# Modelling and Inference with Conditional Gaussian Probabilistic Decision Graphs[☆]

Jens D. Nielsen[a,*], José A. Gámez[a], Antonio Salmerón[b]

[a] *Albacete Research Institute on Informatics, University of Castilla-La Mancha, Campus Universitario s/n, 02071 Albacete, Spain*
*{dalgaard,jose.gamez}@dsi.uclm.es*

[b] *Dept. Statistics and Applied Mathematics, University of Almería, La Cañada de San Urbano s/n, 04120 Almería, Spain*
*antonio.salmeron@ual.es*

## Abstract

Probabilistic decision graphs (PDGs) are probabilistic graphical models that represent a factorisation of a discrete joint probability distribution using a "decision graph"-like structure over local marginal parameters. The structure of a PDG enables the model to capture some context specific independence relations that are not representable in the structure of more commonly used graphical models such as Bayesian networks and Markov networks. This sometimes makes operations in PDGs more efficient than in alternative models. PDGs have previously been defined only in the discrete case, assuming a multinomial joint distribution over the variables in the model. We extend PDGs to incorporate continuous variables, by assuming a Conditional Gaussian (CG) joint distribution. We also show how inference can be carried out in an efficient way.

*Keywords:* Probabilistic decision graphs, Conditional Gaussian distribution, Hybrid Graphical Models, Inference

## 1. Introduction

The Probabilistic Decision Graph (PDG) model was introduced in [2] as an efficient representation of probabilistic transition systems. In this study, we consider the more general version of PDGs proposed in [3].

PDGs are probabilistic graphical models that can represent some context specific independencies that are not efficiently captured by conventional graphical models, such as Markov Network or Bayesian Network (BN) models. Furthermore, probabilistic inference can be carried out directly in the PDG structure and has a time complexity linear in the size of the PDG model.

[☆] This work has been supported by the Spanish Ministry of Science and Innovation, through projects TIN2007-67418-C03-01,02, TIN2010-20900-C04-02,03 and by EFRD (FEDER) funds. A preliminary version of this paper was presented in the FLAIRS-23 Conference [1].

[*] Corresponding author

So far, PDGs have only been studied as representations of joint distributions over *discrete* random variables, showing a competitive performance when compared to BN or Latent class Naïve BN estimation models [4]. The PDG model has also been successfully applied to supervised classification problems [5] and unsupervised clustering [6].

However, it is common in practice to find problems where discrete and continuous variables coexist. This fact has motivated the development of graphical models, mainly hybrid Bayesian networks, oriented to handle discrete and continuous variables simultaneously [7, 8, 9, 10, 11].

In this paper, we introduce an extension of PDG models that incorporates *continuous* variables, and therefore expands the class of problems that can be handled by these models. More precisely, we define a new class of PDG models, called *conditional Gaussian PDGs* and show how they represent a joint distribution over a set of discrete and continuous variables, of class conditional Gaussian. We also show how probabilistic inference can be carried out over this new structure, taking advantage of the efficiency already shown for discrete PDGs.

## 2. The Conditional Gaussian model

We will use uppercase letters to denote random variables, and boldfaced uppercase letters to denote random vectors, e.g. $\mathbf{X} = \{X_0, X_1, \ldots, X_N\}$. By $R(X)$ we denote the set of possible states of variable $X$, and similarly for random vectors, $R(\mathbf{X}) = \times_{X_i \in \mathbf{X}} R(X_i)$. By lowercase letters $x$ (or $\mathbf{x}$) we denote some element of $R(X)$ (or $R(\mathbf{X})$). When $\mathbf{x} \in R(\mathbf{X})$ and $\mathbf{Y} \subseteq \mathbf{X}$, we denote by $\mathbf{x}[\mathbf{Y}]$ the projection of $\mathbf{x}$ onto coordinates $\mathbf{Y}$. Throughout this document we will consider a set $\mathbf{W}$ of discrete variables and a set $\mathbf{Z}$ of continuous variables, and we will use $\mathbf{X} = \mathbf{W} \cup \mathbf{Z}$.

The Conditional Gaussian (CG) model [12, 13] allows a factorised representation of a joint probability distribution over discrete and continuous variables, and that factorisation can be encoded by a Bayesian network with the restriction that discrete variables are not allowed to have continuous parents.

In the CG model, the conditional distribution of each discrete variable $W \in \mathbf{W}$ given their parents is a multinomial, whilst the conditional distribution of each continuous variable $Z \in \mathbf{Z}$ with discrete parents $\mathbf{E} \subseteq \mathbf{W}$ and continuous parents $\mathbf{C} \subseteq \mathbf{Z}$, is given by

$$f(z|\mathbf{E} = \mathbf{e}, \mathbf{C} = \mathbf{c}) = \mathcal{N}(z; \alpha(\mathbf{e}) + \boldsymbol{\beta}(\mathbf{e})^{\mathsf{T}}\mathbf{c}, \sigma^2(\mathbf{e}))\,, \tag{1}$$

for all $\mathbf{e} \in R(\mathbf{E})$ and $\mathbf{c} \in R(\mathbf{C})$, where $\alpha$ and $\boldsymbol{\beta}$ are the coefficients of a linear regression model of $Z$ given its continuous parents which could be a different model for each configuration of the discrete variables $\mathbf{E}$.

In the CG model, after fixing any configuration of the discrete variables, the joint distribution of any subset $\mathbf{C} \subseteq \mathbf{Z}$ of continuous variables is a multivariate Gaussian. In the following we will show how the parameters of the multivariate Gaussian can be obtained from the ones in the CG representation. To this end, consider a set of $n$ continuous variables $Z_1, \ldots, Z_n$ with a conditionally specified joint density

$$f(z_1, \ldots, z_n) = \prod_{i=1}^{n} f(z_i | z_{i+1}, \ldots, z_n) \,, \tag{2}$$

where the $k$-th factor, $1 \leq k \leq n$, is such that

$$f(z_k | z_{k+1}, \ldots, z_n) = \mathcal{N}(z_k; \mu_{z_k|z_{k+1},\ldots,z_n}, \sigma^2_{z_k|z_{k+1},\ldots,z_n}) \,,$$

and therefore it holds that the joint is

$$f(z_1, \ldots, z_n) = \mathcal{N}(z_1, \ldots, z_n; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \,,$$

where $\boldsymbol{\mu}$ is the n-dimensional vector of means and $\boldsymbol{\Sigma}$ is the covariance matrix of the multivariate distribution over random variables $Z_1, \ldots, Z_n$. We will use the notation $\boldsymbol{\mu}_{z_i}$ and $\boldsymbol{\Sigma}_{z_i,z_j}$ to index $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Please note that the $\boldsymbol{\Sigma}_{z_i,z_i}$ contains the marginal variance $\sigma^2_{z_i}$, and $\boldsymbol{\Sigma}_{z_i,z_j}$ contains the covariance of $Z_i$ and $Z_j$, also denoted as $\sigma_{z_i,z_j}$. The conversion between the parameters of the joint and the conditional specification is established as follows (see for instance [14, Theorems 7.3 and 7.4]). According to Eq. (1), the conditional mean $\mu_{z_k|z_{k+1},\ldots,z_n}$ is a linear regression model over $Z_{k+1}, \ldots, Z_n$. If we write that regression model as

$$\mu_{z_k|z_{k+1},\ldots,z_n} \quad = \quad \alpha_k + \boldsymbol{\beta}^{\mathsf{T}}_{(+k)} \mathbf{z}_{(+k)} \,, \tag{3}$$

where $\mathbf{z}^{\mathsf{T}}_{(+k)} = (z_{k+1}, \ldots, z_n)$, $\boldsymbol{\beta}^{\mathsf{T}}_{(+k)} = (\beta^k_{k+1}, \ldots, \beta^k_n)$, it is known that the regression coefficients verify that

$$\boldsymbol{\Sigma}_{z_k,z_i} = \sum_{j=k+1}^{n} \beta^k_j \boldsymbol{\Sigma}_{z_i,z_j}, \quad i = k+1, \ldots, n, \tag{4}$$

and

$$\alpha_k = \boldsymbol{\mu}_{z_k} - \boldsymbol{\beta}^{\mathsf{T}}_{(+k)} \boldsymbol{\mu}_{(+k)}, \tag{5}$$

where $\boldsymbol{\mu}^{\mathsf{T}}_{(+k)} = (\boldsymbol{\mu}_{z_{k+1}}, \ldots, \boldsymbol{\mu}_{z_n})$.

The conditional variance can be obtained using the law of total variance, which states that for any random variable $Z$ and random vector $\mathbf{U}$, it holds that

$$\mathrm{Var}(Z) = E[\mathrm{Var}(Z|\mathbf{U})] + \mathrm{Var}(E[Z|\mathbf{U}]).$$

In this context, it means that

$$\begin{aligned} \sigma^2_{z_k} &= E[\sigma^2_{z_k|z_{k+1},\ldots,z_n}] + \mathrm{Var}(\mu_{z_k|z_{k+1},\ldots,z_n}) \\ &= \sigma^2_{z_k|z_{k+1},\ldots,z_n} + \mathrm{Var}(\alpha_k + \boldsymbol{\beta}^{\mathsf{T}}_{(+k)} \mathbf{z}_{(+k)}). \end{aligned}$$

Hence,

$$\sigma^2_{z_k|z_{k+1},\ldots,z_n} \quad = \quad \sigma^2_{z_k} - \boldsymbol{\beta}^{\mathsf{T}}_{(+k)} \boldsymbol{\Sigma}_{(+k)} \boldsymbol{\beta}_{(+k)} \,, \tag{6}$$

where $\boldsymbol{\Sigma}_{(+k)}$ is the projection of $\boldsymbol{\Sigma}$ to variables $Z_{k+1}, \ldots, Z_n$.

## 3. Discrete PDGs with Multinomial distribution

We need to introduce some notation before we define the PDG model. Let $G$ be a directed graph over nodes $\mathbf{V}$[1]. Let $\nu \in \mathbf{V}$, we then denote by $pa_G(\nu)$ the set of parents of node $\nu$ in $G$, by $ch_G(\nu)$ the set of children of $\nu$ in $G$, by $de_G(\nu)$ the set of descendants of $\nu$ in $G$, that is recursively defined as $de_G(\nu) = \{\nu' : \nu' \in ch_G(\nu) \vee [\nu' \in ch_G(\nu'') \wedge \nu'' \in de_G(\nu)]\}$, and we use as shorthand notation $de^*_G(\nu) = de_G(\nu) \cup \nu$. By $an_G(\nu)$ we understand the set of ancestors (or predecessors) of $\nu$ in $G$, that is recursively defined as $an_G(\nu) = \{\nu' : \nu' \in pa_G(\nu) \vee [\nu' \in pa_G(\nu'') \wedge \nu'' \in an_G(\nu)]\}$.

The PDG model was introduced in [3] as a probabilistic graphical model of joint distributions over discrete variables. The structure is formally defined as follows:

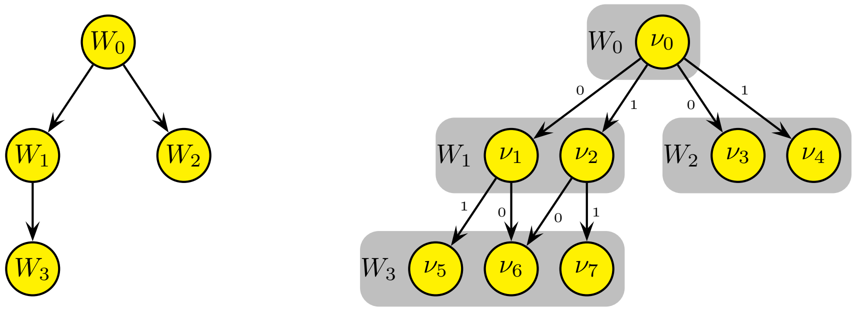
**Definition 1 (The PDG Structure [3]).** *Let $F$ be a forest of directed tree structures over a set of discrete random variables $\mathbf{W}$. A* PDG structure $G = \langle \mathbf{V}, \mathbf{E} \rangle$ *for $\mathbf{W}$ w.r.t. $F$ is a set of* rooted *DAGs, such that:*

1. *Each node $\nu \in \mathbf{V}$ is labelled with exactly one $W \in \mathbf{W}$. By $\mathbf{V}_W$, we will refer to the set of all nodes in a PDG structure labelled with the same variable $W$. For every variable $W$, $\mathbf{V}_W \neq \emptyset$, and we will say that $\nu$ represents $W$ when $\nu \in \mathbf{V}_W$.*
2. *For each node $\nu \in \mathbf{V}_W$, each possible state $w \in R(W)$ and each successor $Y \in ch_F(W)$ there exists exactly* one *edge labelled with $w$ from $\nu$ to some node $\nu'$ representing $Y$. Let $U \in ch_F(W)$, $\nu \in \mathbf{V}_W$ and $w \in R(W)$. By $succ(\nu, U, w)$ we will then refer to the unique node $\nu' \in \mathbf{V}_U$ that is reached from $\nu$ by an edge with label $w$.*

An example of a PDG structure and its corresponding variable forest can be found in Fig. 1(b) and (a) respectively. We will not usually depict the variable forest explicitly as it is included in the PDG structure by the labelling of the nodes, that is, each variable is represented by a specific set of nodes[2]. A PDG structure (e.g. Fig. 1(b)) will then be viewed as a two-layer structure with a variable layer and a node layer. On the variable layer, we have a directed forest structure over the variables $F$ and on the node layer we have a uniquely rooted directed acyclic graph structure. When referring to children, parents, descendants or ancestors of a variable in a PDG structure $G$, we silently refer to the structure $F$. So, using Fig. 1(b) as an example, on the variable layer we have: $pa_G(W_1) = \{W_0\}$, $ch_G(W_0) = \{W_1, W_2\}$, $de_G(W_0) = \{W_1, W_2, W_3\}$ and $an_G(W_3) = \{W_1, W_0\}$. On the node layer, we have: $succ(\nu_0, W_1, 0) = \nu_1$, $succ(\nu_0, W_2, 1) = \nu_3$ and $succ(\nu_1, W_3, 0) = \nu_6$.

[1] We realize that this abuses notation as now $G$ and $\mathbf{V}$ are not a random variable and a random vector, as in Sec. 2. However, the semantics that applies will be clear from context.

[2] Note that each variable can have more than one node representing it.



(a)

(b)

Figure 1: (a) A forest structure $F$ (containing a single tree) over variables $\mathbf{W} = \{W_0, W_1, W_2, W_3\}$. (b) A PDG structure over $\mathbf{W}$ that is consistent with forest $F$.

A PDG structure is instantiated by assigning a real function $f^\nu$ to every node $\nu$ in the structure. The function must have the signature $f^\nu : R(W_i) \rightarrow \mathbb{R}_0^+$, where $\nu \in \mathbf{V}_{W_i}$.

An instantiated PDG structure $G$ over the discrete variables $\mathbf{W}$ is called a Real Function Graph (RFG). It defines the (global) real function $f_G$ with the signature $f_G : R(\mathbf{W}) \rightarrow \mathbb{R}_0^+$, by the following recursive definition:

**Definition 2.** *Let $G$ be an RFG over discrete variables $\mathbf{W}$, and let $\nu \in \mathbf{V}_W$. We then define the local recursive functions:*

$$f_G^\nu(\mathbf{w}) := f^\nu(\mathbf{w}[W]) \prod_{Y \in ch_F(W)} f_G^{succ(\nu, Y, \mathbf{w}[W])}(\mathbf{w}), \tag{7}$$

*for all $\mathbf{w} \in R(\mathbf{W})$. $f_G$ is then defined on $R(\mathbf{W})$ as:*

$$f_G(\mathbf{w}) := \prod_{\nu : \nu \textit{ is a root}} f_G^\nu(\mathbf{w}). \tag{8}$$

The recursive function of Eq. (7) defines a factorisation that includes exactly one factor $f^\nu$ for each $W \in \mathbf{W}$. It will sometimes be convenient to be able to directly refer to the factor that is associated with a given element $\mathbf{w} \in R(\mathbf{W})$. The function *reach* defines exactly this association.

**Definition 3 (Reach).** *A node $\nu$ representing variable $W_i$ in $G$ is* reached *by $\mathbf{w} \in R(\mathbf{W})$ if*

1. *$\nu$ is a root in $G$, or*
2. *$W_j = \mathrm{pa}_F(W_i)$, node $\nu'$ representing variable $W_j$ is* reached *by $\mathbf{w}$ and $\nu = \mathrm{succ}(\nu', W_i, \mathbf{w}[W_j])$.*

*By $reach_G(W_i, \mathbf{w})$ we denote the unique node representing $W_i$ reached by $\mathbf{w}$ in PDG structure $G$.*

As an example, consider again the PDG structure of Fig. 1(b) and let $\mathbf{w} = W_0 = 0, W_1 = 1, W_2 = 1, W_3 = 1$. Then $\mathit{reach}_G(W_0, \mathbf{w}) = \nu_0$, $\mathit{reach}_G(W_1, \mathbf{w}) = \nu_1$, $\mathit{reach}_G(W_2, \mathbf{w}) = \nu_3$ and $\mathit{reach}_G(W_3, \mathbf{w}) = \nu_5$. It should be clear that each node divides the space $R(\mathbf{W})$ into two disjoint sets, the instances that reach the node and those that do not. E.g. $\nu_0$ is reached by all instances in $R(\mathbf{W})$ while $\nu_1$ is reached by all instances $\mathbf{w}$ for which $\mathbf{w}[W_0] = 0$.

Using Def. 3, we can give an alternative definition of $f_G$:

$$f_G(\mathbf{w}) := \prod_{W_i \in \mathbf{W}} f^{\mathit{reach}_G(W_i, \mathbf{w})}(\mathbf{w}[W_i])\,. \tag{9}$$

When all the local functions $f^\nu$ in an RFG $G$ over $\mathbf{W}$ define probability distributions, the function $f_G$ (Def. 2) defines a joint multinomial probability distribution over $\mathbf{W}$ (see [3]). In fact, $f_G^\nu$ in Eq. (7) defines a multinomial distribution over variables $W \cup de_F^*(W)$. We will refer to such RFGs as PDG models.

**Definition 4 (The PDG model [3]).** *A PDG model $\mathcal{G}$ is a pair $\mathcal{G} = \langle G, \theta \rangle$, where $G = \langle \mathbf{V}, \mathbf{E} \rangle$ is a valid PDG structure (Def. 1) over some set $\mathbf{W}$ of discrete random variables and $\theta = \{f^\nu : \nu \in \mathbf{V}\}$ is a set of real functions, each of which defines a discrete probability distribution.*

**Example 1.** *Consider the PDG structure in Fig. 1. It encodes a factorisation of the joint distribution of $\mathbf{W} = \{W_0, W_1, W_2, W_3\}$, with*

$$\begin{array}{ll}
f^{\nu_0} = P(W_0), & f^{\nu_4} = P(W_2|W_0 = 1), \\
f^{\nu_1} = P(W_1|W_0 = 0), & f^{\nu_5} = P(W_3|W_0 = 0, W_1 = 1), \\
f^{\nu_2} = P(W_1|W_0 = 1), & f^{\nu_6} = P(W_3|W_1 = 0, \{W_0 = 0 \vee W_0 = 1\}), \\
f^{\nu_3} = P(W_2|W_0 = 0), & f^{\nu_7} = P(W_3|W_0 = 1, W_1 = 1).
\end{array}$$

*The PDG structure plus the set of conditional distributions given above constitute a PDG model over the set of variables $\mathbf{W} = \{W_0, W_1, W_2, W_3\}$. Assume that we want to evaluate the PDG model for a given configuration of $\mathbf{W}$, for instance, $(0, 1, 1, 1)$. According to Def. 2, the returned value is*

$$\begin{array}{rcl}
f_G(0,1,1,1) & = & f^{\nu_0}(0) f^{\nu_1}(1) f^{\nu_3}(1) f^{\nu_5}(1) \\
& = & P(W_0 = 0) P(W_1 = 1|W_0 = 0) P(W_2 = 1|W_0 = 0) \\
& & P(W_3 = 1|W_0 = 0, W_1 = 1).
\end{array}$$

*Note that the node reached for variable $W_3$ is uniquely defined as $\nu_6$ for all configurations where $W_1 = 0$ (i.e. $\mathbf{w}[W_1] = 0 \Leftrightarrow \mathit{reach}_G(W_3, \mathbf{w}) = \nu_6$), while for $W_1 = 1$ the node reached varies between $\nu_5$ and $\nu_7$ depending on the value of $W_0$. This indicates the existence of context specific independence. More precisely, the conditional distribution of $W_3$ given $W_0$ and $W_1$ is the same regardless of the value of $W_0$ whenever $W_1$ equals 0 (i.e. $P(W_3|W_0 = 1, W_1 = 0) = P(W_3|W_0 = 0, W_1 = 0)$) so in the context of $W_1 = 0$, $W_3$ and $W_0$ are independent.*

## 4. Conditional Gaussian PDGs

In this section we introduce an extension of the multinomial PDG model defined in the previous section. The extension incorporates continuous variables in the model, and we will show afterwards that the factorisation now induces a conditional Gaussian probability distribution. We first define the structural extension.

**Definition 5 (CG-PDG structure).** *Let $F$ be a forest of directed tree structures over a mixed set of discrete and continuous random variables $\mathbf{X} = \mathbf{W} \cup \mathbf{Z}$, where **no** continuous variable $Z \in \mathbf{Z}$ has a discrete variable $W \in \mathbf{W}$ as a child. A* CG-PDG-*structure $G = \langle \mathbf{V}, \mathbf{E} \rangle$ for $\mathbf{X}$ w.r.t. $F$ is then defined exactly as the PDG structure of Def. 1 where:*

1. *each continuous variable is viewed as a single state variable, and*
2. *for each node $\nu$ representing a continuous variable $Z$ and for each variable $Z_c \in ch_F(Z)$, exactly* one *unique child $\nu' \in \mathbf{V}_{Z_c}$ exists, and $\nu'$ has no other parents than $\nu$.*

An example of a CG-PDG structure is displayed in Fig 2. Please note that in a CG-PDG structure, for all $\nu$'s representing some $Z \in \mathbf{Z}$ where $Z_i \in ch_F(Z)$, the set $succ(\nu, Z_i, z)$ is the same regardless of the value $z$. We can therefore leave out the $z$ argument and unambiguously write $succ(\nu, Z_i)$. Moreover, we have that $reach_G(Z, \mathbf{x}) = reach_G(Z, \mathbf{x}[\mathbf{W}])$ for any $X \in \mathbf{X}$, $\mathbf{x} \in R(\mathbf{X})$ and $Z \in \mathbf{Z}$. In fact, for each joint configuration $\mathbf{w}'$ of the discrete predecessors $\mathbf{W}'$ of a continuous variable $Z$ (that is $\mathbf{W}' = an_G(Z) \cap \mathbf{W}$) one unique node representing $Z$ is reached.

**Definition 6 (CG-PDG model).** *A Conditional Gaussian PDG (CG-PDG) model $\mathcal{G}$ over random variables $\mathbf{X} = \mathbf{W} \cup \mathbf{Z}$ is a pair $\mathcal{G} = \langle G, \theta \rangle$, where $G = \langle \mathbf{V}, \mathbf{E} \rangle$ is a CG-PDG structure as defined in Def. 5, and $\theta = \{f^\nu : \nu \in \mathbf{V}\}$ is a set of real functions, and depending on the variable that $\nu$ represents, $f^\nu$ is defined by one of the following cases.*

- *If $\nu$ represents discrete variable $W$, $f^\nu$ defines a multinomial probability distribution over $X$.*
- *If $\nu$ represents continuous variable $Z$ for which $an_G(Z) \cap \mathbf{Z} = \mathbf{U}$ and $an_G(Z) \cap \mathbf{W} = \mathbf{Y}$, then $f^\nu(z, \mathbf{u}) = f(z|\mathbf{u}, \mathbf{y}) = \mathcal{N}(z; \alpha_\nu + \boldsymbol{\beta}_\nu^\mathsf{T}\mathbf{u}, \sigma_\nu^2)$ where $\mathbf{u} \in R(\mathbf{U})$ and $\nu = reach_G(Z, \mathbf{y})$. So $f^\nu$ defines a Gaussian density with conditional mean $\mu_{z|\mathbf{u}} = \alpha_\nu + \boldsymbol{\beta}_\nu^\mathsf{T}\mathbf{u}$ and conditional variance $\sigma_{z|\mathbf{u}}^2 = \sigma_\nu^2$, where $\boldsymbol{\beta}_\nu$ is a vector of $|\mathbf{U}|$ real values and $|\mathbf{U}|$ denotes the cardinal of $\mathbf{U}$.*

In order to simplify the notation, when referring to a function stored in a node $\nu$ corresponding to a continuous variable $Z$, we just write $f^\nu(z)$, even though that function actually depends on the predecessors of $Z$ in the structure.

Before going further, we will give an example of how a CG-PDG model naturally captures the structure of a problem domain with discrete and continuous variables.

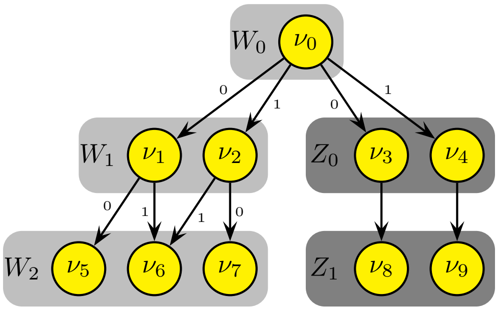


Figure 2: Structure of a CG-PDG with three discrete and two continuous variables.

**Example 2.** *A newspaper delivery van has two possible delivery routes, one of them covering only city A and the other covering city B as well. A 70% of the days, the selected route is the one including only city A. Let us denote by $W_0$ the delivery route $(0 = A, 1 = AB)$. Cities A and B are connected by a pay motorway, with a toll fee of 3 Euro. City B is known to be a busy city traffic much more dense than A, so that the probability of suffering a traffic jam (denoted as $W_1$, with values 0=no and 1=yes) when the selected route includes B is* 0.05*, and* 0.01 *otherwise. If the van suffers a traffic jam, the probability of completing the delivery on time ($W_2$, with values 0=no, 1=yes) is only* 0.5 *regardless of the selected route. If there are no traffic jams, the probability of completing the job on time is* 0.95 *for route A and* 0.8 *for route AB. The cost of the delivery ($Z_1$) depends on the selected route and on the gas consumption ($Z_0$). The gas consumption follows a Gaussian distribution with mean equal to 5 liters and variance of 1 liter$^2$ for route A, whilst the mean is 10 and the variance 1.2 for the other route. The cost also follows a Gaussian distribution, with mean equal to 1.1 times the consumed liters and variance 0.5 when the route is A, and if the route is AB, the mean is increased by the toll fee. The structure in Fig. 2 represents the dependence structure described in this example. A parametrisation of that structure, according to definition 6 and the information given above is as follows: $f^{\nu_0} = P(W_0) = (0.7, 0.3)$, $f^{\nu_1} = P(W_1|W_0 = 0) = (0.99, 0.01)$, $f^{\nu_2} = P(W_1|W_0 = 1) = (0.95, 0.05)$, $f^{\nu_3} = f(z_0|W_0 = 0) = \mathcal{N}(z_0; 5, 1^2)$, $f^{\nu_4} = f(z_0|W_0 = 1) = \mathcal{N}(z_0; 10, 1.2^2)$, $f^{\nu_5} = P(W_2|W_0 = 0, W_1 = 0) = (0.05, 0.95)$, $f^{\nu_6} = P(W_2|W_1 = 1) = (0.5, 0.5)$, $f^{\nu_7} = P(W_2|W_0 = 1, W_1 = 0) = (0.2, 0.8)$, $f^{\nu_8} = f(z_1|z_0, W_0 = 0) = \mathcal{N}(z_1; 1.1z_0, 0.5^2)$, $f^{\nu_9} = f(z_1|z_0, W_0 = 1) = \mathcal{N}(z_1; 3 + 1.1z_0, 0.5^2)$.*

It is clear from Def. 6 that when $\mathbf{Z} = \emptyset$, a CG-PDG model reduces to the multinomial PDG model of Def. 4.

We will extend the meaning of an RFG to include any graph with the structural syntax of Def. 6 and where the nodes contain any real-valued function with the appropriate domain. The definition of the global function $f_G$ in Def. 2 is still valid for such general RFGs and in particular for CG-PDG models. The only minor change would be to Eq. (7) where for a node $\nu$ representing a continuous variable $Z$, the successor nodes are uniquely specified independently of the value of $Z$ (as explained above). That is, for node $\nu$ representing a continuous variable $Z \in \mathbf{X}$ and $\mathbf{x} \in R(\mathbf{X})$, we would define $f_G^{\nu}(\mathbf{x})$ as:

$$f_G^{\nu}(\mathbf{x}) := f^{\nu}(\mathbf{x}) \prod_{Y \in ch_G(Z)} f_G^{succ(\nu,Y)}(\mathbf{x}). \tag{10}$$

We can decompose $f_G$ of CG-PDG $G$ over variables $\mathbf{X}$ as follows. Let $X \in \mathbf{X}$ and let $G \setminus X$ be the CG-PDG structure obtained from $G$ by removing all nodes representing any $X' \in an_G(X)$ (the subtree rooted at any node representing $X$). Then:

$$f_G(\mathbf{x}) := f_{G\setminus X}(\mathbf{x}) \cdot f_G^{reach_G(X,\mathbf{x})}(\mathbf{x}), \tag{11}$$

where $\mathbf{x} \in R(\mathbf{X})$.

The following proposition establishes that when $\mathcal{G}$ *is* a CG-PDG model, then $f_G$ as defined in Def. 2 represents a CG distribution.

**Proposition 1.** *Let $\mathcal{G} = \langle \mathcal{G}, \theta \rangle$ be a CG-PDG model with structure $G = \langle \mathbf{V}, \mathbf{E} \rangle$ over variables $\mathbf{X} = (\mathbf{W}, \mathbf{Z})$ w.r.t. variable forest $F$. Function $f_G$ defines a Conditional Gaussian density over $\mathbf{X}$.*

**Proof:** In order to prove that $f_G$ is a Conditional Gaussian density, we have to show that the joint distribution over the discrete variables is multinomial, and also that for each configuration of the discrete variables, the joint distribution over the continuous variables is multivariate Gaussian (see Sect. 2). That is, we have to show that

i. $\int_{R(\mathbf{Z})} f_G(\mathbf{x}) d\mathbf{z}$ defines a multinomial distribution.
ii. For each $\mathbf{w} \in R(\mathbf{W})$, $f_G(\mathbf{w}, \mathbf{z})$ is a multivariate Gaussian over $\mathbf{Z}$.

If we fix a configuration $\mathbf{w} \in R(\mathbf{W})$, then $f_G$ is just a product of functions of the form $f^{\nu}$, where $\nu$ is a node corresponding to a continuous variable, and therefore, $f_G$ is a product of conditional Gaussians in each branch of the trees in the forest of variables restricted to $\mathbf{w}$, and therefore, for a fixed $\mathbf{w} \in R(\mathbf{W})$, $f_G(\mathbf{w}, \mathbf{z})$ is a multivariate Gaussian density over $\mathbf{z} \in R(\mathbf{Z})$.

Thus, since $f_G(\mathbf{w}, \mathbf{z})$ is a probability density over $R(\mathbf{Z})$, its integral over that domain is equal to 1. Therefore, it holds that

$$\int_{R(\mathbf{Z})} f_G(\mathbf{w}, \mathbf{z}) d\mathbf{z} = f_{G_{\mathbf{W}}}(\mathbf{w}) \prod_{\nu' \in V} \int_{R(\mathbf{Z})} f_G^{\nu'}(\mathbf{z}) d\mathbf{z} = f_{G_{\mathbf{W}}}(\mathbf{w}),$$

where $V = \{\nu' = reach_G(Z, \mathbf{w}) | Z \in \mathbf{Z} \wedge pa_F(Z) \in \{\{\emptyset\} \cup \mathbf{W}\}\}$ (that is $V$ is the set of nodes representing the continuous variables that are roots of a sequence of continuous variable in the variable structure) and $G_{\mathbf{W}}$ is the PDG obtained from structure $G$ by keeping only the variables in $\mathbf{W}$. Finally, according to proposition 3.3 in [15], we know that $f_{G_{\mathbf{W}}}(\mathbf{w})$ defines a multinomial distribution, and hence, so does $\int_{R(\mathbf{Z})} f_G(\mathbf{w}, \mathbf{z}) d\mathbf{z}$. ■

The efficiency of the PDG model over exclusively discrete domains stems from their structure which is a special kind of decision graph only containing chance nodes. The first PDG version presented in [2] extends Binary Decision Diagrams (BDDs) and thereby inherits the efficiency of BDDs, which lies in compact representation and efficient manipulation of boolean functions.

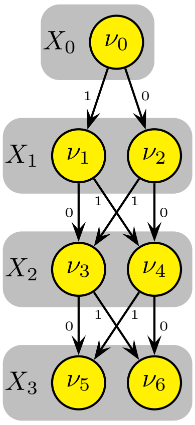


Figure 3: PDG-representation of the parity function.

In Fig. 3, a PDG over 4 binary variables is depicted. The structure encodes the model where $X_3$ is determined by the parity function over the remaining 3 variables that are marginally independent. Adding more variables to the parity function only makes the model grow in size by a small linear factor. Modelling the parity function using a BN model would yield a model that grows by an exponential factor when adding more variables to the function.[3]

The efficiency of the discrete PDG, exemplified by the representation of the parity function (Fig. 3) is inherited by the CG-PDG model. The addition of continuous variables does not restrict the discrete part of the CG-PDG in any way, and the properties of this part of the model stay intact.

## 5. Operations over CG-PDGs

One of the main advantages of the PDG model is that efficient algorithms for exact inference that operate directly on the PDG structure are known. In this section we will show how the original algorithm for exact inference in discrete PDGs by [3] can be almost directly applied to CG-PDGs.

We will first consider the problem of computing the probability (or density value) of some set of variables $\mathbf{Y} \subset \mathbf{X}$ being in the joint state $\mathbf{y} \in R(\mathbf{Y})$ when the joint distribution $P(\mathbf{X})$ is represented by a CG-PDG model $\mathcal{G}$ with

[3] By including suitable artificial latent variables in the domain, there exists an efficient transformation of any PDG into an equivalent BN model [3].

structure $G$. The computation that we wish to perform is what is usually called marginalisation:

$$P\{\mathbf{Y} = \mathbf{y}\} = \sum_{\mathbf{w}' \in R(\mathbf{W}')} \int_{R(\mathbf{Z}')} f_G(\mathbf{w}', \mathbf{z}', \mathbf{y}) d\mathbf{z}' \,, \tag{12}$$

where $\mathbf{W}' = \mathbf{W} \setminus \mathbf{Y}$ and $\mathbf{Z}' = \mathbf{Z} \setminus \mathbf{Y}$. Note that $\mathbf{W}' \cup \mathbf{Z}' \cup \mathbf{Y} = \mathbf{X}$. The next definition is the first step towards efficient computation of Eq. (12).

**Definition 7 (Restriction).** *Let $\mathcal{G}$ be a CG-PDG with structure $G$ over variables $\mathbf{X} = (\mathbf{W}, \mathbf{Z})$, let $\mathbf{Y} \subseteq \mathbf{X}$ and let $\mathbf{y} \in R(\mathbf{Y})$. The* restriction *of $\mathcal{G}$ to $\mathbf{Y} = \mathbf{y}$, denoted as $\mathcal{G}_{\mathbf{Y}=\mathbf{y}}$ is an RFG obtained from $\mathcal{G}$ such that*

1. *$\mathcal{G}$ and $\mathcal{G}_{\mathbf{Y}=\mathbf{y}}$ have the same structure.*
2. *For all $\nu$ representing some discrete variable $X \notin \mathbf{W} \setminus \mathbf{Y}$, $f^\nu$ in $\mathcal{G}_{\mathbf{Y}=\mathbf{y}}$ is copied from $\mathcal{G}$.*
3. *For every discrete variable $W \in \mathbf{Y} \cap \mathbf{W}$ and each node $\nu \in \mathbf{V}_W$, the function $f^\nu(w)$ in $\mathcal{G}_{\mathbf{Y}=\mathbf{y}}$ is copied from $\mathcal{G}$ for $w = \mathbf{y}[W]$ and for any $w \neq \mathbf{y}[Y]$ we set $f^\nu(w) = 0$.*
4. *In all nodes $\nu$ representing a continuous variable $Z \in \mathbf{Y} \cap \mathbf{Z}$, we replace $f^\nu$ with the function value $f^\nu(\mathbf{y}[Z])$.*

*We call the resulting model a* restricted CG-PDG*.*

**Example 3.** *Consider the CG-PDG described in Ex. 2. Its restriction to $(W_2 = 0, Z_0 = 3)$ results in the following changes: $f^{\nu_5}(1) = f^{\nu_6}(1) = f^{\nu_7}(1) = 0$ and $f^{\nu_3}(z_0) = 0.05399097$. This last value results from evaluating at point $3$ a Gaussian density with mean $5$ and standard deviation $1$.*

The restriction operation incorporates into the PDG the information contained in a piece of evidence. Notice that the function value in item 4 of the definition above is a real number only if there are not unobserved continuous variables above $Z$ in the PDG structure. Otherwise, the value of the density would be an algebraic expression depending on the unobserved continuous variables above it. Therefore, we assume that CG-PDGs are restricted in such a way that there are no observed nodes beneath unobserved ones. If the structure is not compatible with the evidence, then it has to be rearranged by swapping nodes, until that restriction is met. For instance, consider two consecutive nodes $\nu_1$ and $\nu_2$ corresponding to variables $Z_1$ and $Z_2$, containing the parameters $(\mu_{z_1}, \sigma^2_{z_1})$ and $(\alpha, \beta, \sigma^2_{z_2|z_1})$, that is, meaning that $Z_1 \sim \mathcal{N}(\mu_{z_1}, \sigma^2_{z_1})$ and $Z_2|Z_1 \sim \mathcal{N}(\alpha + \beta z_1, \sigma^2_{z_2|z_1})$. According to the definition of the CG distribution, parameters $\alpha$ and $\beta$ are computed as

$$\alpha = \mu_{z_2} - \beta \mu_{z_1}$$

and

$$\beta = \frac{\sigma_{z_1,z_2}}{\sigma^2_{z_1}}, \tag{13}$$

where $\sigma_{z_1,z_2}$ stands for the covariance of $Z_1$ and $Z_2$. By swapping the order of $Z_1$ and $Z_2$, the new distributions would be parameterised as $(\alpha', \beta', \sigma^2_{z_1|z_2})$ and $(\mu_{z_2}, \sigma^2_{z_2})$, that is, meaning that $Z_2 \sim \mathcal{N}(\mu_{z_2}, \sigma^2_{z_2})$ and $Z_1|Z_2 \sim \mathcal{N}(\alpha' + \beta' z_2, \sigma^2_{z_1|z_2})$, where

$$\alpha' = \mu_{z_1} - \beta' \mu_{z_2}$$

and

$$\beta' = \frac{\sigma_{z_1,z_2}}{\sigma^2_{z_2}} = \beta \frac{\sigma^2_{z_1}}{\sigma^2_{z_2}}. \tag{14}$$

The unknown values in the expressions above are $\mu_{z_2}$, $\sigma^2_{z_2}$ and $\sigma^2_{z_1|z_2}$, but they can be obtained right on:

$$\mu_{z_2} = E[Z_2] = E[E[Z_2|Z_1]] = E[\alpha + \beta Z_1] = \alpha + \beta E[Z_1] = \alpha + \beta \mu_{z_1}.$$

According to the law of total variance,

$$\sigma^2_{z_2} = \sigma^2_{z_2|z_1} + \mathrm{Var}(E[Z_2|Z_1]) = \sigma^2_{z_2|z_1} + \beta^2 \sigma^2_{z_1}.$$

For the same reason,

$$\begin{aligned} \sigma_{z_1|z_2} &= \sigma^2_{z_1} - \mathrm{Var}(\alpha' + \beta' Z_2) = \sigma^2_{z_1} - \beta'^2 \mathrm{Var}(Z_2) \\ &= \sigma^2_{z_1} - \beta^2 \frac{\sigma^4_{z_1}}{\sigma^4_{z_2}} \sigma^2_{z_2} = \sigma^2_{z_1} - \beta^2 \frac{\sigma^4_{z_1}}{\sigma^2_{z_2}}. \end{aligned}$$

In the general case, the computation of these unknown values is related with the *compilation* operation. Actually, through that operation a restricted CG-PDG can be further modified in order to obtain the mean and variance of all the distributions stored in each node, given the observations. The formal definition of this operation is as follows.

**Definition 8 (Compilation).** *Let $\mathcal{G}$ be a CG-PDG with structure $G$ over variables $\mathbf{X} = (\mathbf{W}, \mathbf{Z})$, let $\mathbf{Y} \subseteq \mathbf{X}$ and let $\mathbf{y} \in R(\mathbf{Y})$. Let $\mathcal{G}_{\mathbf{Y}=\mathbf{y}}$ be the restricted CG-PDG corresponding to evidence $\mathbf{Y} = \mathbf{y}$. The* compilation *of $\mathcal{G}_{\mathbf{Y}=\mathbf{y}}$, denoted as $\mathcal{G}^c_{\mathbf{Y}=\mathbf{y}}$ is an RFG obtained from $\mathcal{G}_{\mathbf{Y}=\mathbf{y}}$ such that*

1. *$\mathcal{G}_{\mathbf{Y}=\mathbf{y}}$ and $\mathcal{G}^c_{\mathbf{Y}=\mathbf{y}}$ have the same structure and parameters for discrete variables.*
2. *For every continuous variable $Z$ with $an_F(Z) \cap \mathbf{Z} = \mathbf{U}$ and every $\nu \in \mathbf{V}_Z$ the following steps are performed in a top-down manner:*
   (a) *A real vector $\mathbf{u}_\nu$ is constructed, indexed by the variables $\mathbf{U}$ and with values $\mathbf{u}_\nu[U] = \mathbf{y}[U]$ if $U \in \mathbf{Y}$ and $\mathbf{u}_\nu[U] = \alpha_{\nu_U}$ if $U \notin \mathbf{Y}$ (where $\nu_U$ is the unique predecessor node of $\nu$ representing $U$). Then a posterior mean $\mu_\nu$ is computed as*

$$\mu_\nu = \alpha_\nu + \boldsymbol{\beta}_\nu^{\mathsf{T}} \mathbf{u}_\nu. \tag{15}$$

(b) *A matrix* $\mathbf{s}_\nu$ *is constructed, indexed by the variables* $\mathbf{S} = an_G(Z) \cap \mathbf{Z}$ *and with values:*

$$\mathbf{s}_\nu[S_1, S_2] = \begin{cases} 0 & \textit{if } S_1 \in \mathbf{Y} \textit{ or } S_2 \in \mathbf{Y} \\ s^2_{\nu_{S_1}} & \textit{if } S_1 = S_2 \textit{ and } S_1 \notin \mathbf{Y} \\ \sigma_{s_1, s_2} & \textit{if } S_1 \notin \mathbf{Y} \textit{ and } S_2 \notin \mathbf{Y} \end{cases}, \tag{16}$$

*where* $\nu_S$ *is the unique predecessor node of* $\nu$ *representing* $S$*. Then a posterior variance* $s^2_\nu$ *is computed as*

$$s^2_\nu = \sigma^2_\nu + \boldsymbol{\beta}^\intercal_\nu \mathbf{s}_\nu \boldsymbol{\beta}. \tag{17}$$

*We call the resulting model a* compiled CG-PDG*.*

Please note that as we are computing step 2 above in a top down sequence, the $s^2_{\nu_{S_1}}$ in the second case of Eq. (16) will always be available from a previous computation. Also note that the covariances required in Eq.(17), can be computed from the $\beta$ coefficients using Eq. (4) given the recursive nature of step 2. in Def. 8.

**Example 4.** *Consider the scenario in Ex. 2. Assume we want to compile the CG-PDG described there, in order to incorporate evidence* $(W_2 = 0)$*. The restriction of the model to* $(W_2 = 0)$ *results in the following changes:* $f^{\nu_5}(1) = f^{\nu_6}(1) = f^{\nu_7}(1) = 0$*, and the compilation of the restricted models requires the updating of the following parameters:* $\mu_{\nu_8} = 5.5$*,* $s^2_{\nu_8} = 1.46$*,* $\mu_{\nu_9} = 8.5$ *and* $s^2_{\nu_9} = 1.9924$*.*

From a *compiled* CG-PDG $\mathcal{G}_{\mathbf{Y}=\mathbf{y}}$ we can compute the probability of the evidence $P\{\mathbf{Y} = \mathbf{y}\}$ as:

$$P\{\mathbf{Y} = \mathbf{y}\} = \sum_{\mathbf{w} \in R(\mathbf{W})} \int_{R(\mathbf{Z})} f_{G_{\mathbf{Y}=\mathbf{y}}}(\mathbf{w}, \mathbf{z}) d\mathbf{z}. \tag{18}$$

In the following we will show how (18) is computed by local computations in the nodes.

We define the *outflow* as the accumulated function value of the real function $f^\nu_G$ defined recursively at $\nu$ by Eq. (7) over its full domain.

**Definition 9.** *Let* $\mathcal{G}$ *be a (possibly compiled) CG-PDG with structure* $G$ *over variables* $\mathbf{X}$ *w.r.t. forest* $F$*. The* outflow *of* $\nu$ *representing random variable* $X_i$ *is defined as:*

$$\mathit{ofl}(\nu) := \sum_{\mathbf{w} \in R(\mathbf{W} \cap de^*_F(X_i))} \int_{R(\mathbf{Z} \cap de^*_F(X_i))} f^\nu_G(\mathbf{w}, \mathbf{z}) d\mathbf{z}. \tag{19}$$

Notice that in an uncompiled CG-PDG the outflow of all nodes is 1. Also notice that Eq. (18) is equal to the product of outflows of all root nodes in the structure.

The next proposition is central in the efficient computation of *outflow*:

**Proposition 2.** *Let $\mathcal{G}$ be a (possibly compiled) CG-PDG with structure $G$ w.r.t. forest $F$ over variables $\mathbf{X}$. The outflow is recursively computed as follows:*

1. *If $\nu$ is a node representing a discrete variable $W$:*

$$ofl(\nu) = \sum_{w \in R(W)} f^{\nu}(w) \prod_{Y \in ch_F(W)} ofl(succ(\nu, Y, w)) \,. \tag{20}$$

2. *If $\nu$ is a node representing a continuous variable $Z$:*

$$ofl(\nu) = \int_{R(Z)} f^{\nu}(z) \prod_{Y \in ch_F(Z)} ofl(succ(\nu, Y)) dz \,. \tag{21}$$

**Proof:** Item 1 is shown in [3, Lemma 4.3]. To prove item 2 we just have to remember that, in a RFG containing continuous variables, all the variables below any continuous variable are continuous as well. Therefore, we have to instantiate Eq. (19) to the case in which there are no discrete variables involved and hence the summation disappears and we are left with only the integration of function $f_G^{\nu}$. Expanding $f_G^{\nu}$ using Eq. (7) we get Eq. (21). ■

Extending previous results of [3, Theorem 4.4], Proposition 2 and the fact that Eq. (18) equals the product of outflows of root nodes, yields an efficient computation of $P\{\mathbf{Y} = \mathbf{y}\}$.

We will now turn to the computation of posterior probability distribution $P(W|\mathbf{Y} = \mathbf{y})$ and posterior densities $f(z|\mathbf{Y} = \mathbf{y})$. We will need to be able to talk about parts of a domain $R(\mathbf{U})$, $\mathbf{U} \subseteq \mathbf{X}$, that reach a specific node, so we define a *Path*-relation as follows:

**Definition 10 (Path).** *Let $\mathcal{G}$ be a (possibly compiled) CG-PDG model with structure $G$ w.r.t. forest $F$ over variables $\mathbf{X}$ and let $\nu$ represent $X \in \mathbf{X}$, $an_F(X) \subseteq \mathbf{Y} \subseteq \mathbf{X}$ and $\mathbf{W}' = \mathbf{Y} \cap \mathbf{W}$. Then*

$$\begin{aligned} Path_G(\nu, \mathbf{Y}) := \{&\mathbf{w}' \in R(\mathbf{W}') \textit{ such that} \\ &\exists \mathbf{x} \in R(\mathbf{X}) : (reach_G(\mathbf{x}, X) = \nu \textit{ and } \mathbf{x}[\mathbf{W}'] = \mathbf{w}')\} \,. \end{aligned} \tag{22}$$

If we consider the structure of Fig. 2 we have that e.g. $Path_G(\nu_6, \{W_1, W_0\}) = \{\{0,1\}, \{1,1\}\}$.

The *inflow* of a node $\nu$ is the accumulation of values of $f_G$ over the part of the domain that reaches $\nu$, and we define it formally as follows.

**Definition 11.** *Let $\mathcal{G}$ be a CG-PDG model with structure $G$ over variables $\mathbf{X} = (\mathbf{W}, \mathbf{Z})$ and forest $F$. Let $\nu \in \mathbf{V}_{X_i}$, $G \setminus X_i$ be the structure obtained from $G$ by removing every node labelled with $X_i$ and their descendants, $\mathbf{W}' = \mathbf{W} \setminus de_F^*(X_i)$ and $\mathbf{Z}' = \mathbf{Z} \setminus de_F^*(X_i)$. The* inflow *of $\nu$ is defined as:*

$$ifl(\nu) := \sum_{\mathbf{w} \in Path_G(\nu, \mathbf{W}')} \int_{R(\mathbf{Z}')} f_{G \setminus X_i}(\mathbf{w}, \mathbf{z}) d\mathbf{z} \,. \tag{23}$$

*When $\{\mathbf{W}' \cup \mathbf{Z}'\} = \emptyset$ (that is, when $X_i$ is a root), we define $ifl(\nu) = 1$.*

For a node $\nu$ in a CG-PDG with structure $G$ over $\mathbf{X}$, the set $Path_G(\nu, \mathbf{X})$ is the part of the domain in which the local function $f^\nu$ is included as a factor in the global function $f_G$. The *inflow* and *outflow* of a node $\nu$ factorises the accumulated function value of $f_G$ over $Path_G(\nu, \mathbf{X})$ in two independent factors.

**Lemma 1.** *Let $\mathcal{G}$ be a (possibly compiled) CG-PDG with structure $G$ over variables $\mathbf{X}$. For any node $\nu$ in $G$, it holds that*

$$ifl(\nu)\, ofl(\nu) = \sum_{\mathbf{w} \in Path_G(\nu, \mathbf{W})} \int_{R\mathbf{Z}} f_G(\mathbf{w}, \mathbf{z}) d\mathbf{z}. \tag{24}$$

**Proof:** We wish to compute the product $ifl(\nu)\, ofl(\nu)$ for an arbitrary node $\nu$ in a CG-PDG. Let node $\nu$ represent variable $X_i$, then $Path_G(\nu, \mathbf{W})$ can be decomposed as $Path_G(\nu, \mathbf{W}) = Path_G(\nu, \mathbf{W} \setminus de^*_F(X_i)) \times R(\mathbf{W} \cap de^*_F(X_i))$, and obviously $R(\mathbf{Z})$ can be decomposed as $R(\mathbf{Z}) = R(\mathbf{Z} \setminus de^*_F(X_i)) \times R(\mathbf{Z} \cap de^*_F(X_i))$. Then:

$$ifl(\nu)\, ofl(\nu) = \sum_{\substack{\mathbf{w} \in \\ Path_G(\nu, \mathbf{W})}} \int_{R(\mathbf{Z})} f_{G \setminus X_i}(\mathbf{w}', \mathbf{z}') f^\nu_G(\mathbf{w}'', \mathbf{z}'') d\mathbf{z}\,,$$

where $\mathbf{w}'$ (and $\mathbf{z}'$) are projections of $\mathbf{w}$ (and $\mathbf{z}$) onto $\mathbf{X} \setminus de^*_F(X_i)$, while $\mathbf{w}''$ (and $\mathbf{z}''$) are projections onto $de^*_F(X_i)$. Finally, from Def. 2 we have that the product $f_{G \setminus X_i}(\mathbf{w}', \mathbf{z}') f^\nu_G(\mathbf{w}'', \mathbf{z}'')$ equals $f_G(\mathbf{w}, \mathbf{z})$. ■

The next theorem establishes the basis for probabilistic inference in CG-PDGs. It indicates how the posterior distribution of every discrete or continuous variables can be obtained by local computations. Furthermore, it also shows how the expectation and variance of each continuous variable can be computed using local computations. The computation of the expected value and variance of any discrete variable is straightforward from Eq. (25), and therefore it is not included in the theorem.

**Theorem 1.** *Let $\mathcal{G}_{\mathbf{Y}=\mathbf{y}}$ be a CG-PDG model restricted to evidence $\mathbf{Y} = \mathbf{y}$. Let $\mathcal{G}^c_{\mathbf{Y}=\mathbf{y}}$ be its compiled version. When* ifl *and* ofl *values have been computed for all nodes in $\mathcal{G}^c_{\mathbf{Y}=\mathbf{y}}$, the following holds. For any discrete variable $W \in \mathbf{W}$ where $W \notin \mathbf{Y}$,*

$$P\{W = w | \mathbf{Y} = \mathbf{y}\} = \gamma \sum_{\nu \in \mathbf{V}_W} f^\nu(w)\, ifl(\nu) \prod_{U \in ch_F(W)} ofl(succ(\nu, U, w))\,. \tag{25}$$

*For any continuous variable $Z \in \mathbf{Z}$, $Z \notin \mathbf{Y}$, it holds that*

$$f(z | \mathbf{Y} = \mathbf{y}) = \gamma \sum_{\nu \in \mathbf{V}_Z} f^\nu(z)\, ifl(\nu) \prod_{U \in ch_F(Z)} ofl(succ(\nu, U))\,. \tag{26}$$

*Furthermore,*

$$\mathrm{E}[Z | \mathbf{Y} = \mathbf{y}] = \gamma \sum_{\nu \in \mathbf{V}_Z} \mu_\nu\, ifl(\nu) \prod_{U \in ch_F(Z)} ofl(succ(\nu, U))\,, \tag{27}$$

*and*

$$\mathrm{Var}(Z|\mathbf{Y}=\mathbf{y}) = \gamma \sum_{\nu\in\mathbf{V}_Z} s_\nu^2 \mathit{ifl}(\nu)^2 \prod_{U\in ch_F(Z)} \mathit{ofl}(\mathit{succ}(\nu,U))^2 \,. \qquad (28)$$

*In all equations $\gamma$ is the normalising factor $\frac{1}{P\{\mathbf{Y}=\mathbf{y}\}}$. In Eq. (27) and Eq. (28), $\mu_\nu$ and $s_\nu^2$, respectively, are computed during compilation (see Def. 8).*

**Proof:** Equations (25) and (26) are a direct consequence of Lemma 1. Now we have to show that the values $\mu_\nu$ and $s_\nu^2$, calculated according to Eq. (15) and (17) correspond to the posterior mean and variance of the distribution stored in node $\nu$. But that is a direct consequence of Equations (3), (4), (5) and (6).

Note that if the CG-PDG is compiled for $\mathbf{Y}=\mathbf{y}$, then for each variable $U\in\mathbf{U}=an_F(Z)\cap\mathbf{Z}$, its expectation is $E[U]=\alpha_{\nu_U}$ if $U\in\mathbf{Y}$ (where $\nu_U$ is the unique predecessor node of $\nu$ representing $U$) and $E[U]=\mathbf{y}[U]$ if $U\notin\mathbf{Y}$. Therefore, for any $\nu\in\mathbf{V}_Z$, the value $\mu_\nu$ computed as in Eq. (15) is actually $E[Z|\mathbf{Y}=\mathbf{y}]$ for the distribution stored in $\nu$.

Note that if the CG-PDG is compiled for $\mathbf{Y}=\mathbf{y}$, then for each variable $U\in\mathbf{U}$, its variance becomes 0 if $U\in\mathbf{Y}$, as well as the covariance with any other variable. Therefore, for any $\nu\in\mathbf{V}_Z$, the value $s_\nu^2$ computed as in Eq. (17) is actually $\mathrm{Var}(Z|\mathbf{Y}=\mathbf{y})$ for the distribution stored in $\nu$.

Also, note that $f(z|\mathbf{Y}=\mathbf{y})$ in equation (26) is a mixture of Gaussian densities, and therefore the expectation of $Z$ is trivially the one in equation (27) and its variance is the one in equation (28). ■

The next proposition is central in the efficient computation of *inflow*.

**Proposition 3.** *Let $\mathcal{G}$ be a (possibly compiled) CG-PDG with structure $G$ w.r.t. forest $F$ over variables $\mathbf{X}$. The inflow is recursively computed as follows:*

1. *If $\nu$ is a root,*

$$\mathit{ifl}(\nu) = \prod_{\nu'\neq\nu,\nu' \text{ is root}} \mathit{ofl}(\nu') \,. \qquad (29)$$

2. *If $\nu$ is not a root, and $X_p = pa_F(X_i)$, and $X_p$ is discrete:*

$$\mathit{ifl}(\nu) = \sum_{x\in R(X_p)} \sum_{\substack{\nu':\\ \nu=\mathit{succ}(\nu',X_i,x)}} [\mathit{ifl}(\nu') f^{\nu'}(x) \prod_{Y\in ch_F(X_p)\setminus X_i} \mathit{ofl}(\mathit{succ}(\nu',Y,x))] \,. \quad (30)$$

3. *If $\nu$ is representing continuous variable $X_i$, $\nu$ is not a root, $X_p = pa_F(X_i)$, $X_p$ is continuous and $\nu'$ is the parent of $\nu$:*

$$\mathit{ifl}(\nu) = \mathit{ifl}(\nu') \prod_{Y\in ch_F(X_p)\setminus X_i} \mathit{ofl}(\mathit{succ}(\nu',Y)) \,. \qquad (31)$$

**Proof:** Items 1 and 2 are shown in [3, Lemma 4.3]. Item 3 follows by realizing that nodes representing continuous variables only have one outgoing arc, at most, towards each child variable. ■

**Proposition 4.** *Computing* inflow *and* outflow *for all nodes in a (possibly restricted) CG-PDG can be done in time linear in the number of edges of the model.*

**Proof:** The proof is a simple extension of the proof of the result [3, Theorem 4.4]. ■

Theorem 1 and Proposition 4 demonstrate that typical probabilistic queries can be answered in time linear in the size of the CG-PDG. The main concern in achieving efficient inference can therefore be directly focused on constructing a small model, which of course may be difficult or even impossible. The size may be exponential in the number of discrete variables in the domain. However, it is considered an advantage to be able to determine complexity of inference directly in the model, as opposed to BN models where inference complexity depends on the size of a secondary Junction Tree model obtained from the BN.

Notice, however, that belief updating is carried out over compiled CG-PDGs. The complexity of the compilation operation is quadratic in the number of continuous variables in the longest brach of the tree of variables of the CG-PDG. This complexity is determined by the need of handling the covariance matrix, which is of quadratic size in the number of variables involved.

**Example 5 (CG-PDG belief updating).** *Consider Ex. 2. Assume we have evidence that the route was not finished in time* ($W_2 = 0$)*, and we then want to update our beliefs of the remaining unknown variables. The first step is to compile the CG-PDG in order to incorporate the evidence. This step is detailed in Ex. 4. After compiling the model, we can compute the outflows using the recursive formulas in Prop. 2. Here we list values consecutively as* $\{\mathit{ofl}(\nu_0), \mathit{ofl}(\nu_1) \ldots \mathit{ofl}(\nu_9)\}$*: {0.10265, 0.0545, 0.215, 1, 1, 0.05, 0.5, 0.2, 1, 1}. Once outflows are computed, inflows can be computed according to Prop. 3, obtaining: {1, 0.7, 0.3, 0.03815, 0.0645, 0.693, 0.022, 0.285, 0.03815, 0.0645}.*

*First, as mentioned earlier, the probability of evidence is just the product of outflows of root nodes which in this example means just* $\mathit{ofl}(\nu_0) = P\{W_2 = 0\} = 0.10265$*. Next, computing the posterior expectations of the continuous variables is done top down from the root to the leaves using Eq. (27) with* $\gamma = \frac{1}{P\{W_2=0\}}$*, and we get* $\mathrm{E}[Z_0|W_2 = 0] = 8.14$ *and* $\mathrm{E}[Z_1|W_2 = 0] = 7.39$*.*

*Posterior variances are computed as a weighted average of the variances stored in nodes representing the given variable using Eq. (28), which yields:* $\mathrm{Var}[Z_0|W_2 = 0] = 0.897$ *and* $\mathrm{Var}[Z_1|W_2 = 0] = 0.1014$*.*

*Finally, computing the marginal distributions for the two unobserved discrete variables* $W_0$ *and* $W_1$ *we use Eq. (25) and get:* $P\{W_0|W_2 = 0\} = \{0.37, 0.63\}$ *and* $P\{W_1|W_2 = 0\} = \{0.89, 0.11\}$*.*

## 6. Modelling CG Bayesian networks using CG-PDGs

In this section we show how a CG Bayesian network [13] can be modelled using a CG-PDG. More precisely, we will concentrate on the context of belief updating. The usual approach to exact belief updating in BN models is by first compiling the model into a Junction Tree (JT) and then performing the computations in this secondary structure. Belief updating in JTs has linear complexity in the size of the model, where in this case we take the number of free parameters of a model to be its size. The size of a clique of a JT composed just by discrete variables is the product of the number of possible values of the variables in the clique, whilst if the clique only contains continuous variables, the size is the number of elements in the covariance matrix (except symmetries) plus the elements in the vector of means.

**Theorem 2.** *Let $J$ be a junction tree over mixed domain $\mathbf{X} = \mathbf{W} \cup \mathbf{Z}$ with CG clique potentials and at least one strong root (see [16]). Then there exists a CG-PDG $\mathcal{G}$ such that:*

- *$\mathcal{G}$ encodes the same joint density as $J$, and*
- *structure $G$ of $\mathcal{G}$ has size linear in the size of $J$.*

**Proof:** We examine the following three cases separately, $\mathbf{Z} = \emptyset$, $\mathbf{W} = \emptyset$ and $\mathbf{Z} \neq \emptyset \wedge \mathbf{W} \neq \emptyset$:

**$\mathbf{Z} = \emptyset$:** In this case the theorem reduces to [3, Theorem 5.1].

**$\mathbf{W} = \emptyset$:** Without loss of generality, we assume that $J$ contains one connected component. If $J$ contains more than one connected component, the following steps are performed for each one of them. We then choose a root clique $C_r$ from $J$ at random, and form a directed tree over the cliques by directing all edges away from $C_r$. Following [3], we denote by $new(C_i)$ the set of variables $C_i \setminus C_j$, where $C_j = pa_J(C_i)$. A variable tree $T$ is then constructed top down by substituting for each clique $C_j$ a linear sequence the variables $new(Cj)$, and branching whenever $J$ branches. Following the definition of the structure of the CG-PDG (see Def. 5) it is clear that each variable will be represented in the CG-PDG $G$ constructed wrt. $T$ by a single node. The local function of node $\nu$, representing variable $Z_i$ which was substituted for clique $C_j$ and where $an_T(Z_i) \cap C_j = \mathbf{U}$, is initialised to the conditional density $f^{\nu}(z_i|\mathbf{U})$ which can be extracted from the potential assigned to clique $C_j$ using the formulae given in Section 2. When $an_T(Z_i) \setminus C_j \neq \emptyset$, the remaining dependents are cancelled by assigning those a zero as $\beta$ value in the local function $f^{\nu}$. From the chain-rule one can now show that the graph function $f_G$ represents the original multivariate Gaussian from $J$.

Concerning the number of parameters, we first assume that $J$ represents a clique $C_i$ in $J$ by a covariance matrix and a mean vector. This means that in total $J$ uses:

$$size(J) = \sum_{C_i \in J} |C_i|^2 + |C_i| \,, \tag{32}$$

parameters. In the CG-PDG we represent in each node $\nu$ the conditional density of the variable $Z$ represented by $\nu$ given its predecessors $an_T(Z)$, which means that we neeed to represent parameters $\alpha$, $\sigma$ and the $\beta$ vector of length $|an_T(Z)|$. Hence, an upper bound on the number of parameters in $G$ is:

$$size(G) \leq \sum_{i=2}^{n} i = \frac{n^2+n-2}{2}\,, \tag{33}$$

where $n$ is the number of variables. Eq. (33) corresponds to arranging all variables in a sequence without any branching. In general we can express the number of parameters in the CG-PDG as:

$$size(G) = \sum_{C_i \in J} \sum_{Z \in new(C_i)} 2 + |an_T(Z)|\,.$$

Some of the entries in the $\beta$-vector will be zero, and if we count only the (possibly) non-zero entries for a variable $Z$ that was part of the substitution for clique $C_i$ we then get:

$$2 + |an_T(Z) \cap C_i|\,.$$

The total number of non-zero parameters that a clique $C_i$ in $J$ will result in, is then:

$$\sum_{a=0}^{|new(C_i)|} 2 + |C_i \setminus new(C_i)| = \frac{|new(C_i)|(4 + 2|C_i \setminus new(C_i)| + |new(C_i)| - 1)}{2}\,. \tag{34}$$

In Eq. (34) we have used the general formula for computing finite sums of the arithmetic progressions. Summing (34) over all cliques then yields the total number of (possibly) non-zero parameters in $G$:

$$\begin{aligned} size_{non-zero}(G) &= \\ &\sum_{C_i \in J} \frac{|new(C_i)|(4 + 2|C_i \setminus new(C_i)| + |new(C_i)| - 1)}{2} \\ &\leq \sum_{C_i \in J} 3|new(C_i)| + \frac{5}{4}|C_i|^2 \end{aligned} \tag{35}$$

It is clear that (35) is linear in (32).

**$\mathbf{Z} \neq \emptyset \wedge \mathbf{W} \neq \emptyset$:** In this case we choose a strong root of $J$ when directing the structure. We then proceed from the root down substituting linear sequences of variables for cliques. As we have chosen a strong root, we will get a variable forest structure where no discrete variable is located below a continuous variable. To induce the CG-PDG structure wrt. the variable

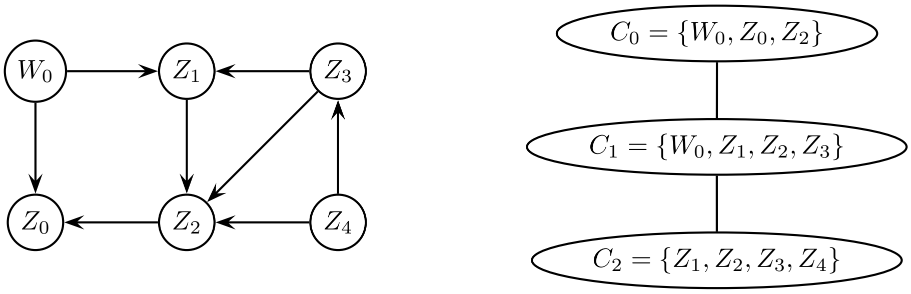

$W_0$
$Z_1$
$Z_3$
$Z_0$
$Z_2$
$Z_4$
$C_0 = \{W_0, Z_0, Z_2\}$
$C_1 = \{W_0, Z_1, Z_2, Z_3\}$
$C_2 = \{Z_1, Z_2, Z_3, Z_4\}$

Figure 4: A mixed BN and its Junction Tree

forest constructed in this way, we first arrange the discrete part following [3]. Then, for each discrete variable with a continuous variable $Z$ as child, we can apply the approach outlined above for exclusively continuous variables, once for each relevant discrete joint configuration. That is, if $Z$ was part of the substitution for clique $C_i$, then we will have a unique node for each joint configuration of the discrete variables of $C_i$, which is one for each member of $R(C_i \cap \mathbf{W})$. The theorem then follows from the correctness of the theorem in the above two cases.

■

**Example 6.** *Consider the BN and JT of Figure 4, taken from [16, Example 2]. The only discrete variable is $W_0$ and the rest are continuous with conditional Gaussian distribution. There are two possible strong roots of the JT, namely $C_0$ or $C_1$. We choose one (say $C_1$) and construct the variable forest for the CG-PDG as described in the proof of Theorem 5.1 in [3]. The method proposed by [3] was devised for discrete domains, but for the construction of the variable forest of our mixed domain, it can be readily applied with one additional constraint: that discrete variables are always added to the forest above continuous variables. That is, when substituting a clique $C$ by a linear sequence of variables, all discrete variables in $C$ are added before any continuous variables in $C$. We obtain the variable forest in Fig 5(a). Now, adding the structure and parameters to the model is done first for the discrete nodes according to the method of [3]. This effectively gives us a PDG over only the discrete variables that encodes the joint distribution over the discrete variables found in the junction tree. In our example, this simply means adding a single node $\nu_0$ representing $W_0$ and with parameter $f_{\nu_0} = P(W_0)$. Then, for the continuous variables with discrete parent we add one node for each state and node of the parent. For the rest of the continuous variables the structure will then be given from the structural syntax of the CG-PDG (see structure of 5(b)). The CG densities for the continuous variables are then computed from the covariance matrix and mean vector from* *the clique potential. Then, for the nodes representing $Z_2$ the CG densities are:*

$$\begin{aligned} f^{\nu_1}(z_2) &= \mathcal{N}(z_2; \mu_{z_2}, \sigma^2_{z_2}) \\ f^{\nu_2}(z_2) &= \mathcal{N}(z_2; \mu_{z_2}, \sigma^2_{z_2}) \end{aligned}$$

*which can be read directly from the potential of $C_1$, using the matrix and vector for $W_0 = w_0$ in $f^{\nu_1}$ and the ones for $W_0 = w_1$ in $f^{\nu_2}$. For $Z_1$ the density is:*

$$f^{\nu_3}(z_1) = \mathcal{N}(z_1; \alpha_{\nu_3} + \beta_{\nu_3} z_2, \sigma^2_{\nu_3}),$$

*where*

$$\begin{aligned} \alpha_{\nu_3} &= \mu_{z_1} - \frac{\sigma_{z_1,z_2}}{\sigma^2_{z_2}} \mu_{z_2}, \\ \beta_{\nu_3} &= \frac{\sigma_{z_1,z_2}}{\sigma^2_{z_2}}, \\ \sigma^2_{\nu_3} &= \sigma^2_{z_1} - \frac{\sigma^2_{z_1,z_2}}{\sigma^2_{z_2}} = \sigma^2_{z_1} - \beta_{\nu_3}\sigma_{z_1,z_2}. \end{aligned}$$

*In the above formula the covariance matrix and means vector from $C_1$ for $W_0 = w_0$ is used, and for $\nu_4$ we would then use the matrices for $W_0 = w_0$. Moving on to $Z_3$, we get:*

$$f^{\nu_5}(z_3) = \mathcal{N}(z_3; \alpha_{\nu_5} + \beta'_{\nu_5} z_2 + \beta''_{\nu_5} z_3, \sigma^2_{\nu_5}),$$

*where*

$$\begin{aligned} \alpha_{\nu_5} &= \mu_{z_3} - \frac{\sigma_{z_3,z_1}}{\sigma^2_{z_1}} \mu_{z_1} - \frac{\sigma_{z_3,z_2}}{\sigma^2_{z_2}} \mu_{z_2}, \\ \beta'_{\nu_5} &= \frac{\sigma_{z_3,z_1}}{\sigma^2_{z_1}}, \\ \beta''_{\nu_5} &= \frac{\sigma_{z_3,z_2}}{\sigma^2_{z_2}}, \\ \sigma^2_{\nu_5} &= \sigma^2_{z_3} - \frac{\sigma^2_{z_3,z_1}}{\sigma^2_{z_1}} - \frac{\sigma^2_{z_3,z_2}}{\sigma^2_{z_2}} = \sigma^2_{z_3} - \beta'_{\nu_5}\sigma_{z_3,z_1} - \beta''_{\nu_5}\sigma_{z_3,z_2}. \end{aligned}$$

*The covariance matrix and means vector in the above formula are from $C_1$ for $W_0 = w_0$. We now move to $Z_0$, which has been substituted for clique $C_0$. Hence, in the following formula, we use the potential from $C_0$, and get:*

$$f^{\nu_7}(z_0) = \mathcal{N}(z_0; \alpha_{\nu_7} + \beta'_{\nu_7} z_2 + \beta''_{\nu_7} z_0 + \beta'''_{\nu_7} z_3, \sigma^2_{\nu_7}),$$

*where*

$$\begin{aligned}
\alpha_{\nu_7} &= \mu_{z_0} - \frac{\sigma_{z_0,z_2}}{\sigma^2_{z_2}}\mu_{z_2}\,, \\
\beta'_{\nu_7} &= \frac{\sigma_{z_0,z_2}}{\sigma^2_{z_2}}\,, \\
\beta''_{\nu_7} &= 0\,, \\
\beta'''_{\nu_7} &= 0\,, \\
\sigma^2_{\nu_5} &= \sigma^2_{z_2} - \frac{\sigma^2_{z_2,z_1}}{\sigma^2_{z_1}} = \sigma^2_{z_2} - \beta'_{\nu_7}\sigma_{z_2,z_1}\,.
\end{aligned}$$

*The zero $\beta$'s in the above formula are due to the fact that we do not find neither $Z_3$ nor $Z_1$ in $C_0$, and the computation of $\sigma$ and $\alpha$ simplifies accordingly.*

*In general, for a node $\nu$ representing continuous variable $X$ with continuous predecessors $\mathbf{Z}$ in the variable forest, let $C$ be the clique from which $X$ was taken and let $\mathbf{Y} = \mathbf{Z} \cap C$. Then we compute $\alpha_\nu$, $\beta_\nu$'s and $\sigma_\nu$ as:*

$$\alpha_\nu = \mu_x - \sum_{Z\in\mathbf{Y}}^{n} \frac{\sigma_{x,z}}{\sigma^2_z}\mu_z\,,$$

*and for all $Z \in \mathbf{Y}$*

$$\begin{aligned}
\beta_\nu[Z] &= \frac{\sigma_{x,z}}{\sigma^2_z}\,, \\
\sigma^2_\nu &= \sigma^2_x - \sum_{Z\in\mathbf{Y}}\sum_{Z'\in\mathbf{Y}} \beta_\nu[Z]\beta_\nu[Z']\sigma_{z,z'}\,.
\end{aligned}$$

*In the above formula the covariance and mean is taken from the clique potential for which the variable $X$ was substituted and the relevant configuration of any discrete variables is used.*

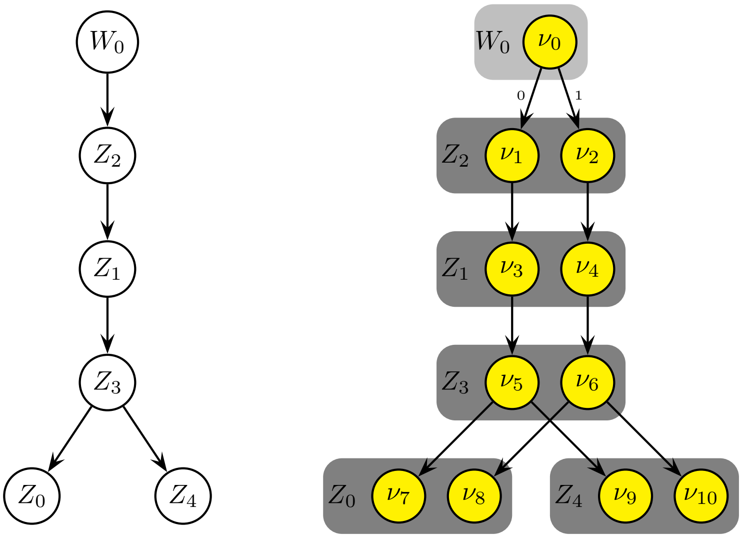
## 7. An experiment comparing JTs and CG-PDGs

In this section we describe an experiment aimed at illustrating the transformation of a JT into a CG-PDG model described in the previous section.

We wish to compare the CG-PDG model to the Junction Tree model empirically, and we therefore repeat one of the experimental settings presented in [17]. In short, the experiment consists of transforming a Junction Tree into an equivalent CG-PDG model and subsequently simplifying the obtained CG-PDG model by merging nodes in the structure. We will review the structural operation of merging nodes used in [17] which is defined only for nodes representing discrete random variables.

**Definition 12.** *Two nodes $\nu_1$ and $\nu_2$ are mergeable iff:*

1. *$\nu_1$ and $\nu_2$ represent the same variable $W$, and*



(a)

(b)

Figure 5: (a) Variable forest, and (b) CG-PDG structure.

2. *for each* $W_i \in ch(F)W$ *and every* $w \in R(W)$ *it holds that* $succ(\nu_1, Y, w)$ *and* $succ(\nu_2, Y, w)$ *are the same node.*

So, two mergeable nodes represent the same variable and have the same children. E.g., in Fig. 2 nodes $\nu_5$ and $\nu_6$ are mergeable, but $\nu_1$ and $\nu_2$ are not as they disagree on the child for value $W_1 = 0$. The structural operation of merging of two mergeable nodes is defined as follows.

**Definition 13.** *Let two nodes* $\nu_1$ *and* $\nu_2$ *in CG-PDG structure* $G$ *be mergeable and representing variable* $W$ *where* $pa_G(W) = W_p$*. By merging* $\nu_1$ *and* $\nu_2$ *we understand the removal of* $\nu_1$ *and* $\nu_2$ *from* $G$ *and the introduction of a new node* $\nu'$ *representing* $W$ *that has the same successors as* $\nu_1$ *and* $\nu_2$ *and has the union of parents of* $\nu_1$ *and* $\nu_2$ *as parents.*

As an example, in Fig. 6(a) and (b) we depict the structure from Fig. 2 after two merge operations. First we merge $\nu_5$ and $\nu_7$ (Fig. 6(a)) and then we merge $\nu_1$ and $\nu_2$. Notice how in the original model (Fig. 2) $\nu_1$ and $\nu_2$ were not mergeable, but only becomes mergeable by first merging $\nu_5$ and $\nu_7$.

We want to also be able to simplify the continuous part of the structure, and obviously the merging of nodes representing discrete random variables does not translate directly to nodes representing continuous random variables.

According to the structural syntax of CG-PDGs, a node representing a continuous random variable can only have a single child. Instead of merging nodes, we will collapse entire branches of nodes representing the same sequence of continuous random variables. Structurally, the collapsing of two continuous

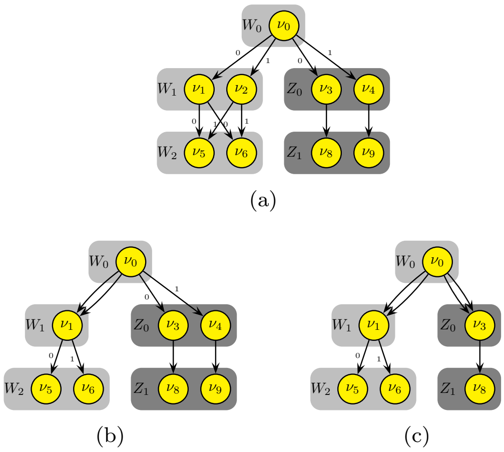


Figure 6: An example of merge and collapse operations performed on the structure of Fig. 2. In (a), original nodes $\nu_5$ and $\nu_7$ have been merged and a new $\nu_5$ node has been introduced. In (c), nodes $\nu_1$ and $\nu_2$ are merged to form a new $\nu_1$. Finally, in (c) the two branches rooted at $\nu_3$ and $\nu_4$ are collapsed forming a single branch rooted at the new $\nu_3$.

branches rooted at nodes $\nu_i$ and $\nu_j$ means to remove one of the branches, say the one rooted at $\nu_i$, and redirecting the edges pointing into $\nu_i$ to point into $\nu_j$.

In Fig. 6(c) we depict the result of merging the two branches rooted at $\nu_3$ and $\nu_4$ into a single branch.

In order to guide the merging/collapsing of nodes/branches, we use the Kullback-Leibler divergence between the two joint densities. When choosing between two possible merge/collapse operations we select the one that yields the smallest Kullback-Leibler divergence between the local joint densities.

In our experiment we used the well-known Waste-Incinerator network described in [12]. We loaded the network in the Hugin$^{\text{TM}}$tool[4], extracted the JT and constructed the equivalent CG-PDG model from it. We then performed merge and collapse operations, every time selecting the operation yielding minimal Kullback-Leibler divergence. After each merge/collapse we collect statistics on the resulting structure and also measure the log-likelihood of the model given a database that was sampled from the original network. In Table 1 we list these values.

The original Junction Tree model had 111 parameters, and our first observation is then that the equivalent CG-PDG model has considerably more parameters, in this case 142. This is not at all unexpected as redundant branches are easily created in order for the CG-PDG structure to encode the correct independence structure. We also notice that we are only able to collapse a few branches without significantly harming the accuracy of the resulting model as measured by the log-likelihood of the data. The third collapse operation yields

[4] `http://www.hugin.com`

| KL | #dis | #con | #nodes | size | ll |
|---|---|---|---|---|---|
| - | 7 | 48 | 55 | 142 | 4.70 |
| 100.62 | 7 | 42 | 49 | 126 | 3.92 |
| 109.81 | 7 | 36 | 43 | 110 | 0.73 |
| 12264.65 | 7 | 30 | 37 | 94 | −1189.96 |
| 0.00 | 6 | 30 | 36 | 92 | −1189.96 |
| 12546.24 | 6 | 24 | 30 | 76 | −8241.87 |
| 0.00 | 5 | 24 | 29 | 74 | −8241.87 |
| 60345.84 | 5 | 18 | 23 | 58 | −13310.96 |
| 67839.30 | 5 | 12 | 17 | 42 | −43267.70 |

Table 1: Results of merging/collapsing the CG-PDG representation of the Waste-Incinerator Bayesian Network. The columns are: Kullback-Leibler divergence between previous model and current one (KL), number of nodes representing discrete variables (#dis), number of nodes representing continuous variables (#con), total number of nodes (#nodes), number of parameters (size) and average log-likelihood of single data cases (ll).

a decrease from 0.73 to -1189.96. As we continue collapsing the accuracy deteriorates further. A positive observation is that by only two collapse operations we have arrived at a CG-PDG structure with size 110 and without drastically worsening the accuracy, log-likelihood is 0.73 compared to 4.70 of the original Junction Tree.

In this experiment we have used a toy-example to show how a JT model can be translated into a CG-PDG model and then simplified the obtained CG-PDG model by removing redundancies by merging and collapsing operators. As previously stated, CG-PDGs are especially fitted to efficiently encoding context specific independencies. However, in this toy example, no context specific independencies are present, and thus, the advantages of using CG-PDGs somewhat vanish. But even when no such independencies exist, the preprocessing step of merging nodes and collapsing branches that are “almost” redundant will produce a simpler structure that is an approximation of the original. It can therefore be seen as a rather simple approach to approximate inference in CP-PDG models, but again its efficiency depends on the amount of redundancy that can be identified in the model. When redundancy is low, a more general approach to approximate inference should be explored, e.g. by simulation of non-evidence variables.

Another inference task of interest in probabilistic reasoning is abductive inference [18], which seeks for the identification of the configuration of maximal probability given some observed evidence. When the target is the subset containing all the unobserved variables we talk about *total* abduction or Most Probable Explanation (MPE), and when the target includes only a subset of the unobserved variables, then we talk about *partial* abduction or Maximum A-posteriori Probability (MAP). MPE has the same complexity as computing the a-posteriori marginal for each variable, and is solved by replacing summation by maximum as marginalization operator in the propagation algorithm. MAP is a more complex problem, because it can be exponential even in cases in which

MPE and marginal computation are easy to solve (polynomial). This is because, solving MAP requires to use both types of marginalisation operators, summation and maximisation, and they do not commute. As a consequence, larger join trees are required to solve MAP. In the case of PDGs, neither MPE nor MAP computation have been approached in the literature. In the discrete case, solving MPE should reduce to modify the marginalisation operator from summation to maximisation in the algorithm developed for computing a-posteriori marginals [3]. The case of MAP is not easy, and it will require a specific PDG structure in which maximum and summation can be done in the required order, that is, similar to the transformation method proposed in this paper in order to assure that discrete variables are placed first in the tree. Regarding the hybrid case, to our knowledge, the problem of computing MPE or MAP has not been studied in CG Bayesian networks, therefore, we set as future research the development of algorithms for computing MPE/MAP in PDGs and CG-PDGs.

## 8. Concluding remarks

In this paper we have introduced the CG-PDG model, an extension of PDGs able to represent hybrid probabilistic models with joint conditional Gaussian distribution. The new model keeps the expression power and representational efficiency of its predecessor in what concerns the discrete part, and the continuous part is also compactly represented with a number of parameters linear on the number of continuous variables once the discrete part is fixed.

We have shown how probabilistic inference can be carried out efficiently by using the concepts of inflow and outflow of nodes, and taking advantage of the recursive computations of both quantities.

We have also proved that it is always possible to obtain a CG-PDG with a number of parameters linear on the size of an equivalent JT representing a Bayesian network with CG distribution. Through an illustrative example, we have pointed out that the obtained CG-PDGs can be simplified through the merge/collapse operations, in order to speed up the belief updating task.

In the near future we plan to extend the PDGs to another hybrid model, namely the MTE (mixture of truncated exponentials) model [8], in which no structural restrictions, regarding arrangement of discrete and continuous variables, are imposed. We will also study the problem of inducing CG-PDGs from data, that so fas has been successfully addressed for discrete PDGs [17, 19].

## References

[1] J. Nielsen, A. Salmerón, Conditional Gaussian probabilistic decision graphs, in: Proceedings of the FLAIRS-23 Conference, 2010, pp. 549–554.

[2] M. Bozga, O. Maler, On the Representation of Probabilities over Structured Domains, in: Proceedings of the 11th International Conference on Computer Aided Verification, Springer, 1999, pp. 261–273.

[3] M. Jaeger, Probabilistic Decision Graphs - Combining verification and AI techniques for probabilistic inference, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 12 (2004) 19–42.

[4] J. D. Nielsen, M. Jaeger, An empirical study of efficiency and accuracy of probabilistic graphical models, in: Proceedings of theThirdEuropean Workshop on Probabilistic Graphical Models, 2006, pp. 215–222.

[5] J. D. Nielsen, R. Rumí, A. Salmerón, Supervised classification using probabilistic decision graphs, Computational Statistics and Data Analysis 53 (2009) 1299–1311.

[6] M. J. Flores, J. A. Gámez, J. D. Nielsen, The PDG-mixture model for clustering., in: Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery (DaWaK09), 2009, pp. 378–389.

[7] B. R. Cobb, P. P. Shenoy, Inference in hybrid Bayesian networks with mixtures of truncated exponentials, International Journal of Approximate Reasoning 41 (2006) 257–286.

[8] S. Moral, R. Rumí, A. Salmerón, Mixtures of truncated exponentials in hybrid Bayesian networks, in: ECSQARU'01. Lecture Notes in Artificial Intelligence, Vol. 2143, 2001, pp. 135–143.

[9] H. Langseth, T. Nielsen, R. Rumí, A. Salmerón, Parameter estimation and model selection for mixtures of truncated exponentials, International Journal of Approximate Reasoning 51 (2010) 485–498.

[10] V. Romero, R. Rumí, A. Salmerón, Learning hybrid Bayesian networks using mixtures of truncated exponentials, International Journal of Approximate Reasoning 42 (2006) 54–68.

[11] P. Shenoy, J. West, Inference in hybrid Bayesian networks using mixtures of polynomials, International Journal of Approximate Reasoning 52 (2011) 641–657.

[12] S. Lauritzen, Propagation of probabilities, means and variances in mixed graphical association models, Journal of the American Statistical Association 87 (1992) 1098–1108.

[13] S. Lauritzen, N. Wermuth, Graphical models for associations between variables, some of which are qualitative and some quantitative, The Annals of Statistics 17 (1989) 31–57.

[14] D. Koller, N. Friedman, Probabilistic graphical models. Principles and techniques, MIT Press, 2009.

[15] J. Nielsen, On unsupervised learning of probabilistic graphical models, Ph.D. thesis, Aalborg University (2007).

[16] S. L. Lauritzen, F. Jensen, Stable local computation with conditional Gaussian distributions, Statistics and Computing 11 (2001) 191–203.

[17] M. Jaeger, J. D. Nielsen, T. Silander, Learning probabilistic decision graphs, International Journal of Approximate Reasoning 42 (1-2) (2006) 84–100.

[18] J. A. Gámez, Abductive inference in Bayesian networks: A review, in: J. A. Gámez, S. Moral, A. Salmerón (Eds.), Advances in Bayesian Networks, Springer Verlag, 2004, pp. 101–120.

[19] J. D. Nielsen, R. Rumí, A. Salmerón, Structural-EM for learning PDG models from incomplete data, International Journal of Approximate Reasoning 51 (2010) 515–530.