsymbol{y}_{\setminus\mathbb{A}})\cdot\Pr(\boldsymbol{y}_{\mathbb{A}}|\boldsymbol{Y}_{\setminus\mathbb{A}} = \boldsymbol{y}_{\setminus\mathbb{A}})$ and $\Pr(\boldsymbol{y}_{\mathbb{D}}) = \mathcal{B}_{\mathbb{A}\cup\mathbb{B}}(\boldsymbol{y}_{\mathbb{D}})$ for any $\boldsymbol{y}_{\mathbb{D}} \in \mathbb{Y}_{\mathbb{D}}$. Therefore

$$
\begin{aligned}
\mathrm{E}_{\mathbb{A}}(\mathcal{M}_{\mathbb{B}}(\boldsymbol{y}_{\mathbb{D}})) &= \sum_{\boldsymbol{y}_{\mathbb{A}} \in \mathbb{Y}_{\mathbb{A}}} \mathcal{M}_{\mathbb{B}}(\boldsymbol{y}_{\mathbb{A}}, \boldsymbol{y}_{\mathbb{D}}) \cdot \Pr(\boldsymbol{y}_{\mathbb{A}} | \boldsymbol{Y}_{\mathbb{D}} = \boldsymbol{y}_{\mathbb{D}}) \\
&= \sum_{\boldsymbol{y}_{\mathbb{A}} \in \mathbb{Y}_{\mathbb{A}}} \mathcal{M}_{\mathbb{B}}(\boldsymbol{y}_{\mathbb{A}}, \boldsymbol{y}_{\mathbb{D}}) \cdot \Pr(\boldsymbol{y}_{\mathbb{A}}, \boldsymbol{y}_{\mathbb{D}}) / \Pr(\boldsymbol{y}_{\mathbb{D}}) \\
&= \sum_{\boldsymbol{y}_{\mathbb{A}} \in \mathbb{Y}_{\mathbb{A}}} (\mathcal{M}_{\mathbb{B}}(\boldsymbol{y}_{\mathbb{A}}, \boldsymbol{y}_{\mathbb{D}}) \cdot \mathcal{B}_{\mathbb{B}}(\boldsymbol{y}_{\mathbb{A}}, \boldsymbol{y}_{\mathbb{D}}) / \mathcal{B}_{\mathbb{A} \cup \mathbb{B}}(\boldsymbol{y}_{\mathbb{D}})) \\
&= (\mathcal{M}_{\mathbb{B}} \cdot \mathcal{B}_{\mathbb{B}} / \mathcal{B}_{\mathbb{A} \cup \mathbb{B}})_{\mathbb{A}}(\boldsymbol{y}_{\mathbb{D}}).
\end{aligned}
$$

Appendix B.4. Proof of Proposition 4

Combining Propositions. 1 to 3 and using the fact that $\mathcal{B}_{[n]} = 1$, we have

$$
\begin{aligned}
\mathrm{E}[f] = \mathrm{E}_{\mathbb{E}}[(\mathcal{M}/\mathcal{B}_{\setminus \mathbb{E}})_{\setminus \mathbb{E}}] &= ((\mathcal{M}/\mathcal{B}_{\setminus \mathbb{E}})_{\setminus \mathbb{E}} \cdot \mathcal{B}_{\setminus \mathbb{E}} / \mathcal{B}_{\setminus \mathbb{E} \cup \mathbb{E}})_{\mathbb{E}} \\
&= (\mathcal{M}_{\setminus \mathbb{E}} / \mathcal{B}_{\setminus \mathbb{E}} \cdot \mathcal{B}_{\setminus \mathbb{E}})_{\mathbb{E}} = (\mathcal{M}_{\setminus \mathbb{E}})_{\mathbb{E}} = \mathcal{M}_{[n]}.
\end{aligned}
$$

Appendix B.5. Global Variance: First Term (Sec. 4.4)

$$
\begin{aligned}
\mathrm{E}[f^2] &= \mathrm{E}_{\mathbb{E}}[(\mathcal{M}/\mathcal{B}_{\setminus \mathbb{E}})^2_{\setminus \mathbb{E}}] \text{ (using Prop. 2)} \\
&= ((\mathcal{M}/\mathcal{B}_{\setminus \mathbb{E}})^2_{\setminus \mathbb{E}} \cdot \mathcal{B}_{\setminus \mathbb{E}} / \mathcal{B}_{\setminus \mathbb{E} \cup \mathbb{E}})_{\mathbb{E}} \text{ (using Prop. 3)} \\
&= (\mathcal{M}^2_{\setminus \mathbb{E}} / \mathcal{B}_{\setminus \mathbb{E}} / \mathcal{B}_{\setminus \mathbb{E} \cup \mathbb{E}})_{\mathbb{E}} \text{ (using Prop. 1)} \\
&= (\mathcal{M}^2_{\setminus \mathbb{E}} / \mathcal{B}_{\setminus \mathbb{E}})_{\mathbb{E}} \qquad \text{(B.1)} \\
&= (\mathcal{M}^2 / \mathcal{B}_{\setminus \mathbb{E}})_{\mathbb{E} \cup \setminus \mathbb{E}}, \text{ (where the square of } \mathcal{M} \text{ is taken w.r.t. } \mathbb{E}). \qquad \text{(B.2)}
\end{aligned}
$$

Note the computational advantage of Equation (B.2) over Equation (B.1): the latter requires marginalization over variables $\setminus \mathbb{E}$, then $\mathbb{E}$, while the former marginalizes once over all variables, which gives marginalization heuristics full freedom to find the best order.

Appendix B.6. Variance Components: First Term (Sec. 4.5)

$$
\begin{aligned}
\mathrm{E}_i[\mathrm{E}^2_{\setminus i}[f]] &= \mathrm{E}_i[\mathrm{E}^2_{\setminus i}[(\mathcal{M}/\mathcal{B}_{\setminus \mathbb{E}})_{\setminus \mathbb{E}}]] \text{ (using Prop. 2)} \\
&= \mathrm{E}_i[((\mathcal{M}/\mathcal{B}_{\setminus \mathbb{E}})_{\setminus \mathbb{E}} \cdot \mathcal{B}_{\setminus \mathbb{E}} / \mathcal{B}_{\setminus \mathbb{E} \cup \setminus i})^2_{\setminus i}] \text{ (using Prop. 3)} \\
&= \mathrm{E}_i[(\mathcal{M}_{\setminus \mathbb{E}} / \mathcal{B}_{\setminus \mathbb{E}} \cdot \mathcal{B}_{\setminus \mathbb{E}} / \mathcal{B}_{\setminus \mathbb{E} \cup \setminus i})^2_{\setminus i}] \\
&= \mathrm{E}_i[(\mathcal{M}_{\setminus \mathbb{E}} / \mathcal{B}_{\setminus i})^2_{\setminus i}] \text{ (since } \setminus \mathbb{E} \cup \setminus i = \setminus i) \\
&= ((\mathcal{M}_{\setminus \mathbb{E}} / \mathcal{B}_{\setminus i})^2_{\setminus i} \cdot \mathcal{B}_{\setminus i} / \mathcal{B}_{\setminus i \cup \{i\}})_i \text{ (using Prop. 3)} \\
&= (\mathcal{M}^2_{\setminus i} / \mathcal{B}_{\setminus i})_i.
\end{aligned}
$$

Note that, since $\mathcal{M}_{\setminus i}$ is a one-variable network, $\mathcal{M}^2_{\setminus i}$ is most efficiently computed as $(\mathcal{M}_{\setminus i})^2$.