Handling Overlaps When Lifting Gaussian Bayesian Networks

Mattis Hartwig[1] , **Tanya Braun**[1] , **Ralf Möller**[2]
[1]Institute of Information Systems, University of Lübeck, Lübeck, Germany
[2] German Research Centre for Artificial Intelligence, Lübeck, Germany
{hartwig, braun}@ifis.uni-luebeck.de, ralf.moeller@dfki.de

Abstract

Gaussian Bayesian networks are widely used for modeling the behavior of continuous random variables. Lifting exploits symmetries when dealing with large numbers of isomorphic random variables. It provides a more compact representation for more efficient query answering by encoding the symmetries using logical variables. This paper improves on an existing lifted representation of the joint distribution represented by a Gaussian Bayesian network (lifted joint), allowing overlaps between the logical variables. Handling overlaps without grounding a model is critical for modelling real-world scenarios. Specifically, this paper contributes (i) a lifted joint that allows overlaps in logical variables and (ii) a lifted query answering algorithm using the lifted joint. Complexity analyses and experimental results show that — despite overlaps — constructing a lifted joint and answering queries on the lifted joint outperform their grounded counterparts significantly.

1 Introduction and Related Work

Modeling real-world systems in areas such as healthcare, biostatistics, or financial markets requires formalisms to use continuous variables and relationships among them. These areas may also involve many objects in relation to each other, affected by uncertainties. Probabilistic graphical models (PGMs) such as Bayesian Networks (BNs) have been used to model conditional independence assumptions regarding these random variables (randvars) [Koller *et al.*, 2009]. Being able to efficiently answer queries in these models, also often referred to as inference, has triggered a lot of research [Zhang and Poole, 1996; Blei *et al.*, 2017; Salmerón *et al.*, 2018].

To handle many objects efficiently, Poole [2003] introduces first-order probabilistic inference, which exploits symmetries in a model by combining indistinguishable instances using logical variables (logvars) to reason with representatives for the represented instances. Using a compact representation and performing inference with representatives is also referred to as lifting and has been an active research field in the past years [Kimmig *et al.*, 2015; Sharma *et al.*, 2018; Holtzen *et al.*, 2019]. For a discrete PGM, Taghipur et al. [2013] present a lifted variable elimination algorithm and Braun and Möller [2016] a lifted version of the junction tree algorithm by Lauritzen and Spiegelhalter [1988]. In the continuous setting, Choi et al. [2010] present a lifted version for variable elimination in factor graphs with Gaussian pairwise potentials. Hartwig and Möller [2020] apply the concept of lifting to Gaussian Bayesian networks (GBNs), which have been introduced by Shachter and Kenley [1989] and widely applied across multiple fields [Cano *et al.*, 2004; Grzegorczyk, 2010; Huang *et al.*, 2011; Froelich, 2015].

The current approach by Hartwig and Möller [2020] forbids overlaps between logvar sequences, which reduces the applicability dramatically. This paper extends the approach by generalizing existing and developing new algebraic operations to handle overlaps. Example 1 contains a simple scenario that currently cannot be handled in a lifted way, but will be supported by the techniques introduced in this paper.

Example 1 (Simple GBN with overlaps)**.** *Patients (P) have a disease severity level (S) which is influenced by getting a specific dose (D) of a medicine (M). Each medicine (M) has an effectiveness (E). The logvars here are the patients (P) and the medicines (M). The logvars combined with the randvars group together isomorphic instances of patients getting doses of medicine and medicines being effective, leading to so called parameterized randvars (PRVs)* $E(M)$*,* $D(P, M)$*, and* $S(P)$*. Overlaps are present if PRVs share the same logvar, e.g.,* $D(P, M)$ *and* $E(M)$ *share logvar* M*. The GBN puts the PRVs into relation in the form of* $E(M) \rightarrow D(P, M) \rightarrow S(P)$*, where* M *as well as* P *overlap between PRVs.*

Specifically, this paper contributes (i) a lifted representation of the joint distribution of a Gaussian BN with symmetries, allowing overlaps in logical variables and (ii) a lifted query answering approach using this lifted joint. Complexity analyses and experimental results show that – despite overlaps – constructing and answering queries on the lifted joint outperform their grounded counterparts significantly.

The remainder of the paper is structured as follows. Section 2 describes preliminary definitions and recaps the lifting approach by Hartwig and Möller [2020]. Section 3 presents the construction of the lifted joint, Section 4 introduces the updated rules for arithmetic operations with the lifted joint, and Section 5 explains query answering. Section 6 and Section 7 present a theoretical and experimental look at the contributions. We end with the conclusion and next steps.

2 Preliminaries and Notations

This section covers notations and summarizes the result of the work by Hartwig and Möller [2020]. Notation-wise, we follow Taghipour *et al.* [2013], Braun and Möller [2016], and Hartwig and Möller [2020]. Throughout this paper, we use bold symbols for vectors, sets, sequences, and matrices, and thin symbols for scalars or individual elements. In abuse of notation, we use set operations for sequences, where we apply the operation to the set containing the sequence elements.

2.1 Gaussian Bayesian Networks

Based on BNs this section defines GBNs and specifies their semantics.

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

Definition 2. *A* GBN *is a BN where all randvars are continuous and normally distributed. The edges represent linear relationships between the randvars. As in BNs, the joint density can be factorized using the conditional probability densities of $V_t, t = 1, ..., N$ given it parents $\boldsymbol{Pa}(V_t)$:*

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

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

The joint probability distribution $P(\mathbb{V})$ over randvars $\mathbb{V}$ is defined by a multivariate Gaussian distribution: $P(\mathbb{V}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ denotes the mean vector and $\boldsymbol{\Sigma}$ the covariance matrix. This distribution can be constructed from a GBN using the algorithm by Shachter and Kenley [1989], which requires a topological ordering of the randvars in the GBN. They use a transition matrix, here denoted as $\boldsymbol{T}$, where a non-zero entry at $T_{k,t} = \beta_{k,t}$ means that there is a parent-child relationship between V_k and V_t. The matrix $\boldsymbol{T}$ simply contains all β values to allow for matrix operations. The covariance matrix $\boldsymbol{\Sigma}$ is inductively built using $\boldsymbol{T}$. It is for any t and $s = 1, ..., t-1$:

$$\Sigma_{11} = \sigma_{V_1} \tag{2}$$

$$\boldsymbol{\Sigma}_{st} = \boldsymbol{\Sigma}_{ss}\boldsymbol{T}_{st} \tag{3}$$

$$\boldsymbol{\Sigma}_{ts} = \boldsymbol{\Sigma}_{st}^T \tag{4}$$

$$\Sigma_{tt} = \boldsymbol{\Sigma}_{ts}\boldsymbol{T}_{st} + \sigma_t \tag{5}$$

2.2 Parameterized Models

The idea of parameterized models is to work with representatives for a group of randvars sharing a probability distribution and parents. Logvars identify groups and parameterize randvars to represent a set of isomorphic randvars. The aim of lifting is to work solely with PRVs as representatives for the randvars belonging to the PRV.

Definition 3. *Let $\mathbb{V}$ be a set of randvars and $\mathbb{L}$ be a set of logvar names. 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 $\boldsymbol{L} = \varnothing$, the PRV is parameterless and constitutes a propositional randvar. The domain $D(L) = \{l^1, ..., l^H\}$ contains the instances of 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 with a logvar L results in a set of randvars $\boldsymbol{gr}(X) = \{V_{l^1}, ..., V_{l^H}\}$.*

Each PRV X also has a range denoted as $range(X)$ that contains the possible values of X. Since we work with continuous normally distributed randvars, $range(X) = \mathbb{R}$ in general. Instead of a discrete conditional probability table, we specify a mean η and variance λ for each PRV X.

2.3 Parameterized Gaussian Bayesian Networks

Instead of N ground randvars $\mathbb{V}$, a parameterized GBN contains M PRVs $\mathbb{X}$. Each PRV $X_i \in \mathbb{X}$, $i = 1, ..., M$, has a PRV mean η_i and a variance λ_i. 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 of Definition 2. Grounding a PRV X_i leads to a set of ground randvars $\mathbf{V}_i = \boldsymbol{gr}(X_i)$ with $|\mathbf{V}_i| = |D(\boldsymbol{L}_i)|$ 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. A propositional randvar V can be interpreted as a PRV $V(L)$ with $|D(L)| = 1$. There are three different kinds of relationships between two PRVs X_i and X_j:

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

Disjoint logvar sets: The logvars $\mathbf{L}_i$ of the parent PRV X_i and the logvars $\mathbf{L}_j$ of the child PRV X_j are disjoint. Disjoint logvar sets result in a relationship from every randvar in $\boldsymbol{gr}(X_i)$ to every randvar in $\boldsymbol{gr}(X_j)$.

Overlapping logvar sets: The logvars $\mathbf{L}_i$ of the parent PRV X_i and the logvars $\mathbf{L}_j$ of the child PRV X_j overlap, i.e., $\mathbf{L}_{shared} := \mathbf{L}_i \cap \mathbf{L}_j \neq \emptyset$. Grounding results in a relation where each child node is influenced by all $|D(\mathbf{L}_i \setminus \mathbf{L}_j)|$ parents that share the same grounding of $\mathbf{L}_{shared}$.

For ease of notation, we assume a global sorting Ω of all model logvars such that every logvar sequence uses the global sorting for ordering its logvars. This sorting places no restriction on the expressivity of the model. In the course of this paper, we need the following three helper functions.

Definition 4. *For two sequences of logvars $\boldsymbol{L}_i$ and $\boldsymbol{L}_j$, we define a function $seq(\boldsymbol{L}_i, \boldsymbol{L}_j)$ that returns the combined sequence of logvars according to the global sorting Ω. Given $L \in seq(\boldsymbol{L}_i, \boldsymbol{L}_j)$, we define*

$$dim(L, \boldsymbol{L}_i) = \begin{cases} |D(L)| & \textit{if } L \in \boldsymbol{L}_i \\ 1 & \textit{otherwise} \end{cases} \tag{6}$$

$$ov(L, \boldsymbol{L}_i, \boldsymbol{L}_j) = \begin{cases} 1 & \textit{if } L \in (\boldsymbol{L}_i \cap \boldsymbol{L}_j) \\ 0 & \textit{otherwise.} \end{cases} \tag{7}$$

2.4 Queries

Query answering concerns queries for (conditional) marginal probabilities or distributions of a set of randvars (given a set of events as evidence), defined as follows.

Definition 5. *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.*

2.5 Current Lifting Approach

Hartwig and Möller [2020] lift the algorithm by Shachter and Kenley [1989] for parameterized GBNs with disjoint logvar sets between parent and child nodes. The lifted representation of the covariance matrix is given by the M-dimensional vector $\boldsymbol{\lambda}$ and the $M \times M$-dimensional matrix $\boldsymbol{\rho}$, where $\boldsymbol{\lambda}$ is filled with the individual λ values of the PRVs and all elements of $\boldsymbol{\rho}$ are calculated as follows without groundings:

$$\rho_{X_i,X_j} = \sum_{X_u \in \boldsymbol{Pa}(X_j)} \left(\rho_{X_i,X_u}|D(\mathbf{L}_u)| + \delta_{X_u,X_j}\lambda_{X_j}\right)\beta_{X_u,X_j}, \tag{8}$$

where δ is the Kronecker delta function that equals one if $X_u = X_j$. The lifted representation can be grounded by

$$\Sigma_{V_s,V_t} = \rho_{lif(V_s),lif(V_t)} + \delta_{V_s,V_t}\lambda_{lif(V_s)}, \tag{9}$$

where $lif(V_s)$ refers to the PRV that contains the specific ground randvar V_s.

Equation (8) makes it apparent that overlaps are excluded as it posits that all groundings of a parent PRV have connections to all groundings of the child PRV. This all-to-all connection does not happen when overlaps occur in the two logvar sequences. With this limitation, answering a query $P(\boldsymbol{Q}|\boldsymbol{E} = \boldsymbol{e})$ is done using lifted operations for matrix multiplication and inversion together with

$$\boldsymbol{\mu}^* = \boldsymbol{\mu}_{\boldsymbol{Q}} + \boldsymbol{\Sigma}_{\boldsymbol{QE}}\boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1}(\boldsymbol{e} - \boldsymbol{\mu}_{\boldsymbol{E}}) \tag{10}$$

$$\boldsymbol{\Sigma}^* = \boldsymbol{\Sigma}_{\boldsymbol{QQ}} - \boldsymbol{\Sigma}_{\boldsymbol{QE}}\boldsymbol{\Sigma}_{\boldsymbol{EE}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{EQ}}. \tag{11}$$

For details on how to come up with the closed form for conditional Gaussian distributions, see, e.g., Eaton [1983].

3 Handling Overlaps in the Lifted Joint

When allowing overlaps between the logvars of a parent and a child PRV, connections from parent to child grounding only exist if the groundings share the same instances of the overlapping logvars. This violates the assumptions about disjoint logvars, causing that Eq. (8) no longer holds. Therefore, we need to specify the general form of the transition matrix $\boldsymbol{T}$ given overlaps and construct a lifted representation of the covariance matrix anew.

The main result of this section is a lifted representation for the values in the covariance matrix in the form of $M \cdot M$ lifted covariance vectors, where each covariance vector $\boldsymbol{\rho}^{(i,j)}$ has a dimensionality of $2^{\boldsymbol{L}_i \cap \boldsymbol{L}_j}$, which does not depend on the domain sizes, together with a lifted construction approach to get the lifted representation. Specifically, this section derives the main result and shows that the lifted representation is an exact alternative representation that exploits the structure imposed by isomorphic randvars. For brevity: If we index the covariance matrix $\boldsymbol{\Sigma}$ with a PRV index, we refer to the block in the covariance matrix belonging to the PRV.

3.1 General Form of the Transition Matrix

The transition matrix $\boldsymbol{T}$ can be structured into $M \times M$ blocks. Each block $\boldsymbol{T}_{i,j}$ describes the relationships between all randvars represented by X_i and all randvars represented by X_j. As the randvars are isomorphic, we can proceed block-wise instead of line-wise in Eqs. (2) to (5). In the non-overlapping case, a block contains either zeroes, namely, if there is no parent-child relationship between X_i and X_j, or the same entries denoted as $\beta_{i,j}$ resulting in an equation $\boldsymbol{T}_{i,j} = \beta_{i,j}\mathbf{J}_{|D(\boldsymbol{L}_i)|\times|D(\boldsymbol{L}_j)|}$ for a block, where $\mathbf{J}$ denotes the all-ones matrix. For a full overlap, each randvar in $gr(X_i)$ has a connection to only one randvar in $gr(X_j)$, namely, where X_i and X_j are grounded with the same instances. Given a global ordering, a full overlap results in a block design of $\boldsymbol{T}_{i,j} = \beta_{i,j}\mathbf{I}_{|D(\boldsymbol{L}_i)|\times|D(\boldsymbol{L}_j)|}$. Given a full overlap, we could add non-overlapping logvars. Adding a non-overlapping additional logvar L_p in the parent sequence implies that previously one randvar is now replaced by $|D(L_p)|$ randvars, which leads to a Kronecker multiplication of a $\mathbf{J}_{|D(L_p)|\times 1}$ column vector (parent variables correspond to rows). Analogously, a non-overlapping additional logvar L_c in the child PRV results in a Kronecker multiplication of a $\mathbf{J}_{1\times|D(L_c)|}$ row vector. Depending on the global ordering of the logvars, we get the following general formula for $\boldsymbol{T}$, with $\mathbf{J}^0 = \mathbf{I}$:

$$\boldsymbol{T}_{i,j} = \beta_{i,j} \bigotimes_{l \in seq(\boldsymbol{L}_i,\boldsymbol{L}_j)} \mathbf{J}_{dim(l,\boldsymbol{L}_i)\times dim(l,\boldsymbol{L}_j)}^{ov(l,\boldsymbol{L}_i,\boldsymbol{L}_j)}. \tag{12}$$

3.2 Constructing the Covariance Matrix

The lifted representation of the transition matrix $\boldsymbol{T}$ allows for calculating a lifted version of the covariance matrix $\boldsymbol{\Sigma}$ as the dimensions of the blocks are determined by the logvars in the same way. The critical steps are Eqs. (3) and (5), which are inductive formulations. Therefore, we use an induction-like approach to show that all blocks follow a fixed structure, which enables a lifted $\boldsymbol{\Sigma}$.

During construction and query answering, we will need to iterate over all possible combinations of $\mathbf{J}$ and $\mathbf{I}$ for a logvar sequence $\boldsymbol{L}$. For ease of notation, we use that $\mathbf{J}^1 = \mathbf{J}$ and $\mathbf{J}^0 = \mathbf{I}$ and iterate over a binary vector $\boldsymbol{q}$ from $\mathbf{0} = 0\ldots0$ (all $\mathbf{I}$) to $\mathbf{1} = 1\ldots1$ (all $\mathbf{J}$) for $\boldsymbol{L}$ to use as exponents to $\mathbf{J}$ as basis. The term $\boldsymbol{Q}$ refers to the set of all possible permutations. Each position in $\boldsymbol{q}$ can be mapped to the corresponding logvars in $\boldsymbol{L}$. A superscript i means that $\boldsymbol{q}^{(i)}$ refers to logvars $\boldsymbol{L}_i$, a superscript (i,j) refers to the logvars in $\boldsymbol{L}_i \cap \boldsymbol{L}_j$. Without an overlap, we set $\boldsymbol{q}$ to $\mathbf{1}$. In slight abuse of notation, we use a projection $\pi_{\boldsymbol{L}}(\boldsymbol{q})$ to mean projecting $\boldsymbol{q}$ onto those entries that are mapped to logvars in $\boldsymbol{L}$. Similarly, a selection $\sigma_{\boldsymbol{q}}(\boldsymbol{Q}')$ yields those vectors $\boldsymbol{q}'$ in $\boldsymbol{Q}'$ where $\boldsymbol{q}'$, projected onto the shared logvars with $\boldsymbol{q}$, has the same values as $\boldsymbol{q}$.

The assumed structure of an on-diagonal block in $\boldsymbol{\Sigma}$, representing the relationship of PRV X_i with itself, is

$$\boldsymbol{\Sigma}_{i,i} = \sum_{\boldsymbol{q}^{(i)}=\mathbf{0}}^{\mathbf{1}} \rho_{\boldsymbol{q}^{(i)}}^{(i)} \bigotimes_{L \in \boldsymbol{L}_i} \mathbf{J}_{|D(L)|\times|D(L)|}^{\pi_L(\boldsymbol{q}^{(i)})}, \tag{13}$$

for which we have to store the $2^{|\boldsymbol{L}_i|}$-dimensional vector $\boldsymbol{\rho}^{(i)}$.

For X_i, X_j, we assume the following off-diagonal structure:

$$\boldsymbol{\Sigma}_{i,j} = \sum_{\boldsymbol{q^{(i,j)}}=\boldsymbol{0}}^{\boldsymbol{1}} \rho^{(i,j)}_{\boldsymbol{q^{(i,j)}}} \bigotimes_{L \in seq(\boldsymbol{L}_i, \boldsymbol{L}_j)} \mathbf{J}^{qexp(\boldsymbol{q}^{(i,j)}, L)}_{dim(L,\boldsymbol{L}_i) \times dim(L,\boldsymbol{L}_j)} \tag{14}$$

with

$$qexp(\boldsymbol{q}^{(i,j)}, L) = \begin{cases} \pi_L(\boldsymbol{q}^{(i,j)}) & \text{if } L \in (\boldsymbol{L}_i \cup \boldsymbol{L}_j) \\ 1 & \text{otherwise.} \end{cases} \tag{15}$$

Equation (13) is a special case of Eq. (14) as both logvar sequences involved are equal, i.e., $\boldsymbol{q^{(i,j)}} = \boldsymbol{q^{(i)}}$, $seq(\boldsymbol{L}_i, \boldsymbol{L}_i) = \boldsymbol{L}_i$, and dim and $qexp$ simplifying to their first cases.

Next, we show that, given the assumed structures, all blocks added to the covariance matrix follow the same format. The formula for a new block is.

$$\boldsymbol{\Sigma}_{i,j} = \sum_{n=1}^{j-1} \boldsymbol{\Sigma}_{i,n} \boldsymbol{T}_{n,j} \tag{16}$$

Inserting Eqs. (12) and (14) into Eq. (16) for one arbitrary summand $n = k$ results in

$$\sum_{\boldsymbol{q}^{(i,k)}=\boldsymbol{0}}^{\boldsymbol{1}} \left(\rho^{(i,k)}_{\boldsymbol{q}^{(i,k)}} \bigotimes_{L \in seq(\boldsymbol{L}_i, \boldsymbol{L}_k)} \mathbf{J}^{qexp(\boldsymbol{q}^{(i,k)}, L)}_{dim(L,\boldsymbol{L}_i) \times dim(L,\boldsymbol{L}_k)} \right) \cdot \left(\beta_{k,j} \bigotimes_{L \in seq(\boldsymbol{L}_k, \boldsymbol{L}_j)} \mathbf{J}^{ov(L,\boldsymbol{L}_k,\boldsymbol{L}_j)}_{dim(L,\boldsymbol{L}_k) \times dim(L,\boldsymbol{L}_j)} \right) \tag{17}$$

For the combinations of L_i, L_k, and L_j, there are seven possible cases (for brevity, we write $\ell = |D(L)|$):

1. $L \in \boldsymbol{L}_j \cap \boldsymbol{L}_k$ and $L \notin \boldsymbol{L}_i$, result: $\mathbf{I}_\ell \mathbf{J}_{\ell \times 1} = \mathbf{J}_{\ell \times 1}$
2. $L \in \boldsymbol{L}_j$ and $L \notin \boldsymbol{L}_k \cup \boldsymbol{L}_i$, result: $\mathbf{J}_{\ell \times 1} \cdot 1 = \mathbf{J}_{\ell \times 1}$
3. $L \in \boldsymbol{L}_k \cap \boldsymbol{L}_i$ and $L \notin \boldsymbol{L}_j$, result: $\mathbf{J}_{1 \times \ell} \mathbf{I}_\ell = \mathbf{J}_{1 \times \ell}$
4. $L \in \boldsymbol{L}_i$ and $L \notin \boldsymbol{L}_k \cup \boldsymbol{L}_j$, result: $1 \cdot \mathbf{J}_{1 \times \ell} = \mathbf{J}_{1 \times \ell}$
5. $L \in \boldsymbol{L}_k$ and $L \notin \boldsymbol{L}_i \cup \boldsymbol{L}_j$, result: $\mathbf{J}_{1 \times \ell} \mathbf{J}_{\ell \times 1} = \ell$
6. $L \in \boldsymbol{L}_j \cap \boldsymbol{L}_k \cap \boldsymbol{L}_i$, result: $\mathbf{I}_\ell \mathbf{I}_\ell = \mathbf{I}_\ell$
7. $L \in \boldsymbol{L}_i \cap \boldsymbol{L}_j$ and $L \notin \boldsymbol{L}_k$, result: $\mathbf{J}_{\ell \times 1} \mathbf{J}_{1 \times \ell} = \mathbf{J}_{\ell \times \ell}$

Cases 1 and 2 result in the same kron factor, independently of L_k. The same holds for Cases 3 and 4. Case 5 requires a $|D(L)|$ factor for all summands, which does not affect the structure. Cases 6 and 7 depend on L_k. Both cases are taken care of in Eq. (14) as whenever $\boldsymbol{L}_i \cap \boldsymbol{L}_j \neq \emptyset$, we keep the corresponding logvars in the summation index. Every entry of the new $\boldsymbol{\rho}^{(i,j)}$ vector can be calculated as follows.

$$\rho^{(i,j)}_{\boldsymbol{q}^{(i,j)}} = \sum_{n=1}^{j-1} \sum_{\boldsymbol{q} \in \sigma_{\boldsymbol{q^{(i,j)}}}(\boldsymbol{Q^{(i,n)}})} \rho^{(i,n)}_{\boldsymbol{q}} \beta_{n,j} \prod_{L \in (\boldsymbol{L}_n \setminus \boldsymbol{L}_j)} |D(L)|^{\pi_L(\boldsymbol{q})} \tag{18}$$

The transpose calculation in Eq. (4) stays in the same structure given by Eq. (14):

$$\begin{aligned} &\boldsymbol{\Sigma}^T_{i,j} \\ &= \left(\sum_{\boldsymbol{q}^{(i,j)}=\boldsymbol{0}}^{\boldsymbol{1}} \rho^{(i,j)}_{\boldsymbol{q}^{(i,j)}} \bigotimes_{L \in seq(\boldsymbol{L}_i, \boldsymbol{L}_j)} \mathbf{J}^{qexp(\boldsymbol{q}^{(i,j)}, L)}_{dim(L,\boldsymbol{L}_i) \times dim(L,\boldsymbol{L}_j)} \right)^T \\ &= \sum_{\boldsymbol{q}^{(i,j)}=\boldsymbol{0}}^{\boldsymbol{1}} \rho^{(i,j)}_{\boldsymbol{q}^{(i,j)}} \bigotimes_{L \in seq(\boldsymbol{L}_i, \boldsymbol{L}_j)} \mathbf{J}^{qexp(\boldsymbol{q}^{(i,j)}, L)}_{dim(L,\boldsymbol{L}_j) \times dim(L,\boldsymbol{L}_i)} \\ &= \boldsymbol{\Sigma}_{j,i} \end{aligned} \tag{19}$$

Based on the formula for calculating a new off-diagonal block, we apply Eq. (5) to calculate a new on-diagonal block. The only difference to the off-diagonal case is that we additionally add the PRV variance λ_{X_j} to the diagonal after calculating the $\boldsymbol{\rho}$ vector using Eq. (18), meaning, we add PRV variance λ_{X_j} to the one summand indexed by $0 \ldots 0$:

$$\rho^{(j,j)}_{0 \ldots 0} \leftarrow \rho^{(j,j)}_{0 \ldots 0} + \lambda_{X_j} \tag{20}$$

The last part of our induction-like proof is to show that our starting point is in line with our assumptions. Based on the definition by Shachter and Kenley [1989], we know that $\boldsymbol{\Sigma}_{11} = \lambda_{X_1} \mathbf{I}$, which is a specific case of the general form from Eq. (2), where only $\rho^{(1)}_{0 \ldots 0}$ has a value unequal to zero.

3.3 Lifted Representation

With Λ as the overall number of logvars, we can record that the lifted representation requires

1. $M \cdot M$ lifted covariance vectors, where each covariance vector $\boldsymbol{\rho}^{(i,j)}$ has a dimensionality of $2^{|\boldsymbol{L}_i \cap \boldsymbol{L}_j|}$,
2. an M-dimensional mean vector $\boldsymbol{\eta}$,
3. a Λ-dimensional cardinality vector $\boldsymbol{\tau}$ for the logvars, and
4. a $M \times \Lambda$-dimensional mapping ψ from PRVs to involved logvars.

Based on Eq. (14), we can convert the lifted representation of the covariance matrix back into a grounded representation. Given the lifted joint of the full joint distribution represented by a parameterized GBN and a fully lifted construction, we now have to specify arithmetic operations for working with the lifted joint, which we need for lifted query answering.

4 Arithmetic Operations in the Lifted Joint

Since handling overlaps results in a more complex lifted representation of the joint distribution, we need new rules for working with this lifted representation to support lifted query answering. Adding or subtracting two blocks with matching dimensions as well as a summation over the same index set is straight-forward, i.e., the corresponding ρ values are added or subtracted. Multiplication and inversion are more complex and we show next how to handle them.

4.1 Multiplication

Multiplying two blocks $\boldsymbol{\Sigma}_{i,j}, \boldsymbol{\Sigma}_{j,k}$ of the same structure of Eq. (14) boils down to a repeated application of Eq. (17) because every summand of $\boldsymbol{\Sigma}_{i,j}$ is multiplied with every summand of $\boldsymbol{\Sigma}_{j,k}$, and every individual summand of $\boldsymbol{\Sigma}_{j,k}$ has the same structure as the $\boldsymbol{T}$ matrix, meaning we can calculate the ρ-value of the resulting matrix in a lifted way as follows:

$$\rho^{(new)}_{\boldsymbol{q}^{(i,k)}} = \sum_{\boldsymbol{q}^{(i,j)}=\boldsymbol{0}}^{\boldsymbol{1}} \sum_{\boldsymbol{q}^{(j,k)}=\boldsymbol{0}}^{\boldsymbol{1}} id(\boldsymbol{q}^{(i,k)}, or(\boldsymbol{q}^{(i,j)}, \boldsymbol{q}^{(j,k)})) \\ \rho^{(i,j)}_{\boldsymbol{q}^{(i,j)}} \rho^{(j,k)}_{\boldsymbol{q}^{(j,k)}} \prod_{L \in \boldsymbol{L}_j} |D(L)|^{lexp(L, \boldsymbol{q}^{(i,j)}, \boldsymbol{q}^{(j,k)})} \tag{21}$$

with $or(\boldsymbol{q}^{(i,j)}, \boldsymbol{q}^{(j,k)})$ as the bitwise or-operation, a function id that returns 1 if the resulting logvar sequence of the or-operation is identical to the values in $\boldsymbol{q}^{(i,k)}$ at the logvar positions referenced in $\boldsymbol{q}^{(i,k)}$, i.e.,

$$id(\boldsymbol{q}^{(i,k)}, \boldsymbol{q}) = \begin{cases} 1 & \text{if } \pi_{\boldsymbol{q}^{(i,k)}}(\boldsymbol{q}) = \boldsymbol{q}^{(i,k)} \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

and a function $lexp$

$$lexp(L, \boldsymbol{q}^{(i,j)}, \boldsymbol{q}^{(j,k)}) = \\ \begin{cases} 1 & \text{if } (L \in \boldsymbol{L}_j \setminus \boldsymbol{L}_i \vee \pi_L(\boldsymbol{q}^{(i,j)}) = \boldsymbol{1}) \\ & \quad \wedge (L \in \boldsymbol{L}_j \setminus \boldsymbol{L}_k \vee \pi_L(\boldsymbol{q}^{(j,k)}) = \boldsymbol{1}) \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

which returns 1 if the referenced logvar L fulfils both parts of a conjunction. The first part asks that L occurs only in $\boldsymbol{L}_j$ and not $\boldsymbol{L}_i$ or that the value of L in $\boldsymbol{q}^{(i,j)}$ is 1, which can only happen if the first disjunct is false. The second part asks the same regarding k instead of i. The returned value as an exponent ensures that the factor of the domain size $|D(L)|$ occurs whenever two $\mathbf{J}$'s meet.

The main idea of Eq. (21) is to take into account all possible combinations of $\mathbf{J}$ and $\mathbf{I}$ matrices that occur in the summations of both blocks. Block matrix equation rules allow us to use Eq. (21) also for multiplying rows or matrices of structured blocks as long as the dimensions are matching.

4.2 Inversion

In the non-overlapping case, Hartwig and Möller [2020] use the recursive block matrix inversion approach by Bernstein [2009], that we transfer to this case where overlaps are allowed:

$$\begin{bmatrix} \tilde{\boldsymbol{A}} & \tilde{\boldsymbol{B}} \\ \tilde{\boldsymbol{C}} & \tilde{\boldsymbol{D}} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{P} \\ \boldsymbol{Q} & \boldsymbol{R} \end{bmatrix} \\ = \begin{bmatrix} \tilde{\boldsymbol{A}}^{-1} + \tilde{\boldsymbol{A}}^{-1} \tilde{\boldsymbol{B}} \tilde{\boldsymbol{F}}^{-1} \tilde{\boldsymbol{C}} \tilde{\boldsymbol{A}}^{-1} & -\tilde{\boldsymbol{A}}^{-1} \tilde{\boldsymbol{B}} \tilde{\boldsymbol{F}}^{-1} \\ -\tilde{\boldsymbol{F}}^{-1} \tilde{\boldsymbol{C}} \tilde{\boldsymbol{A}}^{-1} & \tilde{\boldsymbol{F}}^{-1} \end{bmatrix}, \tag{24}$$

where

$$\tilde{\boldsymbol{F}} = \tilde{\boldsymbol{D}} - \tilde{\boldsymbol{C}} \tilde{\boldsymbol{A}}^{-1} \tilde{\boldsymbol{B}}. \tag{25}$$

This approach needs to be able to perform matrix multiplication, which we have already described and a way to invert blocks in the structure of on-diagonal blocks (see Eq. (13)). For this inversion, we use the lemma by Searle and Henderson [1979], for which they provide a detailed deduction and proof in their work:

Lemma 1. *Given*

$$\boldsymbol{V}_p = \sum_{\boldsymbol{q}=\boldsymbol{0}}^{\boldsymbol{1}} \theta_{\boldsymbol{q}} (\boldsymbol{J}^{q_p}_{n_p} \otimes ... \otimes \boldsymbol{J}^{q_1}_{n_p}), \tag{26}$$

of order $N_p = \prod_{r=1}^{p} n_r$ where q_i refers to the entries in $\boldsymbol{q}$ and n_i to each dimension, the inverse is given by

$$\boldsymbol{V}_p^{-1} = \sum_{\boldsymbol{q}=\boldsymbol{0}}^{\boldsymbol{1}} \kappa_{\boldsymbol{q}} (\mathbf{J}^{q_p}_{n_p} \otimes ... \otimes \mathbf{J}^{q_1}_{n_p}), \tag{27}$$

where

$$\boldsymbol{T}_p = \begin{bmatrix} 1 & 0 \\ 1 & n_p \end{bmatrix} \otimes ... \otimes \begin{bmatrix} 1 & 0 \\ 1 & n_1 \end{bmatrix} \tag{28}$$

and

$$\boldsymbol{\kappa} = \boldsymbol{T}^{-1} \frac{1}{\boldsymbol{T\theta}}. \tag{29}$$

With Lemma 1, lifted subtraction, and lifted matrix multiplication, we can now perform a block matrix inversion to invert the full covariance matrix in a lifted way.

5 Lifted Query Answering

A query $P(\boldsymbol{Q}|\boldsymbol{E} = \boldsymbol{e})$ as given in Definition 5 involves a query set $\boldsymbol{Q}$ and evidence $\boldsymbol{E} = \boldsymbol{e}$. Equations (10) and (11) describe the formulas for determining the posterior distribution $\mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$. The involved matrices $\boldsymbol{\Sigma_{QE}}$, $\boldsymbol{\Sigma_{EE}}$, and $\boldsymbol{\Sigma_{EQ}}$ are sub-matrices of the overarching $\boldsymbol{\Sigma}$ matrix. Not all sub-matrices follow the structure of Eq. (14). The structure occurs if all ground observations $\boldsymbol{E}$ and query randvars $\boldsymbol{Q}$ refer to the same set of logvar instances for all shared logvars. We call the query to be *liftable* if $\boldsymbol{E}$ and $\boldsymbol{Q}$ fulfill this condition. Then, $\boldsymbol{\rho_{lif(E)}}$ contains only a subset of ρ vectors and the cardinality for all operations is reduced to the logvar cardinality occurring in the query. Using the lifted representation combined with the lifted operations for multiplication and inversion described in Section 4, a lifted representation $\boldsymbol{\rho}^*$ for the matrix $\boldsymbol{\Sigma}^*$ can be generated. To get the grounded query answer $\boldsymbol{\Sigma}^*$, $\boldsymbol{\rho}^*$ can simply be grounded. The $\boldsymbol{\mu}^*$ needs to be calculated grounded because it involves grounded evidence. We can ground $\boldsymbol{\Sigma_{QE}} \boldsymbol{\Sigma_{EE}}^{-1}$ and multiply it with the vector $\boldsymbol{e} - \boldsymbol{\mu_E}$ to get $\boldsymbol{\mu}^*$.

6 Complexity

We discuss the runtime complexity of constructing the lifted joint and using it for query answering. As stated above we have N randvars, combined into M PRVs, using Λ logvars. We use S to denote the longest occurring logvar sequence.

6.1 Constructing the Lifted Joint

For constructing the ground level covariance matrix in the joint distribution, we have an upper bound of $O(N^3)$. Our lifted approach is fully independent of N, because we directly iterate over M PRVs. Allowing for overlaps results in iterations over binary vectors. We need to potentially iterate over $O(2^\Lambda)$ permutations, resulting in an overall runtime complexity of $O(M^3 2^\Lambda)$. Since $M \ll N$ and $2^\Lambda \ll N$ with decent domain sizes, it is a significant improvement.

6.2 Lifted Query Answering

We denote the number of queried randvars as N_Q and the number of evidence randvars as N_E. Furthermore, we refer to the number of PRVs involved in the N_Q and N_E randvars using M_Q and M_E, respectively.

Matrix multiplication and inversion drive the complexity of query answering. Although there are matrix inversion and multiplication algorithms that have a runtime complexity of less than $O(N^3)$, for the sake of simplicity, we take $O(N^3)$ as an upper bound [Le Gall, 2014], which results in a runtime complexity of $O(N_E^3 + N_E^2 N_Q + N_E N_Q^2)$. Depending on the set sizes, one of the terms dominates the other.

In the lifted case, multiplying two blocks of size 2^S has a complexity of $O(2^{2S})$ because in the worst case we need to iterate fully through all possible $\mathbf{I}$, $\mathbf{J}$ combinations. Thus, the matrix multiplications operations have complexity of $O(M_E^2 M_Q 2^{2S})$ and $O(M_E M_Q^2 2^{2S})$. The inversion of one individual block on the diagonal of the covariance matrix involves the multiplication of two matrices, both with at most $2^S \times 2^S$ dimensions, resulting in a complexity of $O(2^{3S})$. In the block matrix inversion, we have at most M_E recursive function calls and Kronecker component block multiplications of at most M_E blocks, resulting in a combined inversion complexity of $O(M_E 2^{3S} + M_E^2 2^S)$.

To calculate the mean vector, we still have ground operations. However, the dependency on both N_Q and N_E is only linear: Grounding the result of $\boldsymbol{\Sigma_{QE}} \boldsymbol{\Sigma_{EE}^{-1}}$ is in $O(N_Q N_E)$ and multiplying the result with the ground level evidence vector $\boldsymbol{e} - \boldsymbol{\mu_E}$ is in $O(N_Q N_E)$, as well.

Overall, this results in a complexity of $O(M_E^2 M_Q 2^{2S} + M_E M_Q^2 2^{2S} + M_E 2^{3S} + N_Q N_E)$. Unfortunately, the last term means that query answering is not fully lifted. Compared to the non-overlapping case, N_Q now also appears as a ground term. We cannot really avoid N_E because to incorporate ground evidence, we always need to at least visit each evidence randvar once. The dependency on N_Q might be a focus for further research, though.

7 Experimental Evaluation

We demonstrate the complexity results by evaluating constructing the lifted joint and answering queries.

We set up an experiment with 4 PRVs and 5 logvars that are partially shared. We increase the domain sizes of 2 logvars in exponential steps (from 2 to 2^7) resulting in around 20,000 randvars in the grounded model for the largest domains. For conditional query answering, we introduce evidence for the majority of the randvars and use a query set of 4 randvars.

Figures 1 and 2 show the results. For both benchmarks, the lifted version is significantly faster. Similar to the results by Hartwig and Möller [2020], construction is independent of the domain sizes of the logvars (i.e., the number of grounded randvars) and query answering shows some dependence on the number of evidence randvars. The results also show that working with the more complex structure of representing blocks in the covariance matrix with $\boldsymbol{\rho}$ comes with additional overhead recognizable in the low cardinality cases.

In summary, the runtimes reflect that models with overlaps in the logvar sequences make constructing and query answering more complex compared to models without overlaps. However, applicability of the approach increases immensely. Additionally, compared to the ground case, the lifted version gains dramatically in terms of runtime.

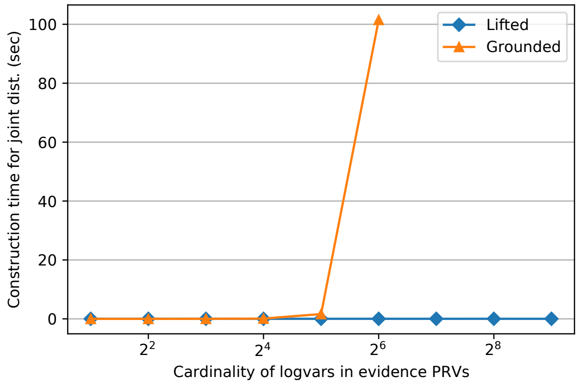

Figure 1: Runtimes for constructing the lifted joint

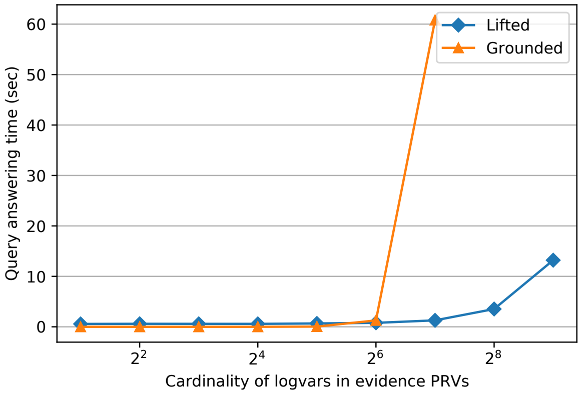

Figure 2: Runtimes for query answering

8 Conclusion and Outlook

We present a generalized approach to lifting GBNs, consisting of constructing a lifted joint and lifted query answering using the lifted joint. Whereas previous work lifts the inductive construction approach of the full joint of a GBN only for parameterized GBNs that do not have overlaps in the logvars, the presented approach works for parameterized GBNs where no restrictions are placed on the logvars, which goes beyond liftability results of discrete models, is however tied to the continuous setting. Allowing overlaps, we significantly extend the applicability of lifting in GBNs. We show that constructing a lifted joint and answering queries on the lifted joint leads to savings in terms of complexity. The experimental results show that this theoretical speed-up can also accomplished in an implementation with impressive results.

For future work, we look into lifting evidence handling as well as query answering with parameterized queries as done for discrete parameterized models [Taghipour *et al.*, 2013; Braun and Möller, 2018] targeting the $O(N_E N_Q)$ term. Another avenue concerns hybrid models where continuous and discrete randvars simultaneously occur in the model.

References

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

[Blei *et al.*, 2017] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

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

[Braun and Möller, 2018] Tanya Braun and Ralf Möller. Parameterised Queries and Lifted Query Answering. In *IJCAI-18 Proc. of the 27th International Joint Conference on AI*, pages 4980–4986. IJCAI Organization, 2018.

[Cano *et al.*, 2004] Rafael Cano, Carmen Sordo, and José M Gutiérrez. Applications of Bayesian networks in meteorology. In *Advances in Bayesian networks*, pages 309–328. Springer, 2004.

[Choi *et al.*, 2010] Jaesik Choi, David J. Hill, and Eyal Amir. Lifted inference for relational continuous models. In *Workshops at the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

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

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

[Grzegorczyk, 2010] Marco Grzegorczyk. An introduction to Gaussian Bayesian networks. In *Systems Biology in Drug Discovery and Development*, pages 121–147. Springer, 2010.

[Hartwig and Möller, 2020] Mattis Hartwig and Ralf Möller. Lifted query answering in gaussian bayesian networks. In *Proceedings of Machine Learning Research*, 2020.

[Holtzen *et al.*, 2019] Steven Holtzen, Todd Millstein, and Guy Van den Broeck. Generating and Sampling Orbits for Lifted Probabilistic Inference. *arXiv preprint arXiv:1903.04672*, 2019.

[Huang *et al.*, 2011] Shuai Huang, Jing Li, Jieping Ye, Adam Fleisher, Kewei Chen, Teresa Wu, and Eric Reiman. Brain effective connectivity modeling for Alzheimer's disease by sparse Gaussian bayesian network. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 931–939, 2011.

[Kimmig *et al.*, 2015] Angelika Kimmig, Lilyana Mihalkova, and Lise Getoor. Lifted graphical models: a survey. *Machine Learning*, 99(1), 2015.

[Koller *et al.*, 2009] Daphne Koller, Nir Friedman, and Francis Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[Lauritzen and Spiegelhalter, 1988] Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):157–194, 1988.

[Le Gall, 2014] François 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.

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

[Salmerón *et al.*, 2018] Antonio Salmerón, Rafael Rumí, Helge Langseth, Thomas Nielsen, and Anders Madsen. A Review of Inference Algorithms for Hybrid Bayesian Networks. *Journal of Artificial Intelligence Research*, 62:799–828, 2018.

[Searle and Henderson, 1979] Shayle R Searle and Harold Y Henderson. Dispersion matrices for variance components models. *Journal of the American Statistical Association*, 74(366a):465–470, 1979.

[Shachter and Kenley, 1989] Ross D Shachter and C Robert Kenley. Gaussian Influence Diagrams. *Management Science*, 35(5):527–550, 1989.

[Sharma *et al.*, 2018] Vishal Sharma, Noman Ahmed Sheikh, Happy Mittal, Vibhav Gogate, and Parag Singla. Lifted marginal MAP inference. *arXiv preprint arXiv:1807.00589*, 2018.

[Taghipour *et al.*, 2013] 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.

[Zhang and Poole, 1996] Nevin Lianwen Zhang and David Poole. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, 1996.