Lifted Query Answering in Gaussian Bayesian Networks

Mattis Hartwig HARTWIG@IFIS.UNI-LUEBECK.DE

Ralf Möller MOELLER@IFIS.UNI-LUEBECK.DE

Institute of Information Systems, University of Lübeck, Ratzeburger Allee 160, 23562 Lübeck, Germany

Abstract

Gaussian Bayesian networks are widely used for modeling behaviors of continuous random variables. Lifting exploits symmetries when dealing with large numbers of isomorphic random variables to support more compact representations and more efficient query answering. This paper presents a lifted construction and representation of a joint distribution derived from a Gaussian Bayesian network and a lifted query answering algorithm on the lifted joint distribution. To lift the query answering, needed algebraic operations that work fully in the lifted space are developed. A theoretical complexity analysis and experimental results show that both the lifted joint construction and the lifted query answering significantly outperform their grounded counterparts.

Keywords: gaussian bayesian networks; lifting; query answering; exact inference.

1. Introduction and Related Work

Modeling real-world systems from areas such as healthcare, bio-statistics, financial markets etc. requires formalisms to be able to represent continuous variables and relationships among them. Often, these areas involve many objects and uncertainties. Over the last three decades, probabilistic graphical models (PGMs) such as Bayesian Networks (BNs) have been used to model these random variables (randvars) and the relationships between them (Koller et al., 2009). Being able to efficiently answer queries to these models, also referred to as inference, has triggered a lot of research (Zhang and Poole, 1996; Salmerón et al., 2018).

Poole (2003) introduces first-order probabilistic inference, which exploits symmetries in a model by combining instances to reason with representatives. Exploiting these symmetries for isomorphic randvars is called lifting. Lifted inference has been an active research field in the past years (Kimmig et al., 2015). For a discrete PGM, Taghipour et al. (2013) have developed a lifted variable elimination algorithm and Braun and Möller (2016) have developed a lifted version of the junction tree algorithm. In the continuous setting, Choi et al. (2010) have developed a lifted version for variable elimination in factor graphs with Gaussian pairwise potentials. This paper applies the lifting concept to Gaussian Bayesian networks (GBNs) which have been introduced by Shachter and Kenley (1989) and widely applied across multiple fields (Cano et al., 2004; Froelich, 2015).

The paper contribution is threefold. First, we present an algorithm to construct a lifted representation of the joint distribution of a GBN. Second, we develop lifted algorithm for query answering significantly outperforming its ground counterpart. Third, we derive algebraic operations that can be fully computed in a lifted space, which can possibly be transferred to other use cases as well. Additionally the lifted query answering in GBNs builds the basis for further lifted research handling GBN based hybrid models (Lauritzen and Jensen, 2001; Madsen, 2008; Salmerón et al., 2018).

The remainder is structured as follows. Section 2 describes the preliminary definitions. Section 3 presents a lifted version for constructing the joint probability distribution of a GBN. Section 4 develops algebraic operations to work in a lifted space, which are used in Section 5 to develop a lifted query answering algorithm. Section 6 contains a complexity analysis, which is verified by an experimental evaluation in Section 7. The conclusion and next steps are discussed in Section 8.

2. Preliminaries

This section covers the preliminaries for GBNs, parameterised models and queries. Throughout the paper, we use bold symbols for vectors, sets, and matrices and thin symbols for scalars or elements.

2.1 Gaussian Bayesian Networks

Definition 1 *A BN is a directed acyclic graph whose vertices represent N randvars $V_i \in \mathbb{V}$ with $i = 1, ..., N$ and whose edges $\mathbb{U} \subset (\mathbb{V}, \mathbb{V})$ represent the dependencies between the randvars. The set of parents $\mathbf{Pa}(V_i)$ of a randvar $V_i \in \mathbb{V}$ is defined as the set of randvars $\boldsymbol{V}_k$ that have a directed edge to V_i.*

Definition 2 *A GBN is a BN where all randvars are normally distributed. The edges represent linear relationships between the randvars. The joint distribution can be factorized using the conditional probability distributions of V_i with $i = 1, ..., N$ given it parents $\boldsymbol{Pa}(V_i)$:*

$$P(V_i|\boldsymbol{Pa}(V_i)) \sim \mathcal{N}\left(\mu_{V_i} + \sum_{V_k \in \boldsymbol{Pa}(V_i)} \beta_{V_k,V_i}(v_k - \mu_{V_k}), \sigma^2_{V_i}\right), \tag{1}$$

where μ_{V_k} and μ_{V_i} are marginal means, $\sigma^2_{V_i}$ is the conditional variance and β_{V_k,V_i} represents the influence of parent V_k on its child V_i.

The joint probability distribution $P(\mathbb{V})$ over the randvars $\mathbb{V}$ is a multivariate normal distribution

$$P(\mathbb{V}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2}$$

where $\boldsymbol{\mu}$ denotes the mean vector $\boldsymbol{\Sigma}$ denotes the covariance matrix.

2.2 Parameterised Models

The idea of parameterised models is to work solely with representatives for a group of isomorphic randvars, sharing a probability distribution and parents. The isomorphic randvars are also called instances of a group. Logical variables (logvars) identify the groups and parameterise randvars to represent a set of isomorphic randvars. Logvars have a domain that contains the names of the instances. The model of all individual randvars is called ground model. We follow the notation and semantics introduced by Poole (2003), Taghipour et al. (2013) and Braun and Möller (2016).

Definition 3 *Let $\mathbb{V}$ be a set of randvars and $\mathbb{L}$ be a set of logvars. A PRV X is a syntactical construct of a randvar $V \in \mathbb{V}$ combined with a sequence of logvars $\boldsymbol{L} \subseteq \mathbb{L}$ into $V(\boldsymbol{L})$ to represent a set of randvars. If $\boldsymbol{L} = \emptyset$, the PRV is parameterless and constitutes a propositional randvar. The domain $D(L) = \{L^1, ..., L^H\}$ contains the instances of a logvar L. The domain of a sequence of logvars is defined as $D(\boldsymbol{L}) = \times_{L \in \boldsymbol{L}} D(L)$. Grounding a PRV X results in a set of randvars $\boldsymbol{gr}(X) = \{V_{L^1}, ..., V_{L^H}\}$.*

Each PRV X has a range denoted as $range(X)$ that containing possible values of X. Here, we work with continuous normally distributed randvars resulting in the general $range(X) = \mathbb{R}$.

2.3 Queries

Definition 4 *A query $P(\boldsymbol{Q}|\boldsymbol{E} = \boldsymbol{e})$ consists of a query set $\boldsymbol{Q} \subseteq \mathbb{V}$, and a set of events $\{E_h = e_h\}_{h=1}^{O}$ with $e_h \in \mathbb{R}$ where $E_h \in \mathbb{V}$ and O is the number of observations.*

Based on Definition 4, query answering in probabilistic graphical model allows for two different types of queries. First a marginal probability distribution, i.e., $P(\boldsymbol{Q})$, and second a a conditional probability distribution given a set of events $\boldsymbol{E} = \boldsymbol{e}$, i.e., $P(\boldsymbol{Q}|\boldsymbol{E} = \boldsymbol{e})$.

3. Lifted Joint Distribution

In this section, we describe a lifted algorithm for constructing a lifted representation of the joint multivariate Gaussian distribution for a parameterised GBN.

3.1 Parameterised Gaussian Bayesian Networks

Instead of N ground randvars $\mathbb{V}$, a parameterised GBN contains M PRVs $\mathbb{X}$. Each PRV $X_s \in \mathbb{X}$ with $s = 1, ..., M$ has a PRV mean η_s and a variance λ_s. Each edge between a parent X_u and a child X_s has a linear influence β_{X_u, X_s} that describes the linear relationship between the PRVs analogously to the propositional GBN described in Section 2.1. Grounding a PRV X_s leads to a set of ground randvars $\mathbf{V}_s = \boldsymbol{gr}(X_s)$ with $|\mathbf{V}_s| = |D(L_s)|$ that have the same mean and variance. Grounding a topologically ordered list of PRVs results in a topologically ordered list of ground randvars because the parent-child relationships for each ground randvar are defined by the parent child relationships of the corresponding PRV. If we consider a propositional random variable, we can interpret it as a PRV with a logvar that has a domain size $|D(L)| = 1$. With that interpretation, there are three different kinds of relationships between two PRVs X_s and X_t with $t = 1, ..., M$ possible:

No Parent: A PRV has no parent PRV resulting in no influencing factors on the PRV.

Disjoint logvar sets: The sequence of logvars $\mathbf{L}_s$ of the PRV X_s is disjoint from the sequence of logvars $\mathbf{L}_t$ of the parent PRV X_t. This results in a relation from every randvar in $\boldsymbol{gr}(X_s)$ to every randvar in $\boldsymbol{gr}(X_t)$.

Overlapping logvar sets: The sequence of logvars L_s of the PRV X_s is overlapping with the sequence of logvars L_t of the parent PRV X_t. Grounding the relation between the two PRVs X_s and X_t sharing a sequence of logvars $\mathbf{L}_s \cap \mathbf{L}_t = \mathbf{L}_{shared}$ results in a ground relation where each child node is influenced by all $|D(\mathbf{L}_t \setminus \mathbf{L}_s)|$ parents that share the same instantiation of the shared logvars.

Overlaps result in a break of the symmetry structure, because not all random variables belonging to the same PRV share the same set of parents anymore. The overlaps can be eliminated by grounding the overlapping logvars and creating new PRVs. For the rest of the paper, we assume that overlaps are eliminated. We denote $lif(V_i)$ to identify a PRV that contains a specific ground randvar V_i, $\boldsymbol{lif}(\mathbf{V})$ to identify the PRV sequence for a sequence of ground randvars $\mathbf{V}$, $\boldsymbol{red}(\boldsymbol{lif}(\mathbf{V}))$ to identify a set within a sequence of PRVs and $one(X_s)$ to identify any single randvar of a PRV X_s.

3.2 Constructing a Lifted Joint

Shachter and Kenley (1989) have developed a recursive algorithm to convert a GBN into a multivariate normal distribution. This section describes a lifted version of the algorithm. If the number of isomorphic instances is high, the covariance matrix of the multivariate normal distribution is filled with many duplicate values. We develop a lifted representation that allows to eliminate duplicates for more efficient memory usage and faster calculations.

In the propositional case, the randvars are brought into a topological ordering. Then the covariance matrix between all randvars is created recursively. The covariance $Cov(V_i, V_j)$ between randvars V_i and V_j with $i, j = 1, ..., N$ where $i \neq j$ is symmetric, i.e.,$Cov(V_i, V_j) = Cov(V_j, V_i)$, and recursively calculated with

$$Cov(V_i, V_j) = \sum_{V_k \in Pa(V_i)} Cov(V_j, V_k)\beta_{V_k,V_j} + \delta_{V_i,V_j}\sigma^2_{V_i}. \tag{3}$$

where δ_{V_i,V_j} is the Kronecker delta that is only one if $V_i = V_j$ and otherwise zero.

Instead of looking into propositional randvars V_i and V_j, we are now looking at two sets of equally behaving randvars $\mathbf{V}_s$ and randvars $\mathbf{V}_t$ grouped into PRVs X_s and X_t with $s, t = 1, ..., M$, respectively. Parents of X_s are denoted as $X_u \in \boldsymbol{Pa}(X_s)$. As described above, we can get the number of randvars in PRVs by looking at the domain size of its logvars, e.g., $|D(\mathbf{L}_s)|$. In short, we denote the number of randvars of a PRV X_s as $|X_s| = |D(\mathbf{L}_s)|$.

In the ground case, we would calculate the covariance between all $|X_s|$ randvars $\mathbf{V}_s = \boldsymbol{gr}(X_s)$ and all $|X_t|$ randvars $\mathbf{V}_t = \boldsymbol{gr}(X_t)$. The set of parents $\boldsymbol{Pa}(V_i)$ for any randvar V_i contains for every parent PRV $X_u \in \boldsymbol{Pa}(lif(V_i))$, $|X_u|$ isomorphic randvars. We can reformulate Equation 3 into

$$Cov(V_i, V_j) = \sum_{X_u \in \boldsymbol{Pa}(lif(V_i))} \left(\sum_{V_k \in \boldsymbol{gr}(X_u)} Cov(V_j, V_k)\beta_{V_k,V_j} \right) + \delta_{V_i,V_j}\sigma^2_{V_i}. \tag{4}$$

The sum in brackets is working on ground level. If the recursive $Cov(V_j, V_k)$ was equal for all V_j and V_k we could calculate it once and multiply it with the number of randvars $|X_u|$ to replace the sum. However, if V_j is equal to V_k, the recursive $Cov(V_j, V_k)$ returns a different value caused by the last term containing the Kronecker delta. This prevents us from multiplying $Cov(V_j, V_k)$ with the number of randvars $|X_u|$ to replace the sum. To get rid of the sum in brackets nevertheless, we differentiate between two cases.

Case 1: If the randvar V_j is no parent of the randvar V_i, the $\delta_{V_j,V_k}\sigma^2_{V_j}$ in the recursive call $Cov(V_j, V_k)$ is always zero, resulting in a fully equal covariance $Cov(V_j, V_k)$ for all $V_l \in \boldsymbol{gr}(X_k)$, which reduces the second summation to a product between any covariance $Cov(V_j, V_k)$ and the number of parent randvars $|X_k|$:

$$Cov(V_i, V_j) = \sum_{X_u \in \boldsymbol{Pa}(lif(V_i))} Cov(V_j, one(X_u))|X_u| + \delta_{V_i,V_j}\sigma^2_{V_i}. \tag{5}$$

Case 2: If the randvar V_j is a parent of the randvar V_i, then exactly one randvar $V_k \in \boldsymbol{gr}(X_u)$ will be equal to randvar V_j, which would result in a different covariance $Cov(V_j, V_k)$. The key is that if randvar V_j is a parent of randvar V_i, all other randvars $\boldsymbol{gr}(lif(V_j))$ of the corresponding PRV $lif(V_j)$ are also parents of randvars $gr(lif(V_i))$. This means, independent of the specific randvar in the covariance function, there will always be exactly one covariance $Cov(V_j, V_k)$ where the $\delta_{V_j,V_k}\sigma^2_{V_j}$ is nonzero.

Based on the two cases, we can reformulate Equation 4 into

$$Cov(V_i, V_j) = \sum_{X_u \in \boldsymbol{Pa}(lif(V_i)} \Big(Cov(V_j, one(X_u))|X_u| + \delta_{lif(V_j),X_u}\sigma_{V_j} \Big) \beta_{u,j} + \delta_{V_i,V_j}\sigma^2_{V_i}. \tag{6}$$

Since the sum over $X_k \in \boldsymbol{Pa}(lif(V_i))$ is equal for all combinations between randvars $\boldsymbol{gr}(lif(V_i))$ and randvars $\boldsymbol{gr}(lif(V_j))$, we can calculate and store the sum separately as

$$CovL(X_s, X_t) = \sum_{X_u \in \boldsymbol{Pa}(X_s)} \left(CovL(X_t, X_u)|X_u| + \delta(X_t, X_u)\lambda_{X_t}\right)\beta_{X_t, X_u}. \tag{7}$$

The last term $\delta_{V_i,V_j}\sigma^2_{V_i}$ of Equation 6 only needs to be added if ground covariance between two equal randvars V_i and V_i is calculated. We can use the PRV covariance $CovL$ to calculate a ground covariance with

$$Cov(V_i, V_j) = CovL(lif(V_i), lif(V_j)) + \delta(V_i, V_j)\lambda_{lif(V_i)}. \tag{8}$$

Equations 7 and 8 show that we can calculate the ground covariance matrix only using the lifted PRV covariance $CovL$ and the PRV variances λ_{X_s}. All PRV covariances $CovL$ can be stored in a $M \times M$-dimensional matrix $\boldsymbol{\rho}$ and the variances λ_{X_s} can be stored into a M-dimensional vector λ. With this information, we can create the ground covariance matrix anytime.

In addition to the covariance matrix, the multivariate Gaussian also needs a mean vector $\boldsymbol{\mu}$. The ground mean-vector contains the means μ_i of all randvars $V_i \in \mathbb{V}$. Since the randvars within a PRV X_s have the same mean, the M-dimensional PRV mean vector $\boldsymbol{\eta}$ is a lifted version of $\boldsymbol{\mu}$. We get the mean for a ground randvar V_i by expanding the lifted mean vector $\boldsymbol{\eta}$ as $\mu_{V_i} = \eta_{lif(V_i)}$.

Summarized, to store all information of the lifted joint distribution over M PRVs $X_1, ..., X_M$, we need an M-dimensional lifted mean vector $\boldsymbol{\eta}$, an $M \times M$-dimensional PRV covariance matrix $\boldsymbol{\rho}$, an M-dimensional variance correction vector $\boldsymbol{\lambda}$ and an M-dimensional cardinality vector $\boldsymbol{\tau}$.

4. Working with Liftable Matrices

In this section, we develop a formal framework for working with the lifted joint distribution. We start by formalizing the structure of ground and lifted covariance matrices. Afterwards, we discuss how to perform matrix operations like matrix multiplication and matrix inversion in the lifted space.

4.1 Liftable Matrices

Definition 5 *A symmetric $N \times N$-dimensional matrix*

$$\mathbf{Z} = \begin{bmatrix} \boldsymbol{B}_{1,1} & \dots & \boldsymbol{B}_{1,M} \\ \vdots & \ddots & \vdots \\ \boldsymbol{B}_{M,1} & \dots & \boldsymbol{B}_{M,M} \end{bmatrix} \tag{9}$$

has a liftable structure if it consists of $M \times M$ blocks $\boldsymbol{B}_{s,t}$, with $s, t = 1, ..., M$, where each block on the diagonal of $\mathbf{Z}$ is a square matrix that follows $\boldsymbol{B}_{s,s} = \rho_{s,s}\boldsymbol{J}^{\tau_s \times \tau_s} + \lambda_s \boldsymbol{I}^{\tau_s \times \tau_s}$, and each block on the off-diagonals of $\mathbf{Z}$ follows $\boldsymbol{B}_{s,t} = \rho_{s,t}\boldsymbol{J}^{\tau_s \times \tau_t}$,for $s \neq t$, where $\boldsymbol{J}$ is the all-ones matrix, $\boldsymbol{I}$ the identify matrix and, $\lambda_1, ...\lambda_M$ $\tau_1, ...\tau_M$ and $\rho_{1,1}, ...\rho_{M,M}$ are scalars.

In the following, we call a block with the on-diagonal structure a OnD block and a block with the off-diagonal structure a OffD block. Bringing this into relation with the PRVs from the previous section, each block $\boldsymbol{B}_{s,t}$ can represent the ground covariance between the randvars in $\boldsymbol{gr}(X_s)$ and in $\boldsymbol{gr}(X_t)$. Definition 5 shows that each block $\boldsymbol{B}_{s,t}$ has a lifted representation by a PRV covariance

$\rho_{s,t}$ and if the block is a OnD block, a variance λ_s. The information of a matrix with liftable structure can be stored in a $M \times M$-dimensional matrix denoted as $\boldsymbol{\rho}$, an M-dimensional vector $\boldsymbol{\lambda}$, and an M-dimensional vector $\boldsymbol{\tau}$ as done in the previous section for the lifted representation of the covariance matrix. Consequently, a row of OffD blocks can be represented in a lifted way by a row of PRV covariances $\boldsymbol{\rho}$ and a submatrix of $\mathbf{Z}$ by a submatrix of $\boldsymbol{\rho}$ and a subvector of $\boldsymbol{\lambda}$. Getting the $N \times N$-dimensional ground covariance matrix can be done by inserting the values from $\boldsymbol{\rho}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\tau}$ into the block generating equations of Definition 5.

As a simplification, we denote $\boldsymbol{\Lambda} = \boldsymbol{dia}(\boldsymbol{\lambda})$, where $\boldsymbol{dia}(\boldsymbol{\lambda})$ contains the scalar λ_s at position $\Lambda_{s,s}$ and zeroes everywhere else. We can rewrite the two block equations from Definition 5 into $\boldsymbol{B}_{X_s,X_t} = \rho_{s,t}\boldsymbol{J} + \Lambda_{s,t}\boldsymbol{I}$, because the scalar $\Lambda_{s,t}$ is zero for $s \neq t$.

4.2 Lifted Matrix Multiplication

Lifted storing a matrix with a liftable structure alone is not enough for lifted query answering. We also need to perform calculations without grounding the stored matrix, as we will see in the next section. Basic matrix algebra shows that adding or subtracting block matrices $\boldsymbol{Z}^1$ and $\boldsymbol{Z}^2$ with matching dimensions can be done by adding and subtracting their matrices $\boldsymbol{\rho}^1$ and $\boldsymbol{\rho}^2$ and vectors $\boldsymbol{\lambda}^1$ and $\boldsymbol{\lambda}^2$. In the following, we work on lifted matrix multiplication.

Lemma 6 *Let $\boldsymbol{B}_{s,t}$ and $\boldsymbol{B}_{t,w}$ with $s, t, w = 1, ..., M$ be two blocks of the block matrix $\mathbf{Z}$, then it is*

$$\boldsymbol{B}_{s,t} \cdot \boldsymbol{B}_{t,w} = x\boldsymbol{J}^{\tau_s \times \tau_t} + y\boldsymbol{I}^{\tau_t \times \tau_w}, \tag{10}$$

where x and y can be calculated only using the lifted representation by

$$x = \rho_{s,t}\rho_{t,w}\tau_t + \rho_{s,t}\Lambda_{t,w} + \rho_{t,w}\Lambda_{s,t} \text{ and } y = \Lambda_{s,t}\Lambda_{t,w}. \tag{11}$$

Proof We use the rule for multiplying all-ones matrices $J^{G\times H}J^{H\times K} = HJ^{G\times K}$, where G, H and K are scalars that define the matrix dimensions, in the multiplication of the two blocks

$$\begin{aligned}\boldsymbol{B}_{s,t} \cdot \boldsymbol{B}_{t,w} &= (\rho_{s,t}\boldsymbol{J}^{\tau_s \times \tau_t} + \Lambda_{s,t}\boldsymbol{I}^{\tau_s \times \tau_t}) \cdot (\rho_{t,w}\boldsymbol{J}^{\tau_t \times \tau_w} + \Lambda_{t,w}\boldsymbol{I}^{\tau_t \times \tau_w}) \\ &= \rho_{s,t}\rho_{t,w}\tau_t\boldsymbol{J}^{\tau_s \times \tau_w} + \rho_{s,t}\Lambda_{t,w}\boldsymbol{J}^{\tau_s \times \tau_w} + \rho_{t,w}\Lambda_{s,t}\boldsymbol{J}^{\tau_s \times \tau_w} + \Lambda_{s,t}\Lambda_{t,w}\boldsymbol{I}^{\tau_s \times \tau_w} \\ &= (\rho_{s,t}\rho_{t,w}\tau_t + \rho_{s,t}\Lambda_{t,w})\boldsymbol{J}^{\tau_s \times \tau_w} + \Lambda_{s,t}\Lambda_{t,w}\boldsymbol{I}^{\tau_s \times \tau_w}.\end{aligned} \tag{12}$$

■

The results can be stored in a lifted way by storing the values of x and y of the new block. Based on Lemma 6, the value y of the result is nonzero only if two OnD structured blocks are multiplied and is zero if one of the multiplied blocks is a OffD structured block. In the algorithm developed in the next section, we need operations that work on multiple blocks in a lifted way.

Lemma 7 *For $s, t = 1, ..., M$, let $\boldsymbol{B}_{s,s}$ be an OnD structured $\tau_s \times \tau_s$-dimensional block of $\boldsymbol{Z}$, $\boldsymbol{B}_{s+1:M,s}$ be a column of blocks with*

$$\boldsymbol{B}_{s+1:M,s} = \begin{bmatrix} \boldsymbol{B}_{s+1,s} \\ \vdots \\ \boldsymbol{B}_{M,s} \end{bmatrix} \tag{13}$$

and $\boldsymbol{B}_{s:M,s:M}$ be a block matrix with

$$\boldsymbol{B}_{s:M,s:M} = \begin{bmatrix} \boldsymbol{B}_{s,s} & \dots & \boldsymbol{B}_{s,M} \\ \vdots & \ddots & \vdots \\ \boldsymbol{B}_{M,s} & \dots & \boldsymbol{B}_{M,M} \end{bmatrix}. \tag{14}$$

We can perform the multiplication of (i) a single block with a row or column of blocks ($\boldsymbol{B}_{s+1:M,s} \cdot \boldsymbol{B}_{s,s}$), (ii) a row of blocks with a column blocks ($\boldsymbol{B}_{s+1:M,s} \cdot \boldsymbol{B}^T_{s+1:M,s}$) and (iii) a row or column of blocks with a block matrix ($\boldsymbol{B}_{s+1:M,s} \cdot \boldsymbol{B}_{s:M,s:M}$) in lifted space.

Proof For (i) with block matrix multiplication and Lemma 6 it is

$$\boldsymbol{B}_{s+1:M,s} \cdot \boldsymbol{B}_{s,s} = \begin{bmatrix} \boldsymbol{B}_{s+1,s}\boldsymbol{B}_{s,s} \\ \vdots \\ \boldsymbol{B}_{M,s}\boldsymbol{B}_{s,s} \end{bmatrix} = \begin{bmatrix} (\rho_{s+1,s}\rho_{s,s}\tau_s + \rho_{s+1,s}\Lambda_{s,s})\boldsymbol{J} \\ \vdots \\ (\rho_{M,s}\rho_{s,s}\tau_s + \rho_{M,s}\Lambda_{s,s})\boldsymbol{J} \end{bmatrix}. \tag{15}$$

The output column of blocks can be stored using a column vector $\boldsymbol{\rho}_{result1}$ which can be directly calculated by $\boldsymbol{\rho}_{result1} = \boldsymbol{\rho}_{s+1:M,s}\rho_{s,s}\tau_s + \boldsymbol{\rho}_{s+1:M,s}\Lambda_{s,s}$. Analogously, we can calculate the multiplication of a row vector with a block in lifted space. For (ii) it is

$$\boldsymbol{B}_{s+1:M,s} \cdot \boldsymbol{B}^T_{s+1:M,s} = \begin{bmatrix} \boldsymbol{B}_{s+1,s}\boldsymbol{B}_{s,s+1} & \dots & \boldsymbol{B}_{s+1,s}\boldsymbol{B}_{s,M} \\ \vdots & \ddots & \vdots \\ \boldsymbol{B}_{M,s}\boldsymbol{B}_{s,s+1} & \dots & \boldsymbol{B}_{M,s}\boldsymbol{B}_{s,M} \end{bmatrix}, \tag{16}$$

which can analogously be stored in matrix $\boldsymbol{\rho}_{result2}$ and calculated in a lifted way by $\boldsymbol{\rho}_{result2} = \boldsymbol{\rho}_{s+1:M,s}\boldsymbol{\rho}^T_{s+1:M,s}\tau_s$. For (iii) it is

$$\boldsymbol{B}_{s+1:M,s+1:M} \cdot \boldsymbol{B}_{s+1:M,s} = \begin{bmatrix} \sum_{w=s+1}^{M} \boldsymbol{B}_{s+1,w}\boldsymbol{B}_{w,s} \\ \vdots \\ \sum_{w=s+1}^{M} \boldsymbol{B}_{M,w}\boldsymbol{B}_{w,s} \end{bmatrix} = \begin{bmatrix} \left(\sum_{w=i+1}^{M} \rho_{s+1,w}\rho_{w,s}\tau_w + \lambda_{s+1}\rho_{s+1,s}\right)\boldsymbol{J} \\ \vdots \\ \left(\sum_{w=i+1}^{M} \rho_{M,w}\rho_{w,s}\tau_w + \lambda_{M}\rho_{M,s}\right)\boldsymbol{J} \end{bmatrix} \tag{17}$$

which can be calculated and stored in a column vector in a lifted way. For $\boldsymbol{B}_{s+1:M,s}T \cdot \boldsymbol{B}_{s+1:M,s+1:M}$ it is analogous. ■

4.3 Lifted Matrix Inversion

In this section we describe an analytic and lifted formula for inverting a OnD structured matrix analogue to Henderson and Searle (1981).

Lemma 8 *Let $\boldsymbol{L}$ be an $G \times G$ OnD structured matrix of the form $\boldsymbol{L} = a\boldsymbol{J} + b\boldsymbol{I}$. Then the inverse of L can be calculated analytically by $\boldsymbol{L}^{-1} = x\boldsymbol{J} + y\boldsymbol{I}$, where*

$$x = -\frac{a}{b(aG+b)} \text{ and } y = \frac{1}{b}. \tag{18}$$

Proof To prove this lemma, we follow the definition of an inverse $\boldsymbol{L}\boldsymbol{L}^{-1} = \boldsymbol{I}$. Solving the equation $\boldsymbol{L}\boldsymbol{L}^{-}1 = (a\boldsymbol{J} + b\boldsymbol{I})(x\boldsymbol{J} + y\boldsymbol{I}) = \boldsymbol{I}$ is equivalent to solving the linear equation system (LES)

$$(a+b)(x+y) + (G-1)ax = 1, \tag{19}$$

$$(a+b)x + a(x+y) + (G-2)ax = 0. \tag{20}$$

Solving the LES for x and y results in Equation 18. ■

5. Lifted Query Answering

This section covers query answering using the lifted version of the joint distribution.

5.1 Lifted Answering for a Marginal Query

As defined in Section 2.3 a marginal query $P(\boldsymbol{Q})$ is a query without evidence. Obtaining a marginal distribution of a multivariate normal distribution is trivial. One can simply select the means $\boldsymbol{\mu_Q}$ and covariance matrix $\boldsymbol{\Sigma_{QQ}}$ that are corresponding to the queried randvars and insert them into the probability distribution $P(\boldsymbol{Q}) = \mathcal{N}(\boldsymbol{\mu_Q}; \boldsymbol{\Sigma_{QQ}})$. For the lifted case, we select the means $\boldsymbol{\mu_Q}$ and covariance matrix $\boldsymbol{\Sigma_{QQ}}$ of the corresponding PRVs as described in Section 3.1 by

$$\boldsymbol{\mu_Q} = \boldsymbol{\eta_{lif(Q)}} \text{ and } \boldsymbol{\Sigma_{QQ}} = \boldsymbol{\rho_{lif(Q),lif(Q)}} + \boldsymbol{dia}(\boldsymbol{\lambda_{lif(Q)}}). \tag{21}$$

5.2 Lifted Answering for a Conditional Query

In the propositional case, calculating a conditional probability distribution $P(\boldsymbol{Q}|\boldsymbol{E} = \boldsymbol{e})$ for a multivariate Gaussian follows

$$\boldsymbol{\mu}^* = \boldsymbol{\mu_Q} + \boldsymbol{\Sigma_{QE}}\boldsymbol{\Sigma_{EE}^{-1}}(\boldsymbol{e} - \boldsymbol{\mu_E}) \text{ and } \boldsymbol{\Sigma}^* = \boldsymbol{\Sigma_{QQ}} + \boldsymbol{\Sigma_{QE}}\boldsymbol{\Sigma_{EE}^{-1}}\boldsymbol{\Sigma_{EQ}}. \tag{22}$$

The distribution $\mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$ is the query answer (Eaton, 1983). Matrix inversion and matrix multiplication are driving the complexity of the query answering with evidence. Since the ground covariance matrix $\boldsymbol{\Sigma}$ has a liftable structure also $\boldsymbol{\Sigma_{EE}}$ has a liftable structure and will be stored in a lifted way. For this section the $\boldsymbol{\rho^E}$, $\boldsymbol{\lambda^E}$ and $\boldsymbol{\tau^E}$ only contain values for PRVs that have randvars in the evidence. $\boldsymbol{\tau^E}$ contains the number of randvars for each PRV in the evidence. We can partition the $O \times O$ ground matrix $\boldsymbol{\Sigma_{EE}}$, where O is the number of observed variables, into $K \times K$ blocks $\boldsymbol{B}_{q,r}$ where $q, r = 1, ..., K$ based on the structure of the K PRVs in the evidence as detailed in Definition 5.

Lemma 8 allows us to invert a single block on the diagonal but not the whole matrix $\boldsymbol{\Sigma_{EE}}$. To break down the inversion we use the block matrix inversion formula (Bernstein, 2009)

$$\begin{bmatrix} \tilde{\boldsymbol{A}} & \tilde{\boldsymbol{B}} \\ \tilde{\boldsymbol{C}} & \tilde{\boldsymbol{D}} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{O} & \mathbf{P} \\ \mathbf{Q} & \mathbf{R} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{A}}^{-1} + \tilde{\boldsymbol{A}}^{-1}\tilde{\mathbf{B}}\tilde{\mathbf{F}}^{-1}\tilde{\mathbf{C}}\tilde{\mathbf{A}}^{-1} & -\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{B}}\tilde{\mathbf{F}}^{-1} \\ -\tilde{\mathbf{F}}^{-1}\tilde{\mathbf{C}}\tilde{\mathbf{A}}^{-1} & \tilde{\mathbf{F}}^{-1} \end{bmatrix}, \text{ where } \tilde{\mathbf{F}} = \tilde{\mathbf{D}} - \tilde{\mathbf{C}}\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{B}}. \tag{23}$$

To get the inverted covariance matrix $\boldsymbol{\Sigma_{EE}^{-1}}$ we use lifted matrix multiplication and the block matrix inversion recursively. In the following, we describe the recursive block matrix inversion. Pseudocode can be found in Algorithm 1.

Step 1: The input for the function is the lifted representation $\boldsymbol{\rho}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\tau}$ of a liftable matrix $\boldsymbol{Z}$. In the first call the (the first recursion step) it is $\boldsymbol{\rho^E}$, $\boldsymbol{\lambda^E}$ and $\boldsymbol{\tau^E}$ representing $\boldsymbol{\Sigma_{EE}}$. If the matrix $\boldsymbol{\rho}$ has only one element, Z would consist of only one block in OnD structure and we can use Lemma 8 to calculate the lifted representation of $\boldsymbol{Z}^{-1}$, where $\boldsymbol{Z}^{-1}$ is again OnD structured. The resulting $\boldsymbol{\rho}_{res}$, $\boldsymbol{\lambda}_{res}$ and $\boldsymbol{\tau}$ are returned and the function call terminates. If $\boldsymbol{\rho}$ has more than one element the algorithm continues.

Step 2: The grounded matrix $\boldsymbol{Z}$ would be split into four blocks based on Equation 23. These four 4 blocks are constructed of the $K \times K$ (where K is decreased by one in each recursion step) blocks forming the matrix $\boldsymbol{Z}$. Matrix $\tilde{\boldsymbol{A}}$ consists of the first block ($\tilde{\boldsymbol{A}} = \boldsymbol{B}_{1,1}$, $\rho_{\tilde{\boldsymbol{A}}} = \rho_{1,1}$, $\lambda_{\tilde{\boldsymbol{A}}} = \lambda_1$) the rest is constructed analogously.

Step 3: The inversion function is called for the lifted representation of $\tilde{\boldsymbol{A}}$ and Step 1 will directly return the lifted inverse. The lifted version $\boldsymbol{\rho}_{\tilde{\boldsymbol{F}}}$, $\boldsymbol{\lambda}_{\tilde{\boldsymbol{F}}}$ and $\boldsymbol{\tau}_{\tilde{\boldsymbol{F}}}$ of matrix $\tilde{\boldsymbol{F}}$ can be calculated using the lifted multiplication rules from Lemma 7 where $\boldsymbol{\lambda}_{\tilde{\boldsymbol{F}}} = \boldsymbol{\lambda}_{\tilde{\boldsymbol{D}}}$ and $\boldsymbol{\tau}_{\tilde{\boldsymbol{F}}} = \boldsymbol{\tau}_{\tilde{\boldsymbol{D}}}$. The inversion algorithm is recursively called for $\boldsymbol{\rho}_{\tilde{\boldsymbol{F}}}$, $\boldsymbol{\lambda}_{\tilde{\boldsymbol{F}}}$ and $\boldsymbol{\tau}_{\tilde{\boldsymbol{F}}}$.

Step 4: Once the recursive call returns the lifted inverse for $\boldsymbol{\rho}_{\tilde{\boldsymbol{F}}}$, $\boldsymbol{\lambda}_{\tilde{\boldsymbol{F}}}$ and $\boldsymbol{\tau}_{\tilde{\boldsymbol{F}}}$, the blocks the lifted representations for $\mathbf{O}$, $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{R}$ can be calculated using Lemma 7. Based on Equation 23, the four lifted representations of $\mathbf{O}$, $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{R}$ are combined into a $\boldsymbol{\rho}_{\boldsymbol{Z}^{-1}}$, $\boldsymbol{\lambda}_{\boldsymbol{Z}^{-1}}$ and $\boldsymbol{\tau}_{\boldsymbol{Z}^{-1}}$ and returned.

With Lemma 7, the conditional covariance matrix $\boldsymbol{\Sigma}^*$ from Equation 22 can be calculated fully lifted using the result of the recursive lifted matrix inversion. For the conditional mean $\boldsymbol{\mu}^*$, we need to calculate the observed deviation from the mean $(\boldsymbol{e} - \boldsymbol{\mu_E})$ on ground level, because the evidence can be different for the observed ground randvars. Even if the evidence is different across randvars it is multiplied with the same factor because $\boldsymbol{\Sigma_{QE}}\boldsymbol{\Sigma_{EE}^{-1}}$ is equal for all ground randvars belonging to the same PRV. We can group the evidence $\boldsymbol{e}$ by the K PRVs $\boldsymbol{X^E} = \boldsymbol{red}(\boldsymbol{lif}(\boldsymbol{E}))$ and convert the ground vector in a K-dimensional vector

$$\begin{bmatrix} \sum_{E_h \in \boldsymbol{gr}(\boldsymbol{X}_1^{\boldsymbol{E}})} e_h \\ \vdots \\ \sum_{E_h \in \boldsymbol{gr}(\boldsymbol{X}_K^{\boldsymbol{E}})} e_h \end{bmatrix} - \boldsymbol{\eta}_{1:K} \tag{24}$$

that can be used by the lifted multiplication rules to calculate the conditional mean $\boldsymbol{\mu}^*$. The summation is the only operation that needs to happen on ground level.

6. Complexity Analysis

This section covers the run-time and space complexity of creating the joint distribution and of answering conditional probability queries. We have N randvars in the GBN combined into M PRVs.

6.1 Complexity for Constructing the Joint

Constructing the ground version of the joint probability distribution calculates the $N \times N$-dimensional covariance matrix by iterating in two loops over the randvars. In each step, the covariance between two randvars V_i and V_j is calculated by taking into account all parents of either V_i or V_j depending on which is first in a topological ordering. If we denote T as the average number of parents in the GBN, we get a time complexity of $O(N^2T)$. Depending on the network type, T can be dependent

Algorithm 1 Lifted recursive block matrix inversion

procedure LIFTEDINVERSION($\boldsymbol{\rho}, \boldsymbol{\lambda}, \boldsymbol{\tau}$) $\triangleright$ lifted representation as an input
  if $\boldsymbol{\lambda}.size = 1$ **then**
    $\lambda_{inv}, \rho_{inv} \leftarrow 1/\lambda, -\rho/(\lambda(\rho\tau + \lambda))$ $\triangleright$ here, ρ, λ and τ are scalars
    return $\rho_{inv}, \lambda_{inv}, \tau$
  $\rho_{\tilde{\boldsymbol{A}}}, \lambda_{\tilde{\boldsymbol{A}}}, \tau_{\tilde{\boldsymbol{A}}} \leftarrow \rho_{1,1}, \lambda_1, \tau_1$
  $\boldsymbol{\rho}_{\tilde{\boldsymbol{B}}}, \boldsymbol{\rho}_{\tilde{\boldsymbol{C}}} \leftarrow \boldsymbol{\rho}_{1,2:K}, \boldsymbol{\rho}_{2:K,1}$
  $\boldsymbol{\rho}_{\tilde{\boldsymbol{D}}}, \boldsymbol{\lambda}_{\tilde{\boldsymbol{D}}}, \boldsymbol{\tau}_{\tilde{\boldsymbol{D}}} \leftarrow \boldsymbol{\rho}_{2:K,2:K}, \boldsymbol{\lambda}_{2:K}, \boldsymbol{\tau}_{2:K}$
  $\rho_{\tilde{\boldsymbol{A}}^{-1}}, \lambda_{\tilde{\boldsymbol{A}}^{-1}}, \tau_{\tilde{\boldsymbol{A}}^{-1}} \leftarrow$ LIFTEDINVERSION($\rho_{\tilde{\boldsymbol{A}}}, \lambda_{\tilde{\boldsymbol{A}}}, \tau_{\tilde{\boldsymbol{A}}}$) $\triangleright$ recursive Call
  $\boldsymbol{\rho}_{\tilde{\boldsymbol{F}}} \leftarrow \boldsymbol{\rho}_{\tilde{\boldsymbol{D}}} - \tau_{\tilde{\boldsymbol{A}}}(\tau_{\tilde{\boldsymbol{A}}}\rho_{\tilde{\boldsymbol{A}}^{-1}} + \lambda_{\tilde{\boldsymbol{A}}^{-1}})\boldsymbol{\rho}_{\tilde{\boldsymbol{C}}}\boldsymbol{\rho}_{\tilde{\boldsymbol{B}}}$
  $\boldsymbol{\lambda}_{\tilde{\boldsymbol{F}}}, \boldsymbol{\tau}_{\tilde{\boldsymbol{f}}} \leftarrow \boldsymbol{\lambda}_{\tilde{\boldsymbol{D}}}, \boldsymbol{\tau}_{\tilde{\boldsymbol{D}}}$
  $\rho_{\tilde{\boldsymbol{F}}^{-1}}, \boldsymbol{\lambda}_{\tilde{\boldsymbol{F}}^{-1}}, \boldsymbol{\tau}_{\tilde{\boldsymbol{F}}^{-1}} \leftarrow$ LIFTEDINVERSION($\boldsymbol{\rho}_{\tilde{\boldsymbol{F}}}, \boldsymbol{\lambda}_{\tilde{\boldsymbol{F}}}, \boldsymbol{\tau}_{\tilde{\boldsymbol{F}}}$) $\triangleright$ recursive Call
  $\rho_{\boldsymbol{O}}, \boldsymbol{\rho}_{\boldsymbol{P}}, \boldsymbol{\rho}_{\boldsymbol{Q}}, \boldsymbol{\rho}_{\boldsymbol{R}} \leftarrow$ CALCULATELIFTED
  $\boldsymbol{\lambda}_{inv}, \boldsymbol{\rho}_{inv} \leftarrow$ STACK($\lambda_{\tilde{\boldsymbol{A}}^{-1}}, \boldsymbol{\lambda}_{\tilde{\boldsymbol{F}}^{-1}}$), STACK($\rho_{\boldsymbol{O}}, \boldsymbol{\rho}_{\boldsymbol{P}}, \boldsymbol{\rho}_{\boldsymbol{Q}}, \boldsymbol{\rho}_{\boldsymbol{R}}$)
  return $\boldsymbol{\rho}_{inv}, \boldsymbol{\lambda}_{inv}, \boldsymbol{\tau}$ $\triangleright$ lifted representation as an output

or independent of N. Resulting in a lower bound of $\Omega(N^2)$ and a upper bound $O(N^3)$. The space complexity of storing the covariance matrix $\boldsymbol{\Sigma}$ and the mean vector $\boldsymbol{\mu}$ is $O(N^2)$.

In the presented lifted version, the two loops are depending on M to calculate the $M \times M$-dimensional PRV covariance $\boldsymbol{\rho}$. The calculation of the individual covariances $\rho_{s,t}$ uses the number of randvars as a scalar for multiplication. The avg. number of parent PRVs is driving the complexity analogously to the ground case resulting in a lower bound of $\Omega(M^2)$ and a upper bound $O(M^3)$. The space complexity is in $O(M^2)$. Summarized, the space and time complexity for constructing the lifted joint distribution is solely dependent on the number of PRVs.

6.2 Complexity for Conditional Query Answering

We denote N_Q as the number of querried randvars and N_E as the number of evidence randvars. Furthermore we denote M_Q and M_E as the number of PRVs involved in the N_Q and N_E randvars respectively. The matrix multiplication and inversion are driving the complexity of the query answering. We are aware that there are matrix inversion and multiplication algorithms that have a run-time complexity of less than $O(N^3)$ but for simplicity and without changing the overall argumentation we take $O(N^3)$ as an upper bound (Le Gall, 2014). In the ground case, the matrix inversion has a complexity of $O(N_E^3)$ and the matrix multiplication a complexity of $O(N_E^2 N_Q)$. If the query contains more query variables, the multiplication dominates the complexity and if the query contains more evidence the inversion dominates the complexity.

In the lifted case, the matrix inversion is constant for one block but the algorithm requires $2(M_E - 1)$ recursive calls to invert $\boldsymbol{\Sigma_{EE}}$ resulting in a linear complexity of $O(M_E)$. The lifted matrix multiplication within the recursive calls has a upper bound of $O(M_E^3)$ because there will be no matrix multiplication involving bigger than $M_E \times M_E$-dimensional matrices resulting in a runtime complexity for the matrix inversion of $O(M_E^4)$. Analogously to the ground case, the matrix multiplications are in $O(M_E^2 M_Q)$. The final calculation for the conditional mean vector $\boldsymbol{\mu}^*$ involve the summation a ground version of the evidence vector $\boldsymbol{e}$ resulting a complexity of $O(N_E M_Q)$. If $N_E >> M_E, M_Q$, which is often the case when working with lifted models, the complexity of

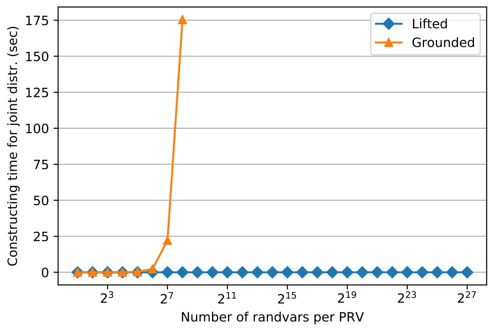

Figure 1: Eval. for constructing the joint dis.

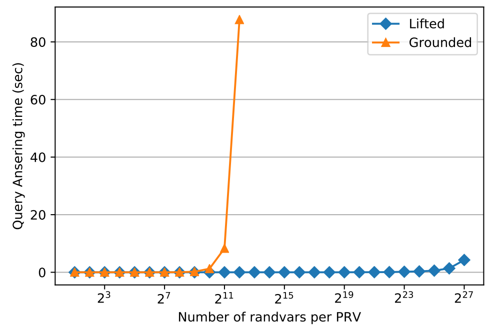

Figure 2: Eval. for query answering

calculating the $\boldsymbol{\mu}^*$-vector dominates. But instead of working in a cubic complexity as in the ground case, we reduced it to a linear complexity of N_E. If N_E, M_E and M_Q are in the same order of magnitude, the lifted version stays in a complexity of $O(M_E^4)$ or $O(M_E^2 M_Q)$.

7. Evaluation

Summarized, we verify the theoretical complexity analysis in the experiments. First, we evaluate the lifted construction of the joint distribution described in Section 3. Second, we evaluate the conditional query answering as a whole including the lifted matrix inversion. For each step, we set up a experiment containing 4 PRVs with exponentially increasing number of ground randvars (from 2 to 2^{27}). For the conditional query answering, we introduced evidence for half of the randvars and queried the other half. In the ground case we stopped if the runtime got extraordinary high. The results can be found in Figures 1 and 2. For both benchmarks, the lifted version is significantly faster. The plots show that the lifted construction of the joint distribution is independent of the number of randvars within the PRVs. Answering a conditional probability query is not fully independent of the number of randvars. As described in the previous section the number of evidence variables has a linear influence on the runtime. The influence can be seen in Figure 2 when the number of randvars is getting very high.

8. Conclusion and Outlook

In this paper, we propose a new lifted version of an algorithm for constructing a joint multivariate Gaussian distribution for a given Gaussian Bayesian network and a new exact lifted querying algorithm using the lifted representation of the multivariate Gaussian distribution. Additionally, we develop algebraic operations working in the lifted space that can possibly be transferred to other use cases as well. The lifted algorithms both significantly outperform existing ground level algorithms, which is proven theoretically and verified in a experimental evaluation. Performance increases can enable use cases from various fields that involve modeling the behavior for large numbers continuous random variables. Developing similar algorithms to handle hybrid Bayesian networks containing both discrete and continuous random variables and preventing the grounding of overlapping logvar sequences are promising directions for future research.

References

D. S. Bernstein. *Matrix mathematics: theory, facts, and formulas*. Princeton university press, 2009.

T. Braun and R. Möller. Lifted junction tree algorithm. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 30–42. Springer, 2016.

R. Cano, C. Sordo, and J. M. Gutiérrez. Applications of Bayesian networks in meteorology. In *Advances in Bayesian networks*, pages 309–328. Springer, 2004.

J. Choi, D. J. Hill, and E. Amir. Lifted inference for relational continuous models. In *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

M. L. Eaton. Multivariate statistics: a vector space approach. *John Wiley & Sons, Inc.*, 1983.

W. Froelich. Forecasting daily urban water demand using dynamic Gaussian Bayesian network. In *International conference: beyond databases, architectures and structures*, pages 333–342. Springer, 2015.

H. V. Henderson and S. R. Searle. On deriving the inverse of a sum of matrices. *Siam Review*, 23 (1):53–60, 1981.

A. Kimmig, L. Mihalkova, and L. Getoor. Lifted graphical models: a survey. *Machine Learning*, 99(1), 2015. ISSN 15730565. doi: 10.1007/s10994-014-5443-2.

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

S. L. Lauritzen and F. Jensen. Stable local computation with conditional Gaussian distributions. *Statistics and Computing*, 11(2):191–203, apr 2001. ISSN 1573-1375. doi: 10.1023/A:1008935617754.

F. Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303, 2014.

A. L. Madsen. Belief update in CLG Bayesian networks with lazy propagation. *International Journal of Approximate Reasoning*, 49(2):503–521, 2008.

D. Poole. First-order probabilistic inference. In *IJCAI*, volume 3, pages 985–991, 2003.

A. Salmerón, R. Rumí, H. Langseth, T. Nielsen, and A. Madsen. A Review of Inference Algorithms for Hybrid Bayesian Networks. *Journal of Artificial Intelligence Research*, 62:799–828, 2018. doi: 10.1613/jair.1.11228.

R. D. Shachter and C. R. Kenley. Gaussian Influence Diagrams. *Management Science*, 35(5): 527–550, 1989. doi: 10.1287/mnsc.35.5.527.

N. Taghipour, D. Fierens, J. Davis, and H. Blockeel. Lifted Variable Elimination: Decoupling the Operators from the Constraint Language. *Journal of Artificial Intelligence Research*, 47:393–439, jul 2013. ISSN 1076-9757. doi: 10.1613/jair.3793.

N. L. Zhang and D. Poole. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, 1996.