Learning conditional Gaussian networks

Bøttcher, Susanne Gammelgaard

Publication date:
2005

Document Version
Publisher's PDF, also known as Version of record

AALBORG UNIVERSITY

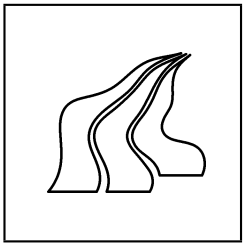
Learning conditional Gaussian networks

by

Susanne G. Bøttcher

R-2005-22 June 2005

Department of Mathematical Sciences
Aalborg University
Fredrik Bajers Vej 7 G ▪ DK - 9220 Aalborg Øst ▪ Denmark
Phone: +45 96 35 80 80 ▪ Telefax: +45 98 15 81 29
URL: http://www.math.aau.dk



ISSN 1399–2503 ▪ On-line version ISSN 1601–7811

Learning Conditional Gaussian Networks

Susanne G. Bøttcher
Aalborg University, Denmark

Abstract

This paper considers conditional Gaussian networks. The parameters in the network are learned by using conjugate Bayesian analysis. As conjugate local priors, we apply the Dirichlet distribution for discrete variables and the Gaussian-inverse gamma distribution for continuous variables, given a configuration of the discrete parents. We assume parameter independence and complete data. Further, to learn the structure of the network, the network score is deduced. We then develop a local master prior procedure, for deriving parameter priors in these networks. This procedure satisfies parameter independence, parameter modularity and likelihood equivalence. Bayes factors to be used in model search are introduced. Finally the methods derived are illustrated by a simple example.

1 Introduction

The aim of this paper is to present a method for learning the parameters and structure of a Bayesian network with discrete and continuous variables. In Heckerman, Geiger & Chickering (1995) and Geiger & Heckerman (1994), this was done for respectively discrete networks and Gaussian networks.

We define the local probability distributions such that the joint distribution of the random variables is a conditional Gaussian (CG) distribution. Therefore we do not allow discrete variables to have continuous parents, so the network factorizes into a discrete part and a mixed part. The local conjugate parameter priors are for the discrete part of the network specified as Dirichlet distributions and for the mixed part of the network as Gaussian-inverse gamma distributions, for each configuration of discrete parents.

To learn the structure, D, of a network from data, d, we use the network score, $p(d, D)$, as a measure of how probable D is. To be able to calculate this score for all possible structures, we derive a method for finding the prior distribution of the parameters in the possible structures, from marginal priors calculated from an imaginary database. The method satisfies parameter independence, parameter modularity and likelihood equivalence. If used on networks with only discrete or only continuous variables, it coincides with the methods developed in Heckerman et al. (1995) and Geiger & Heckerman (1994).

When many structures are possible, some kind of strategy to search for the structure with the highest score, has to be applied. In Cooper & Herskovits (1992), different search strategies are presented. Many of these strategies use Bayes factors for comparing the network scores of two different networks that differ by the direction of a single arrow or by the presence of a single arrow. We therefore deduce the Bayes factors for these two cases. To reduce the number of comparisons needed, we identify classes of structures for which the corresponding Bayes factor for testing an arrow between the same two variables in a network, is the same.

Finally a simple example is presented to illustrate some of the methods developed.

In this paper, we follow standard convention for drawing a Bayesian network and use shaded nodes to represent discrete variables and clear nodes to represent continuous variables.

The results in Section 2 to Section 7 are also published in Bøttcher (2001).

2 Bayesian Networks

A Bayesian network is a graphical model that encodes the joint probability distribution for a set of variables X. For terminology and theoretical aspects on graphical models, see Lauritzen (1996). In this paper we define it as consisting of

- A directed acyclic graph (DAG) $D = (V, E)$, where V is a finite set of vertices and E is a finite set of directed edges between the vertices. The DAG defines the structure of the Bayesian network.
- To each vertex $v \in V$ in the graph corresponds a random variable X_v, with state space $\mathcal{X}_v$. The set of variables associated with the graph D is then $X = (X_v)_{v \in V}$. Often we do not distinguish between a variable X_v and the corresponding vertex v.
- To each vertex v with parents $\text{pa}(v)$, there is attached a local probability distribution, $p(x_v|x_{\text{pa}(v)})$. The set of local probability distributions for all variables in the network is denoted $\mathcal{P}$.
- The possible lack of directed edges in D encodes conditional independencies between the random variables X through the factorization of the joint probability distribution,

$$p(x) = \prod_{v \in V} p(x_v|x_{\text{pa}(v)}).$$

A Bayesian network for a set of random variables X is thus the pair $(D, \mathcal{P})$. In order to specify a Bayesian network for X, we must therefore specify a DAG D and a set $\mathcal{P}$ of local probability distributions.

3 Bayesian Networks for Mixed Variables

In this paper we are interested in specifying networks for random variables X of which some are discrete and some are continuous. So we consider a DAG $D = (V, E)$ with vertices $V = \Delta \cup \Gamma$, where Δ and Γ are the sets of discrete and continuous vertices, respectively. The corresponding random variables X can then be denoted $X = (X_v)_{v \in V} = (I, Y) = ((I_\delta)_{\delta \in \Delta}, (Y_\gamma)_{\gamma \in \Gamma})$, *i.e.* we use I and Y for the sets of discrete and continuous variables, respectively. We denote the set of levels for each discrete variable $\delta \in \Delta$ as $\mathcal{I}_\delta$.

In this paper we do not allow discrete variables to have continuous parents. This *e.g.* ensures availability of exact local computation methods, see Lauritzen (1992) and Lauritzen & Jensen (2001). The joint probability distribution then factorizes as follows:

$$p(x) = p(i, y) = \prod_{\delta \in \Delta} p(i_\delta | i_{\text{pa}(\delta)}) \prod_{\gamma \in \Gamma} p(y_\gamma | i_{\text{pa}(\gamma)}, y_{\text{pa}(\gamma)}),$$

where $i_{\text{pa}(\gamma)}$ and $y_{\text{pa}(\gamma)}$ denote observations of the discrete and continuous parents respectively, *i.e.* $i_{\text{pa}(\gamma)}$ is an abbreviation of $i_{\text{pa}(\gamma) \cap \Delta}$ etc.

We see that the joint probability distribution factorizes into a purely discrete part and a mixed part. First we look at the discrete part.

3.1 The Discrete Part of the Network

We assume that the local probability distributions are unrestricted discrete distributions with

$$p(i_\delta | i_{\text{pa}(\delta)}) \geq 0 \quad \forall \quad \delta \in \Delta.$$

A way to parameterize this is to let

$$\theta_{i_\delta | i_{\text{pa}(\delta)}} = p(i_\delta | i_{\text{pa}(\delta)}, \theta_{\delta | i_{\text{pa}(\delta)}}), \tag{1}$$

where $\theta_{\delta | i_{\text{pa}(\delta)}} = (\theta_{i_\delta | i_{\text{pa}(\delta)}})_{i_\delta \in \mathcal{I}_\delta}$.

Then $\sum_{i_\delta \in \mathcal{I}_\delta} \theta_{i_\delta | i_{\text{pa}(\delta)}} = 1$ and $0 \leq \theta_{i_\delta | i_{\text{pa}(\delta)}} \leq 1$. All parameters associated with a node δ is denoted θ_δ, *i.e.* $\theta_\delta = (\theta_{\delta | i_{\text{pa}(\delta)}})_{i_{\text{pa}(\delta)} \in \mathcal{I}_{\text{pa}(\delta)}}$.

Using this parameterization, the discrete part of the joint probability distribution is given by

$$p(i | (\theta_\delta)_{\delta \in \Delta}) = \prod_{\delta \in \Delta} p(i_\delta | i_{\text{pa}(\delta)}, \theta_{\delta | i_{\text{pa}(\delta)}}).$$

3.2 The Mixed Part of the Network

Now consider the mixed part. We assume that the local probability distributions are Gaussian linear regressions on the continuous parents, with parameters depending on the configuration of the discrete parents. Let the parameters in the distribution be given by $\theta_{\gamma|i_{\mathrm{pa}(\gamma)}} = (m_{\gamma|i_{\mathrm{pa}(\gamma)}}, \beta_{\gamma|i_{\mathrm{pa}(\gamma)}}, \sigma^2_{\gamma|i_{\mathrm{pa}(\gamma)}})$. Then

$$(Y_\gamma | i_{\mathrm{pa}(\gamma)}, y_{\mathrm{pa}(\gamma)}, \theta_{\gamma|i_{\mathrm{pa}(\gamma)}}) \sim \mathcal{N}(m_{\gamma|i_{\mathrm{pa}(\gamma)}} + \beta_{\gamma|i_{\mathrm{pa}(\gamma)}} y_{\mathrm{pa}(\gamma)}\ , \sigma^2_{\gamma|i_{\mathrm{pa}(\gamma)}}), \qquad (2)$$

where $\beta_{\gamma|i_{\mathrm{pa}(\gamma)}}$ are the regression coefficients, $m_{\gamma|i_{\mathrm{pa}(\gamma)}}$ is the regression intercept, and $\sigma^2_{\gamma|i_{\mathrm{pa}(\gamma)}}$ is the conditional variance. Thus for each configuration of the discrete parents of γ, the distribution of Y_γ is Gaussian with mean and variance given as in (2). There are three special cases of the above situation, namely when γ has no discrete parents, when it has no continuous parents and when it has no parents at all. If it has no discrete parents, (2) is just the Gaussian distribution,

$$(Y_\gamma | y_{\mathrm{pa}(\gamma)}, \theta_\gamma) \sim \mathcal{N}(m_\gamma + \beta_\gamma y_{\mathrm{pa}(\gamma)}\ , \sigma^2_\gamma),$$

and $\theta_\gamma = (m_\gamma, \beta_\gamma, \sigma^2_\gamma)$. When γ has no continuous parents, we have

$$(Y_\gamma | i_{\mathrm{pa}(\gamma)}, \theta_{\gamma|i_{\mathrm{pa}(\gamma)}}) \sim \mathcal{N}(m_{\gamma|i_{\mathrm{pa}(\gamma)}}, \sigma^2_{\gamma|i_{\mathrm{pa}(\gamma)}}),$$

with $\theta_{\gamma|i_{\mathrm{pa}(\gamma)}} = (m_{\gamma|i_{\mathrm{pa}(\gamma)}}, \sigma^2_{\gamma|i_{\mathrm{pa}(\gamma)}})$, *i.e.* for each γ, the mean depends solely on $i_{\mathrm{pa}(\gamma)}$. Finally, when γ has no parents at all,

$$(Y_\gamma | \theta_\gamma) \sim \mathcal{N}(m_\gamma, \sigma^2_\gamma),$$

with $\theta_\gamma = (m_\gamma, \sigma^2_\gamma)$.

With $\theta_\gamma = (\theta_{\gamma|i_{\mathrm{pa}(\gamma)}})_{i_{\mathrm{pa}(\gamma)} \in \mathcal{I}_{\mathrm{pa}(\gamma)}}$, the mixed part of the joint distribution can be written as

$$p(y|i, (\theta_\gamma)_{\gamma\in\Gamma}) = \prod_{\gamma\in\Gamma} p(y_\gamma | i_{\mathrm{pa}(\gamma)}, y_{\mathrm{pa}(\gamma)}, \theta_{\gamma|i_{\mathrm{pa}(\gamma)}}).$$

3.3 The Joint Network

If we let $\theta = ((\theta_\delta)_{\delta\in\Delta}, (\theta_\gamma)_{\gamma\in\Gamma})$, the joint probability distribution for $X = (I, Y)$ is given by

$$p(x|\theta) = \prod_{\delta\in\Delta} p(i_\delta | i_{\mathrm{pa}(\delta)}, \theta_{\delta|i_{\mathrm{pa}(\delta)}}) \prod_{\gamma\in\Gamma} p(y_\gamma | i_{\mathrm{pa}(\gamma)}, y_{\mathrm{pa}(\gamma)}, \theta_{\gamma|i_{\mathrm{pa}(\gamma)}}). \qquad (3)$$

It can easily be shown by induction that when the local probability distributions are given as defined in (1) and (2), the joint probability distribution for X is a CG distribution with density of the form

$$p(x|\theta) = p(i, y|\theta) = p(i)|2\pi\Sigma_i|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(y - M_i)^{\mathrm{T}} \Sigma_i^{-1} (y - M_i)\}.$$

For each i, M_i is the unconditional mean, that is unconditional on continuous variables and Σ_i is the covariance matrix for all the continuous variables in the network. In Shachter & Kenley (1989) formulas for calculating Σ_i from the local probability distributions can be found.

A Bayesian network, where the joint probability distribution is a CG distribution is in the following called a *CG network*.

4 Learning the Parameters in a CG Network

When constructing a Bayesian network there is, as mentioned earlier, two things to consider, namely specifying the DAG and specifying the local probability distributions. In this section we assume that the structure of the DAG is known and the distribution type is given as in the previous section and we consider the specification of the parameters in the distributions. For this we need the concept of conjugate Bayesian analysis.

4.1 Conjugate Bayesian Analysis

There are several ways of assessing the parameters in probability distributions. An expert could specify them, or they could be estimated from data. In our approach we encode our uncertainty about the parameter θ in a *prior* distribution $p(\theta)$, use data to update this distribution, *i.e.* learn the parameter and hereby, by using Bayes' theorem, obtain the *posterior* distribution $p(\theta|\text{data})$, see DeGroot (1970).

Consider a situation with one random variable X. Let θ be the parameter to be assessed, Θ the parameter space and d a random sample of size n from the probability distribution $p(x|\theta)$. We call d our database and $x^c \in d$ a case. Then, according to Bayes' theorem,

$$p(\theta|d) = \frac{p(d|\theta)p(\theta)}{p(d)}, \qquad \theta \in \Theta, \tag{4}$$

where $p(d|\theta) = \prod_{x^c \in d} p(x^c|\theta)$ is the joint probability distribution of d, also called the likelihood of θ. Furthermore the denominator is given by

$$p(d) = \int_\Theta p(d|\theta)p(\theta)d\theta,$$

and for fixed d it may be considered as a normalizing constant. Therefore (4) can be expressed as

$$p(\theta|d) \propto p(d|\theta)p(\theta),$$

where the proportionality constant is determined by the relation $\int_\Theta p(\theta|d)d\theta = 1$.

When the prior distribution belongs to a given family of distributions and the posterior distribution, after sampling from a specific distribution, belongs to the same

family of distributions, then this family is said to be closed under sampling and called a *conjugate family* of distributions. Further, if a parameter or the distribution of a parameter has a certain property which is preserved under sampling, then this property is said to be a *conjugate property*.

In a conjugate family of distributions it is generally straightforward to calculate the posterior distribution.

4.2 Some Simplifying Properties

In the previous section we showed how to update a prior distribution for a single parameter θ. In a Bayesian network with more than one variable, we also have to look at the relationship between the different parameters for the different variables in the network. In this paper we assume that the parameters associated with one variable is independent of the parameters associated with the other variables. This assumption was introduced by Spiegelhalter & Lauritzen (1990) and we denote it *global parameter independence*. In addition to this, we will assume that the parameters are independent for each configuration of the discrete parents, which we denote as *local parameter independence*. So if the parameters have the property of global parameter independence and local parameter independence, then

$$p(\theta) = \prod_{\delta \in \Delta} \prod_{i_{\mathrm{pa}(\delta)} \in \mathcal{I}_{\mathrm{pa}(\delta)}} p(\theta_{\delta|i_{\mathrm{pa}(\delta)}}) \prod_{\gamma \in \Gamma} \prod_{i_{\mathrm{pa}(\gamma)} \in \mathcal{I}_{\mathrm{pa}(\gamma)}} p(\theta_{\gamma|i_{\mathrm{pa}(\gamma)}}), \tag{5}$$

and we will refer to (5) simply as *parameter independence*.

A consequence of parameter independence is that, for each configuration of the discrete parents, we can update the parameters in the local distributions independently. This also means that if we have *local conjugacy*, *i.e.* the distributions of $\theta_{\delta|i_{\mathrm{pa}(\delta)}}$ and $\theta_{\gamma|i_{\mathrm{pa}(\gamma)}}$ belongs to a conjugate family, then because of parameter independence, we have *global conjugacy*, *i.e.* the joint distribution of θ belongs to a conjugate family.

Further, we will assume that the database d is complete, that is, in each case it contains at least one instance of every random variable in the network. With this we can show that parameter independence is a conjugate property.

Due to the factorization (3) and the assumption of complete data,

$$\begin{aligned} p(d|\theta) &= \prod_{c \in d} p(x^c|\theta) \\ &= \prod_{c \in d} \left(\prod_{\delta \in \Delta} p(i^c_\delta | i^c_{\mathrm{pa}(\delta)}, \theta_{\delta|i_{\mathrm{pa}(\delta)}}) \prod_{\gamma \in \Gamma} p(y^c_\gamma | y^c_{\mathrm{pa}(\gamma)}, i^c_{\mathrm{pa}(\gamma)}, \theta_{\gamma|i_{\mathrm{pa}(\gamma)}}) \right), \end{aligned}$$

where i^c and y^c respectively denotes the discrete part and the continuous part of a

case x^c. Another way of writing the above equation is

$$p(d|\theta) = \prod_{\delta\in\Delta}\prod_{i_{\mathrm{pa}(\delta)}\in\mathcal{I}_{\mathrm{pa}(\delta)}}\prod_{c:i^c_{\mathrm{pa}(\delta)}=i_{\mathrm{pa}(\delta)}} p(i^c_\delta|i_{\mathrm{pa}(\delta)},\theta_{\delta|i_{\mathrm{pa}(\delta)}}) \times \prod_{\gamma\in\Gamma}\prod_{i_{\mathrm{pa}(\gamma)}\in\mathcal{I}_{\mathrm{pa}(\gamma)}}\prod_{c:i^c_{\mathrm{pa}(\gamma)}=i_{\mathrm{pa}(\gamma)}} p(y^c_\gamma|y^c_{\mathrm{pa}(\gamma)},i_{\mathrm{pa}(\gamma)},\theta_{\gamma|i_{\mathrm{pa}(\gamma)}}), \tag{6}$$

where the product over cases is split up into a product over the configurations of the discrete parents and a product over those cases, where the configuration of the discrete parents is the same as the currently processed configuration. Notice however that some of the parent configurations might not be represented in the database, in which case the product over cases with this parent configuration just adds nothing to the overall product.

By combining (5) and (6) it is seen that

$$p(\theta|d) = \prod_{\delta\in\Delta}\prod_{i_{\mathrm{pa}(\delta)}\in\mathcal{I}_{\mathrm{pa}(\delta)}} p(\theta_{\delta|i_{\mathrm{pa}(\delta)}}|d)\prod_{\gamma\in\Gamma}\prod_{i_{\mathrm{pa}(\gamma)}\in\mathcal{I}_{\mathrm{pa}(\gamma)}} p(\theta_{\gamma|i_{\mathrm{pa}(\gamma)}}|d),$$

i.e. the parameters remain independent given data. We call this property *posterior parameter independence*. In other words, the properties of local and global independence are conjugate.

Notice that the posterior distribution, $p(\theta|d)$, can be found using *batch* learning or *sequential* learning. In batch learning, $p(\theta|d)$ is found by updating $p(\theta)$ with all cases in d at the same time, *i.e.* in a batch. In sequential learning, $p(\theta)$ is updated one case at a time, using the previous posterior distribution as the prior distribution for the next case to be considered. When the database d is complete, batch learning and sequential learning leads to the same posterior distribution and the final result is independent of the order in which the cases in d are processed. It is of course also possible to process some of the cases in a batch and the rest sequentially, which could be done if *e.g.* a new case is added to an already processed database, see Bernardo & Smith (1994).

4.3 Learning in the Discrete Case

We now consider batch learning of the parameters in the discrete part of the network. Recall that the local probability distributions are unrestricted discrete distributions defined as in (1). As pointed out in the previous section we can, because of the assumption of parameter independence, find the posterior distribution of $\theta_{\delta|i_{\mathrm{pa}(\delta)}}$ for each δ and each configuration of $\mathrm{pa}(\delta)$ independently.

So given a specific configuration of $i_{\mathrm{pa}(\delta)}$, we need to find $p(\theta_{\delta|i_{\mathrm{pa}(\delta)}}|d)$. From Bayes' theorem, Equation (4), we have that

$$p(\theta_{\delta|i_{\mathrm{pa}(\delta)}}|d) \propto \prod_{c:i^c_{\mathrm{pa}(\delta)}=i_{\mathrm{pa}(\delta)}} p(i^c_\delta|i_{\mathrm{pa}(\delta)},\theta_{\delta|i_{\mathrm{pa}(\delta)}})p(\theta_{\delta|i_{\mathrm{pa}(\delta)}}). \tag{7}$$

A conjugate family for multinomial observations is the family of Dirichlet distributions. So let the prior distribution of $\theta_{\delta|i_{\mathrm{pa}(\delta)}}$ be a Dirichlet distribution $\mathcal{D}$ with hyperparameters $\alpha_{\delta|i_{\mathrm{pa}(\delta)}} = (\alpha_{i_\delta|i_{\mathrm{pa}(\delta)}})_{i_\delta \in \mathcal{I}_\delta}$, also written as

$$(\theta_{\delta|i_{\mathrm{pa}(\delta)}}|\alpha_{\delta|i_{\mathrm{pa}(\delta)}}) \sim \mathcal{D}(\alpha_{\delta|i_{\mathrm{pa}(\delta)}}). \tag{8}$$

The probability function for this Dirichlet distribution is given by

$$p(\theta_{\delta|i_{\mathrm{pa}(\delta)}}|\alpha_{\delta|i_{\mathrm{pa}(\delta)}}) = \frac{\Gamma(\alpha_{+\delta|i_{\mathrm{pa}(\delta)}})}{\prod_{i_\delta \in \mathcal{I}_\delta} \Gamma(\alpha_{i_\delta|i_{\mathrm{pa}(\delta)}})} \prod_{i_\delta \in \mathcal{I}_\delta} (\theta_{i_\delta|i_{\mathrm{pa}(\delta)}})^{\alpha_{i_\delta|i_{\mathrm{pa}(\delta)}}-1},$$

where $\alpha_{+\delta|i_{\mathrm{pa}(\delta)}} = \sum_{i_\delta \in \mathcal{I}_\delta} \alpha_{i_\delta|i_{\mathrm{pa}(\delta)}}$ and $\Gamma(\cdot)$ is the gamma function. Because of notational convenience, we do not in what follows write the hyperparameters explicitly in the conditioning.

It then follows from (7) and (8) that the posterior distribution is given as

$$(\theta_{\delta|i_{\mathrm{pa}(\delta)}}|d) \sim \mathcal{D}(\alpha_{\delta|i_{\mathrm{pa}(\delta)}} + n_{\delta|i_{\mathrm{pa}(\delta)}}),$$

where the vector $n_{\delta|i_{\mathrm{pa}(\delta)}} = (n_{i_\delta|i_{\mathrm{pa}(\delta)}})_{i_\delta \in \mathcal{I}_\delta}$, also called the counts, denotes the number of observations in d where δ and $\mathrm{pa}(\delta)$ have that specific configuration. Notice that, for at given parent configuration, the number of observations in a batch, $|b|$, is the same as $n_{+\delta|i_{\mathrm{pa}(\delta)}}$, where $n_{+\delta|i_{\mathrm{pa}(\delta)}} = \sum_{i_\delta \in \mathcal{I}_\delta} n_{i_\delta|i_{\mathrm{pa}(\delta)}}$.

Because of parameter independence, the joint prior distribution of all the parameters for the discrete variables in the network, is given by the product of the local parameter priors.

The above learning procedure can also be used for sequential learning by applying the above formulas one case at a time, using the previous posterior distribution as the prior distribution for the next case to be processed.

4.4 Learning in the Mixed Case

In the mixed case we write the local probability distributions as

$$(Y_\gamma|i_{\mathrm{pa}(\gamma)}, y_{\mathrm{pa}(\gamma)}, \theta_{\gamma|i_{\mathrm{pa}(\gamma)}}) \sim \mathcal{N}(z_{\mathrm{pa}(\gamma)}(m_{\gamma|i_{\mathrm{pa}(\gamma)}}, \beta_{\gamma|i_{\mathrm{pa}(\gamma)}})^{\mathrm{T}}, \sigma^2_{\gamma|i_{\mathrm{pa}(\gamma)}}),$$

where $z_{\mathrm{pa}(\gamma)} = (1, y_{\mathrm{pa}(\gamma)})$. This vector has dimension $k+1$, where k is the number of continuous parents to γ.

As in the discrete case we can because of parameter independence update the parameters for each γ and each configuration of the discrete parents independently. By Bayes' theorem,

$$p(\theta_{\gamma|i_{\mathrm{pa}(\gamma)}}|d) \propto \prod_{c: i^c_{\mathrm{pa}(\gamma)} = i_{\mathrm{pa}(\gamma)}} p(y^c_\gamma|y^c_{\mathrm{pa}(\gamma)}, i_{\mathrm{pa}(\gamma)}, \theta_{\gamma|i_{\mathrm{pa}(\gamma)}}) p(\theta_{\gamma|i_{\mathrm{pa}(\gamma)}}).$$

We now join all the observations y^c_γ for which $i^c_{\text{pa}(\gamma)} = i_{\text{pa}(\gamma)}$ in a vector y^b_γ, *i.e.* $y^b_\gamma = (y^c_\gamma)_{i^c_{\text{pa}(\gamma)}=i_{\text{pa}(\gamma)}}$. The same is done with the observations of the continuous parents of γ, *i.e.* $y^b_{\text{pa}(\gamma)} = (y^c_{\text{pa}(\gamma)})_{i^c_{\text{pa}(\gamma)}=i_{\text{pa}(\gamma)}}$. As the observations in d are independent, $p(y^b_\gamma | y^b_{\text{pa}(\gamma)}, i_{\text{pa}(\gamma)}, \theta_{\gamma|i_{\text{pa}(\gamma)}})$ is the likelihood function for a multivariate normal distribution with mean vector $z^b_{\text{pa}(\gamma)}(m_{\gamma|i_{\text{pa}(\gamma)}}, \beta_{\gamma|i_{\text{pa}(\gamma)}})^{\mathrm{T}}$ and covariance matrix $\sigma^2_{\gamma|i_{\text{pa}(\gamma)}} I$, where I is the identity matrix and $z^b_{\text{pa}(\gamma)}$ is defined through $y^b_{\text{pa}(\gamma)}$.

The posterior distribution of $\theta_{\gamma|i_{\text{pa}(\gamma)}}$ can now be written as

$$p(\theta_{\gamma|i_{\text{pa}(\gamma)}}|d) \quad \propto \quad p(y^b_\gamma | y^b_{\text{pa}(\gamma)}, i_{\text{pa}(\gamma)}, \theta_{\gamma|i_{\text{pa}(\gamma)}}) p(\theta_{\gamma|i_{\text{pa}(\gamma)}}).$$

A standard conjugate family for these observations is the family of Gaussian-inverse gamma distributions. Let the prior joint distribution of $(m_{\gamma|i_{\text{pa}(\gamma)}}, \beta_{\gamma|i_{\text{pa}(\gamma)}})$ and $\sigma^2_{\gamma|i_{\text{pa}(\gamma)}}$ be as follows.

$$\begin{aligned}
(m_{\gamma|i_{\text{pa}(\gamma)}}, \beta_{\gamma|i_{\text{pa}(\gamma)}} | \sigma^2_{\gamma|i_{\text{pa}(\gamma)}}) &\sim \mathcal{N}_{k+1}(\mu_{\gamma|i_{\text{pa}(\gamma)}}, \sigma^2_{\gamma|i_{\text{pa}(\gamma)}} \tau^{-1}_{\gamma|i_{\text{pa}(\gamma)}}) \\
(\sigma^2_{\gamma|i_{\text{pa}(\gamma)}}) &\sim \mathcal{I}\Gamma\left(\frac{\rho_{\gamma|i_{\text{pa}(\gamma)}}}{2}, \frac{\phi_{\gamma|i_{\text{pa}(\gamma)}}}{2}\right).
\end{aligned}$$

The posterior distribution is then

$$\begin{aligned}
(m_{\gamma|i_{\text{pa}(\gamma)}}, \beta_{\gamma|i_{\text{pa}(\gamma)}} | \sigma^2_{\gamma|i_{\text{pa}(\gamma)}}, d) &\sim \mathcal{N}_{k+1}(\mu'_{\gamma|i_{\text{pa}(\gamma)}}, \sigma^2_{\gamma|i_{\text{pa}(\gamma)}} (\tau^{-1}_{\gamma|i_{\text{pa}(\gamma)}})') \\
(\sigma^2_{\gamma|i_{\text{pa}(\gamma)}}|d) &\sim \mathcal{I}\Gamma\left(\frac{\rho'_{\gamma|i_{\text{pa}(\gamma)}}}{2}, \frac{\phi'_{\gamma|i_{\text{pa}(\gamma)}}}{2}\right),
\end{aligned}$$

where

$$\begin{aligned}
\tau'_{\gamma|i_{\text{pa}(\gamma)}} &= \tau_{\gamma|i_{\text{pa}(\gamma)}} + (z^b_{\text{pa}(\gamma)})^{\mathrm{T}} z^b_{\text{pa}(\gamma)} \\
\mu'_{\gamma|i_{\text{pa}(\gamma)}} &= (\tau'_{\gamma|i_{\text{pa}(\gamma)}})^{-1} (\tau_{\gamma|i_{\text{pa}(\gamma)}} \mu_{\gamma|i_{\text{pa}(\gamma)}} + (z^b_{\text{pa}(\gamma)})^{\mathrm{T}} y^b_\gamma) \\
\rho'_{\gamma|i_{\text{pa}(\gamma)}} &= \rho_{\gamma|i_{\text{pa}(\gamma)}} + |b| \\
\phi'_{\gamma|i_{\text{pa}(\gamma)}} &= \phi_{\gamma|i_{\text{pa}(\gamma)}} + (y^b_\gamma - z^b_{\text{pa}(\gamma)} \mu'_{\gamma|i_{\text{pa}(\gamma)}})^{\mathrm{T}} y^b_\gamma \\
&\quad + (\mu_{\gamma|i_{\text{pa}(\gamma)}} - \mu'_{\gamma|i_{\text{pa}(\gamma)}})^{\mathrm{T}} \tau_{\gamma|i_{\text{pa}(\gamma)}} \mu_{\gamma|i_{\text{pa}(\gamma)}},
\end{aligned}$$

where $|b|$ denotes the number of observations in b.

As for the discrete variables, we can with these formulas also use the sequential approach and update the parameters one case at a time.

Further, because of parameter independence, the joint prior distribution is given as the product of the local prior distributions for all parameters in the network.

5 Learning the Structure of a CG Network

In this section we consider how to learn the structure of a CG network.

5.1 The Network Score

There are basically two ways of determining which DAG should represent the conditional independencies between a set of random variables. First, if the relations between the variables are well understood by an expert, then he could specify the DAG, using a causal interpretation of the arrows. Second, we could learn the DAG from data. That is, we could find out how well a DAG D represents the conditional independencies, by measuring how probable D is, given that we have observed data d. Different approaches use different measures. An often used measure is the posterior probability of the DAG, $p(D|d)$, which from Bayes' theorem is given by

$$p(D|d) \propto p(d|D)p(D),$$

where $p(d|D)$ is the likelihood of D and $p(D)$ is the prior probability. As the normalizing constant does not depend upon structure, another measure, which gives the relative probability, is

$$p(D, d) = p(d|D)p(D).$$

We refer to the above measures as *network scores*. So learning the DAG from data, we can in principle first calculate the network scores for all possible DAGs and then select the DAG with the highest network score. If many DAGs are possible, it is computationally infeasible to calculate the network score for all these DAGs. In this situation it is necessary to use some kind of search strategy to find the DAG with the highest score, see *e.g.* Cooper & Herskovits (1992).

In some cases it can be more accurate to average over the possible DAGs for prediction, instead of just selecting a single DAG. So if x is the quantity we are interested in, we can use the weighted average,

$$p(x|d) = \sum_{D \in DAG} p(x|d, D)p(D|d),$$

where DAG is the set of all DAGs and $p(D|d)$ is the weight.

Again, if many DAGs are possible, this sum is to heavy to compute, so instead, by using a search strategy, we can find a few DAGs with high score and average over these.

5.2 The Network Score for a CG Network

In order to calculate the network score for a specific DAG D, we need to know the prior probability and the likelihood of the DAG. For simplicity, we could for

example choose to let all DAGs be equally likely, then

$$p(D|d) \propto p(d|D).$$

In a CG network, the likelihood of the DAG D is given by

$$\begin{aligned} p(d|D) &= \int_{\theta\in\Theta} p(d|\theta, D)p(\theta|D)d\theta \\ &= \prod_{\delta\in\Delta}\prod_{i_{\mathrm{pa}(\delta)}\in\mathcal{I}_{\mathrm{pa}(\delta)}}\int\prod_{c:i^c_{\mathrm{pa}(\delta)}=i_{\mathrm{pa}(\delta)}} p(i^c_\delta|i_{\mathrm{pa}(\delta)}, \theta_{\delta|i_{\mathrm{pa}(\delta)}}, D)p(\theta_{\delta|i_{\mathrm{pa}(\delta)}}|D)d\theta_{\delta|i_{\mathrm{pa}(\delta)}} \\ &\times \prod_{\gamma\in\Gamma}\prod_{i_{\mathrm{pa}(\gamma)}\in\mathcal{I}_{\mathrm{pa}(\gamma)}}\int\prod_{c:i^c_{\mathrm{pa}(\gamma)}=i_{\mathrm{pa}(\gamma)}} p(y^c_\gamma|y^c_{\mathrm{pa}(\gamma)}, i_{\mathrm{pa}(\gamma)}, \theta_{\gamma|i_{\mathrm{pa}(\gamma)}}, D)p(\theta_{\gamma|i_{\mathrm{pa}(\gamma)}}|D)d\theta_{\gamma|i_{\mathrm{pa}(\gamma)}}. \end{aligned}$$

Again we see that we can consider the problem for the discrete part and the mixed part of the network separately.

The discrete part is from the formulas in Section 4.3 found to be

$$\prod_{\delta\in\Delta}\prod_{i_{\mathrm{pa}(\delta)}\in\mathcal{I}_{\mathrm{pa}(\delta)}}\frac{\Gamma(\alpha_{+_\delta|i_{\mathrm{pa}(\delta)}})}{\Gamma(\alpha_{+_\delta|i_{\mathrm{pa}(\delta)}}+n_{+_\delta|i_{\mathrm{pa}(\delta)}})}\prod_{i_\delta\in\mathcal{I}_\delta}\frac{\Gamma(\alpha_{i_\delta|i_{\mathrm{pa}(\delta)}}+n_{i_\delta|i_{\mathrm{pa}(\delta)}})}{\Gamma(\alpha_{i_\delta|i_{\mathrm{pa}(\delta)}})}.$$

In the mixed part of the network, the local marginal likelihoods are non-central t distributions with $\rho_{\gamma|i_{\mathrm{pa}(\gamma)}}$ degrees of freedom, location vector $z^b_{\mathrm{pa}(\gamma)}\mu_{\gamma|i_{\mathrm{pa}(\gamma)}}$ and scale parameter $s_{\gamma|i_{\mathrm{pa}(\gamma)}} = \frac{\phi_{\gamma|i_{\mathrm{pa}(\gamma)}}}{\rho_{\gamma|i_{\mathrm{pa}(\gamma)}}}(I + (z^b_{\mathrm{pa}(\gamma)})\tau^{-1}_{\gamma|i_{\mathrm{pa}(\gamma)}}(z^b_{\mathrm{pa}(\gamma)})^{\mathrm{T}})$. The index b is defined as in Section 4.4.

So the mixed part is given by

$$\prod_{\gamma\in\Gamma}\prod_{i_{\mathrm{pa}(\gamma)}\in\mathcal{I}_{\mathrm{pa}(\gamma)}}\frac{\Gamma((\rho_{\gamma|i_{\mathrm{pa}(\gamma)}}+|b|)/2)}{\Gamma(\rho_{\gamma|i_{\mathrm{pa}(\gamma)}}/2)[\det(\rho_{\gamma|i_{\mathrm{pa}(\gamma)}}s_{\gamma|i_{\mathrm{pa}(\gamma)}}\pi)]^{\frac{1}{2}}} \times$$

$$\left[1+\frac{1}{\rho_{\gamma|i_{\mathrm{pa}(\gamma)}}}(y^b_\gamma - z^b_{\mathrm{pa}(\gamma)}\mu_{\gamma|i_{\mathrm{pa}(\gamma)}})s^{-1}_{\gamma|i_{\mathrm{pa}(\gamma)}}(y^b_\gamma - z^b_{\mathrm{pa}(\gamma)}\mu_{\gamma|i_{\mathrm{pa}(\gamma)}})^{\mathrm{T}}\right]^{\frac{-(\rho_{\gamma|i_{\mathrm{pa}(\gamma)}}+|b|)}{2}}.$$

The network score for a CG network is thus the product of the prior probability for the DAG D, the term for the discrete part and the term for the mixed part. Notice that the network score has the property that it factorizes into a product over terms involving only one node and its parents. This property is called *decomposability*.

To evaluate which DAG or possible several DAGs that represent the conditional independencies in a Bayesian network well, we want to find the DAG or DAGs with the highest network scores. To calculate these scores, we must specify the local probability distributions and the local prior distributions for the parameters for each network under evaluation. In the next section, a method for doing this is developed.

6 The Master Prior Procedure

The papers Heckerman et al. (1995) and Geiger & Heckerman (1994) develops a method for finding the prior distributions for the parameters in respectively the purely discrete case and the purely continuous case. The work is based on principles of likelihood equivalence, parameter modularity, and parameter independence. It leads to a method where the parameter priors for all possible networks are deduced from one joint prior distribution, in the following called a *master prior* distribution.

In this paper we will build on this idea, which can be used on networks with mixed variables. We will therefore in the following describe their method for the pure cases.

6.1 The Master Prior in the Discrete Case

In the purely discrete case, or the discrete part of a mixed network, the following is a well known classical result.

Let A be a subset of Δ and let $B = \Delta \setminus A$. Let the discrete variables i have the joint distribution

$$p(i|\Psi) = \Psi_i.$$

Notice here, that the set $\Psi = (\Psi_i)_{i\in\mathcal{I}}$ contains the parameters for the joint distribution, contrary to θ in Section 3, which contains the parameters for the conditional local distributions.

In the following we use the notation $z_{i_A} = \sum_{j:j_A=i_A} z_j$, where z is any parameter. Then the marginal distribution of i_A is given by

$$p(i_A|\Psi) = \Psi_{i_A},$$

and the conditional distribution of i_B given i_A is

$$p(i_B|i_A, \Psi) = \frac{\Psi_i}{\Psi_{i_A}} = \Psi_{i_B|i_A}.$$

Further if the joint prior distribution for the parameters Ψ is Dirichlet, that is

$$(\Psi) \sim \mathcal{D}(\alpha),$$

where $\alpha = (\alpha_i)_{i\in\mathcal{I}}$, then the marginal distribution of Ψ_A is Dirichlet, *i.e.*

$$(\Psi_A) \sim \mathcal{D}(\alpha_A),$$

with $\alpha_A = (\alpha_{i_A})_{i_A\in\mathcal{I}_A}$. The conditional distribution of $\Psi_{B|i_A}$ is

$$(\Psi_{B|i_A}) \sim \mathcal{D}(\alpha_{B|i_A}),$$

with $\alpha_{B|i_A} = (\alpha_{i_B|i_A})_{i_B \in \mathcal{I}_B}$ and $\alpha_{i_B|i_A} = \alpha_i$. Furthermore the parameters are independent, that is

$$p(\Psi) = \prod_{i_A \in \mathcal{I}_A} p(\Psi_{B|i_A}) p(\Psi_A). \tag{9}$$

From the above result we see, that for each possible parent/child relationship, we can find the marginal parameter prior $p(\Psi_{\delta \cup \text{pa}(\delta)})$. Further, from this marginal distribution we can, for each configuration of the parents, find the conditional local prior distribution $p(\Psi_{\delta|i_{\text{pa}(\delta)}})$. Notice that $\Psi_{\delta|i_{\text{pa}(\delta)}} = \theta_{\delta|i_{\text{pa}(\delta)}}$, where $\theta_{\delta|i_{\text{pa}(\delta)}}$ was specified for the conditional distributions in Section (3.1). Further, because of parameter independence, given by (9), we can find the joint parameter prior for any network as the product of the local priors involved.

To use this method, we must therefore specify the joint Dirichlet distribution, *i.e.* the master Dirichlet prior. This was first done in Heckerman et al. (1995) and here we follow their method. We start by specifying a prior Bayesian network $(D, \mathcal{P})$. From this we calculate the joint distribution $p(i|\Psi) = \Psi_i$. To specify a master Dirichlet distribution, we must specify the parameters $\alpha = (\alpha_{i_\delta})_{i \in \mathcal{I}}$ and for this we use the following relation for the Dirichlet distribution,

$$p(i) = \mathbb{E}(\Psi_i) = \frac{\alpha_i}{n},$$

with $n = \sum_{i \in \mathcal{I}} \alpha_i$. Now we let the probabilities in the prior network be an estimate of $\mathbb{E}(\Psi_i)$, so we only need to determine n in order to calculate the parameters α_i. We determine n by using the notion of an imaginary database. We imagine that we have a database of cases, from which we from total ignorance have updated the distribution of Ψ. The sample size of this imaginary database is thus n. Therefore we refer to the estimate of n as the *imaginary sample size* and it expresses how much confidence we have in the dependency structure expressed in the prior network.

6.2 The Master Prior in the Gaussian Case

For the Gaussian case, the following result is used, see *e.g.* Dawid & Lauritzen (1993). Let A be a subset of Γ and let $B = \Gamma \setminus A$. If

$$(y|m, \Sigma) \sim \mathcal{N}(m, \Sigma),$$

then

$$(y_A|m, \Sigma) \sim \mathcal{N}(m_A, \Sigma_{AA})$$

and

$$(y_B|y_A, m_{B|A}, \beta_{B|A}, \Sigma_{B|A}) \sim \mathcal{N}(m_{B|A} + \beta_{B|A} y_A, \Sigma_{B|A}),$$

where

$$\Sigma = \begin{pmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{pmatrix}, \; \Sigma_{B|A} = \Sigma_{BB} - \Sigma_{BA}\Sigma_{AA}^{-1}\Sigma_{AB},$$

$$m_{B|A} = m_B - \beta_{B|A} m_A \quad \text{and} \quad \beta_{B|A} = \Sigma_{BA}\Sigma_{AA}^{-1}.$$

Further, if

$$(m|\Sigma) \sim \mathcal{N}(\mu, \frac{1}{\nu}\Sigma) \text{ and } (\Sigma) \sim \mathcal{I}W(\rho, \Phi),$$

where the scale matrix Φ is partitioned as Σ, then

- $(m_A|\Sigma_{AA}) \sim \mathcal{N}(\mu_A, \frac{1}{\nu}\Sigma_{AA})$
- $(\Sigma_{AA}) \sim \mathcal{I}W(\rho, \Phi_{AA})$
- $(\Sigma_{B|A}) \sim \mathcal{I}W(\rho + |A|, \Phi_{B|A})$
- $(m_{B|A}, \beta_{B|A}|\Sigma_{B|A}) \sim \mathcal{N}(\mu_{B|A}, \Sigma_{B|A} \otimes \tau_{B|A}^{-1})$
- $m_A, \Sigma_{AA} \perp\!\!\!\perp m_{B|A}, \beta_{B|A}\Sigma_{B|A}$

where

$$\mu_{B|A} = (\mu_B - \Phi_{BA}\Phi_{AA}^{-1}\mu_A, \Phi_{BA}\Phi_{AA}^{-1})$$

and

$$\tau_{B|A}^{-1} = \begin{pmatrix} \frac{1}{\nu} + \mu_A^{\mathrm{T}}\Phi_{AA}^{-1}\mu_A & -\mu_A^{\mathrm{T}}\Phi_{AA}^{-1} \\ -\Phi_{AA}^{-1}\mu_A & \Phi_{AA}^{-1} \end{pmatrix},$$

and $\otimes$ denotes the Kronecker product. Notice that the dimension of $\mu_{B|A}$ is given as $(|B|, |B| \times |A|)$.

As in the discrete case, this result shows us how to deduce the local probability distributions and the local prior distributions from the joint distributions. Further, because of parameter independence, the joint parameter prior for any Gaussian network can be specified as the product of the local priors. Notice that the parameters found here for a node given its parents, coincides with the parameters specified in Section 3.2.

Before we show how to construct the master prior, we need the following result. The Gaussian-inverse Wishart prior is conjugate to observations from a Gaussian distribution (DeGroot 1970). So let the probability distribution and the prior distribution be given as above. Then, given the database $d = \{y^1, \ldots, y^n\}$, the posterior distributions are

$$(m|\Sigma, d) \sim \mathcal{N}(\mu', \frac{1}{\nu'}\Sigma) \text{ and } (\Sigma|d) \sim \mathcal{I}W(\rho', \Phi'),$$

where

$$\begin{aligned} \nu' &= \nu + n, \\ \mu' &= \frac{\nu\mu + n\overline{y}}{\nu + n}, \\ \rho' &= \rho + n, \\ \Phi' &= \Phi + ssd + \frac{\nu n}{\nu + n}(\mu - \overline{y})(\mu - \overline{y})^{\mathrm{T}}, \end{aligned} \tag{10}$$

with

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \quad \text{and} \quad ssd = \sum_{i=1}^{n}(y_i - \overline{y})(y_i - \overline{y})^{\mathrm{T}}.$$

From these updating formulas we see that ν' and ρ' are updated with the number of cases in the database. Further μ' is a weighted average of the prior mean and the sample mean, each weighted by their sample sizes. Finally Φ is updated with the ssd, which expresses how much each observation differs from the sample mean, and an expression for how much the prior mean differs from the sample mean.

To specify the master prior, we need to specify the four parameters ν, μ, ρ and Φ. As for the discrete variables we start by specifying a prior Bayesian network, $(D, \mathcal{P})$. From this, a prior joint probability distribution $p(y|m, \Sigma) = \mathcal{N}(m, \Sigma)$ can be deduced. Now imagine that the mean m and the variance Σ were calculated from an imaginary database, so that they actually are the sample mean and the sample variance. Further, assume that before this imaginary database was observed, we were totally ignorant about the parameters. The formulas in (10) can now be used to "update" the parameters on the basis of the imaginary database. As we have not seen any cases before, ν and ρ are estimated by the size of the imaginary database. Further

$$\mu = m \quad \text{and} \quad \Phi = ssd = (\nu - 1)\Sigma.$$

In Geiger & Heckerman (1994), μ and Φ are found in a slightly different way. They use the fact that the marginal likelihood $p(y)$ is a multivariate non-central t distribution with ρ degrees of freedom, location vector μ and scale matrix $S = \frac{\nu+1}{\nu\rho}\Phi$. Now the mean and covariance matrix in the t distribution is given by

$$\mathbb{E}(y) = \mu \quad \text{and} \quad \mathrm{Cov}(y) = \frac{\rho}{\rho - 2}S.$$

They then let the mean and covariance matrix from the prior network estimate the mean and covariance matrix in the t distribution, which implies that

$$\mu = m \quad \text{and} \quad \Phi = \frac{\nu(\rho - 2)}{\nu + 1}\Sigma.$$

Experimental results have not shown noticeable differences between the two approaches.

6.3 Properties of the Master Prior Procedure

The method for finding prior parameter distributions described in the previous section has some properties, which we will describe here. In this section we use Ψ as a parameter defined for a joint distribution, *i.e.* Ψ can be the parameter for the discrete variables or in the continuous case, $\Psi = (m, \Sigma)$.

Clearly a consequence of using the above method is that the parameters are independent. Further it can be seen, that if a node v has the same parents in two DAGs D and D^*, then

$$p(\Psi_{v|\text{pa}(v)}|D) = p(\Psi_{v|\text{pa}(v)}|D^*).$$

This property is referred to as *parameter modularity*. Now both the discrete and the Gaussian distribution has the property that if the joint probability distribution $p(x)$ can be factorized according to a DAG D, then it can also be factorized according to all other DAGs, which represents the same set of conditional independencies as D. A set of DAGs, D^e, which represents the same independence constraints is referred to as *independence equivalent* DAGs. So let D and D^* be independence equivalent DAGs, then

$$p(x|\Psi, D) = p(x|\Psi, D^*).$$

This means, that from observations alone we can not distinguish between different DAGs in an equivalence class. In the papers Heckerman et al. (1995) and Geiger & Heckerman (1994) it is for respectively the discrete and the Gaussian case shown, that when using the master prior procedure for the construction of parameter priors, the marginal likelihood for data is also the same for independence equivalent networks, *i.e.*

$$p(d|D) = p(d|D^*).$$

This equivalence is referred to as *likelihood equivalence*. Note that likelihood equivalence imply that if D and D^* are independence equivalent networks, then they have the same joint prior for the parameters, *i.e.*

$$p(\Psi|D) = p(\Psi|D^*).$$

7 Local Masters for Mixed Networks

In this section we will show how to specify prior distributions for the parameters in a CG network. In the mixed case, the marginal of a CG distribution is not always a CG distribution. In fact it is only a CG distribution if we marginalize over continuous variables or if we marginalize over a set B of discrete variable, where $(B \perp\!\!\!\perp \Gamma) \mid (\Delta \setminus B)$, see Frydenberg (1990). Consider the following example. We have a network of two variables, i and y, and the joint distribution is given by

$$p(i, y) = p(i)\mathcal{N}(m_i, \sigma_i^2).$$

Then the marginal distribution of y is given as a mixture of normal distributions

$$p(y) = \sum_{i \in \mathcal{I}} p(i)\mathcal{N}(m_i, \sigma_i^2),$$

so there is no simple way of using this directly for finding the local priors.

7.1 The Suggested Solution

The suggested solution is very similar to the solution for the pure cases. We start by specifying a prior Bayesian network $(D, \mathcal{P})$ and calculate the joint probability distribution

$$p(i, y|H) = p(i|\Psi)\mathcal{N}(m_i, \Sigma_i),$$

with $H = (\Psi, (m_i)_{i\in\mathcal{I}}, (\Sigma_i)_{i\in\mathcal{I}})$. So from the conditional parameters in the local distributions in the prior network, we calculate the parameters for the joint distribution. Then we translate this prior network into an imaginary database, with imaginary sample size n. From the probabilities in the discrete part of the network, we can, as in the pure discrete case, calculate α_i for all configurations of i. Now α_i represents how many times we have observed $I = i$ in the imaginary database. We can assume that each time we have observed the discrete variables I, we have observed the continuous variables Y and therefore set $\nu_i = \rho_i = \alpha_i$. Now for each configuration of i, we let m_i be the sample mean in the imaginary database, and Σ_i the sample variance. Further, as for the pure Gaussian case, we use $m_i = \mu_i$ and $\Phi_i = (\nu_i - 1)\Sigma_i$. However, for Φ_i to be positive, ν_i has to larger than 1, for all configurations i and this has an impact on how small we can choose n to be, as $n = \sum_i \nu_i$. If the number of discrete variables is large, and/or the number of configurations of the discrete variables is large, then we might have to let n be larger than the value, that really reflects our confidence in the prior network. For these situations it might therefore be better to *e.g.* let $\Phi_i = \nu_i\Sigma_i$ as we then can choose the value of n any way we want. Or, we can just choose ν_i and ρ_i independently of n.

All the parameters needed to define the joint prior distributions for the parameters are now specified, so

$$\begin{aligned} p(\Psi) &= \mathcal{D}(\alpha), \\ p(M_i|\Sigma_i) &= \mathcal{N}(\mu_i, \frac{1}{\nu_i}\Sigma_i), \\ p(\Sigma_i) &= \mathcal{I}W(\rho_i, \Phi_i). \end{aligned}$$

But we can not use these distributions to derive priors for other networks, so instead we use the imaginary database to derive local master distributions.

Let, for each family $A = v \cup \text{pa}(v)$, the marginal CG distribution of X_a given H_A be given by

$$(X_A|H_A) \sim CG(\Psi_{i_{A\cap\Delta}}, m_{A\cap\Gamma|i_{A\cap\Delta}}, \Sigma_{A\cap\Gamma|i_{A\cap\Delta}}).$$

Then we suggest that the marginal prior distributions, also called the *local masters*, are found in the following way:

Let, for any variable z, $z_{i_{A\cap\Delta}} = \sum_{j:j_{A\cap\Delta}=i_{A\cap\Delta}} z_j$. Then

$$\begin{aligned}(\Psi_{A\cap\Delta}) &\sim \mathcal{D}(\alpha_{A\cap\Delta}),\\ (\Sigma_{A\cap\Gamma|i_{A\cap\Delta}}) &\sim \mathcal{IW}(\rho_{i_{A\cap\Delta}}, (\tilde{\Phi}_{A\cap\Gamma|i_{A\cap\Delta}}),\\ (m_{A\cap\Gamma|i_{A\cap\Delta}}|\Sigma_{A\cap\Gamma|i_{A\cap\Delta}}) &\sim \mathcal{N}(\overline{\mu}_{A\cap\Gamma|i_{A\cap\Delta}}, \frac{1}{\nu_{i_{A\cap\Delta}}}\Sigma_{A\cap\Gamma|i_{A\cap\Delta}}),\end{aligned}$$

where

$$\overline{\mu}_{i_{A\cap\Delta}} = \frac{(\sum_{j:j_{A\cap\Delta}=i_{A\cap\Delta}} \mu_j\nu_j)}{\nu_{A\cap\Delta}},$$

and

$$\tilde{\Phi}_{i_{A\cap\Delta}} = \Phi_{i_{A\cap\Delta}} + \sum_{j:j_{A\cap\Delta}=i_{A\cap\Delta}} \nu_j(\mu_j - \overline{\mu}_{i_{A\cap\Delta}})(\mu_j - \overline{\mu}_{i_{A\cap\Delta}})^{\mathrm{T}}.$$

The equations in the above result are well known from the analysis of variance theory, see *e.g.* Seber (1984). The marginal mean is found as a weighted average of the mean in every group, where a group here is given as a configuration of the discrete parents we marginalize over. The weights are the number of observations in each group. The marginal ssd is given as the within group variation plus the between group variation. Notice that with this method, it is possible to specify mixed networks, where the mean in the mixed part of the network depends on the discrete parents, but the variance does not.

From the local masters we can now, by conditioning as in the pure cases, derive the local priors needed to specify the prior parameter distribution for a CG network. So the only difference between the master procedure and the local master procedure is in the way the marginal distributions are found.

7.2 Properties of the Local Master Procedure

The local master procedure coincides with the master procedure in the pure cases. Further, the properties of the local master procedure in the mixed case, are the same as of the master prior procedure in the pure cases.

Parameter independence and parameter modularity follows immediately from the definition of the procedure. To show likelihood equivalence, we need the following result from Chickering (1995). Let D and D^* be two DAGs and let R_{D,D^*} be the set of edges by which D and D^* differ in directionality. Then, D and D^* are independence equivalent if and only if there exists a sequence of $|R_{D,D^*}|$ distinct arc reversals applied to D with the following properties:

- After each reversal, the resulting network structure is a DAG, *i.e.* it contains no directed cycles and it is independence equivalent to D^*.
- After all reversals, the resulting DAG is identical to D^*.

- If $w \rightarrow v$ is the next arc to be reversed in the current DAG, then w and v have the same parents in both DAGs, with the exception that w is also a parent of v in D.

Note that as we only reverse $|R_{D,D^*}|$ distinct arcs, we only reverse arcs in R_{D,D^*}. For mixed networks this means that we only reverse arcs between discrete variables or between continuous variables, as the only arcs that can differ in directionality are these. So we can use the above result for mixed networks.

From the above we see that we can show likelihood equivalence by showing that $p(d|D) = p(d|D^*)$ for two independence equivalent DAGs D and D^* that differ only by the direction of a single arc. As $p(x|H, D) = p(x|H, D^*)$ in CG networks, we can show likelihood equivalence by showing that $p(H|D) = p(H|D^*)$.

In the following let $v \rightarrow w$ in D and $w \rightarrow v$ in D^*. Further let ∇ be the set of common discrete and continuous parents for v and w. Of course, if v and w are discrete variables, then ∇ only contains discrete variables. The relation between $p(H|D)$ and $p(H|D^*)$ is given by:

$$\begin{aligned} \frac{p(H|D)}{p(H|D^*)} &= \frac{p(H_{v|w\cup\nabla}, D)p(H_{w|\nabla}, D)}{p(H_{w|v\cup\nabla}, D^*)p(H_{v|\nabla}, D^*)} \\ &= \frac{p(H_{v\cup w|\nabla}, D)}{p(H_{v\cup w|\nabla}, D^*)}. \end{aligned} \tag{11}$$

When using the local master procedure, the terms in (11) are equal. This is evident, as we find the conditional priors from distributions over families A, in this case $A = v \cup w \cup \nabla$, which is the same for both networks. Therefore likelihood equivalence follows.

8 Model Search

In the search for Bayesian networks with high network score, we can, in theory, calculate the network score for all possible DAGs and then choose the DAG or DAGs with the highest score.

In Robinson (1977), a recursive formula for the number of possible DAGs that contains n nodes, is found to be

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \left(\begin{array}{c} n \\ i \end{array} \right) 2^{i(n-i)} f(n-i),$$

where $\left(\begin{array}{c} n \\ i \end{array} \right)$ are the binomial coefficient. As we in mixed networks do not allow discrete nodes to have continuous parents, the number of possible mixed DAGs is

given by

$$f(|\Delta|, |\Gamma)|) = f(|\Delta|) \times f(|\Gamma|) \times 2^{|\Delta| \times |\Gamma|},$$

where $f(|\Delta|)$ and $f(|\Gamma|)$ are the numbers of DAGs for respectively the discrete and the continuous nodes, and $2^{|\Delta| \times |\Gamma|}$ denotes the number of different combinations of arrows from discrete to continuous nodes. If the number of random variables in the network is large, it is computationally infeasible to calculate the network score for all the possible DAGs. Therefore different methods for searching for DAGs with high network score have been tried, see *e.g.* Cooper & Herskovits (1992). In Section 8.3 we will describe one of these methods, namely greedy search with random restarts. This method, like many others, make use of Bayes factors as a way of comparing the network scores for two different DAGs. In the next section we will therefore consider Bayes factors for mixed networks.

8.1 Bayes Factors

A way to compare the network score for two different networks, D and D^*, is to calculate the *posterior odds*, given by

$$\frac{p(D|d)}{p(D^*|d)} = \frac{p(D,d)}{p(D^*,d)} = \frac{p(D)}{p(D^*)} \times \frac{p(d|D)}{p(d|D^*)},$$

where $p(D)/p(D^*)$ is the *prior odds* and $p(d|D)/p(d|D^*)$ is the *Bayes factor*.

The posterior odds is for numerical reasons often calculated using the logarithm,

$$\log\left(\frac{p(D|d)}{p(D^*|d)}\right) = \log(p(D|d)) - \log(p(D^*|d)).$$

For two models that differ only by a single arrow, the Bayes factor is, because of decomposability, especially simple. In this section, we will specify the Bayes factor in the case where two DAGs differ by the direction of a single arrow and in the case where two DAGs differ by the presence of a single arrow.

First we look at the former case. As discrete nodes can not have continuous parents, we only look at reversing an arrow between two discrete variables or two continuous variables. In the following let $v \leftarrow w$ in D and $v \rightarrow w$ in D^*. Further let ∇_w be the parents of w in D and ∇_v the parents of v in D^*. As D and D^* only differ by the direction of the arrow between v and w, the parents of w in D^* are ∇_w and v and the parents of v in D are ∇_v and w. Notice that if v and w are discrete nodes, then the nodes in ∇_v and ∇_w can only be discrete, whereas if v and w are continuous nodes, they can be both discrete and continuous.

To simplify, we let the database consist of just one case, so $d = \{x\}$. As the

likelihood terms are decomposable, the Bayes factor is given by

$$\begin{aligned}\frac{p(x|D)}{p(x|D^*)} &= \frac{p(v|\nabla_v, w, D)p(w|\nabla_w, D)}{p(w|\nabla_w, v, D^*)p(v|\nabla_v, D^*)} \\ &= \frac{\int p(x_v|x_{w\cup\nabla_v}, H_{v|w\cup\nabla_v}, D)p(H_{v|w\cup\nabla_v}|D)dH_{v|w\cup\nabla_v}}{\int p(x_w|x_{v\cup\nabla_w}, H_{w|v\cup\nabla_w}, D^*)p(H_{w|v\cup\nabla_w}|D^*)dH_{w|v\cup\nabla_w}} \\ &\times \frac{\int p(x_w|x_{\nabla_w}, H_{w|\nabla_w}, D)p(H_{w|\nabla_w}|D)dH_{w|\nabla_w}}{\int p(x_v|x_{\nabla_v}, H_{v|\nabla_v}, D^*)p(H_{v|\nabla_v}|D^*)dH_{v|\nabla_v}}.\end{aligned}$$

So to calculate the Bayes factor between D and D^*, we only need to consider the terms involving the conditional distributions of v and of w.

Notice that if $\nabla_v = \nabla_w$, then D and D^* are independence equivalent networks and the Bayes factor is equal to one.

Now let D and D^* be two different networks, that differ by a single arrow between the nodes v and w, with $v \leftarrow w$ in D and $v \nleftarrow w$ in D^*. Here v and w can be either both discrete variables, both continuous or v continuous and w discrete. Again, let ∇_v be the set of variables that are parents of v in D^*, so in D the parents of v are ∇_v and w. As the likelihood terms are decomposable, the Bayes factor is given by

$$\begin{aligned}\frac{p(x|D)}{p(x|D^*)} &= \frac{p(x_v|x_{w\cup\nabla_v}, D)}{p(x_v|x_{\nabla_v}, D^*)} \\ &= \frac{\int p(x_v|x_{w\cup\nabla_v}, H_{v|w\cup\nabla_v}, D)p(H_{v|w\cup\nabla_v}|D)dH_{v|w\cup\nabla_v}}{\int p(x_v|x_{\nabla_D}, H_{v|\nabla_v}, D^*)p(H_{v|\nabla_v}|D^*)dH_{v|\nabla_v}}.\end{aligned}$$

8.2 Equivalent Bayes Factors

To compare network scores for all networks which differ by only one arrow, is computationally inefficient. When using the local master procedure, we can reduce the number of comparisons needed.

Our goal is to identify classes of DAGs for which the corresponding Bayes factors for testing an arrow between the same two variables in the network, are the same. So let D_1 and D_1^* be two different networks that differ by a single arrow between the nodes v and w, with $v \leftarrow w$ in D_1 and $v \nleftarrow w$ in D_1^*. Further, let ∇_{v_1} be the set of variables that are parents of v in both D_1 and D_1^*, *i.e.* in D_1 the parents of v are ∇_{v_1} and w and in D_1^* just ∇_{D_1}.

Further let D_2 and D_2^* be another two networks different from D_1 and D_1^* that differ by an arrow between v and w and let ∇_{v_2} be the set of variables that are parents of v in both D_2 and D_2^*. There are two situations to consider, namely when $v \leftarrow w$ in D_2 and when $v \rightarrow w$ in D_2.

Consider first the former situation. The Bayes factor for testing D_1 against D_1^* was

Figure 1: Equivalence due to parameter modularity.

in the previous section found to be

$$\frac{p(x|D_1)}{p(x|D_1^*)} = \frac{\int p(x_v|x_{w\cup\nabla_{v_1}}, H_{v|w\cup\nabla_{v_1}}, D_1)p(H_{v|w\cup\nabla_{v_1}}|D_1)dH_{v|w\cup\nabla_{v_1}}}{\int p(x_v|x_{\nabla_{v_1}}, H_{v|\nabla_{v_1}}, D_1^*)p(H_{v|\nabla_{v_1}}|D_1^*)dH_{v|\nabla_{v_1}}}. \quad (12)$$

Likewise the Bayes factor for testing D_2 against D_2^* is

$$\frac{p(x|D_2)}{p(x|D_2^*)} = \frac{\int p(x_v|x_{w\cup\nabla_{v_2}}, H_{v|w\cup\nabla_{v_2}}, D_2)p(H_{v|w\cup\nabla_{v_2}}|D_2)dH_{v|w\cup\nabla_{v_2}}}{\int p(x_v|x_{\nabla_{v_2}}, H_{v|\nabla_{v_2}}, D_2^*)p(H_{v|\nabla_{v_2}}|D_2^*)dH_{v|\nabla_{v_2}}}.$$

As the local master procedure has the property of parameter modularity, then if $\nabla_{v_1} = \nabla_{v_2}$ it follows that

$$p(H_{v|w\cup\nabla_{v_1}}|D_1) = p(H_{v|w\cup\nabla_{v_2}}|D_2),$$

and

$$p(x_v|x_{w\cup\nabla_{v_1}}, H_{v|w\cup\nabla_{v_1}}, D_1) = p(x_v|x_{w\cup\nabla_{v_2}}, H_{v|w\cup\nabla_{v_2}}, D_2).$$

So the Bayes factor for testing the arrow from v to w is equivalent to testing this arrow in any other network, where v has the same parents as in D_1, *i.e.* if $\nabla_{v_1} = \nabla_{v_2}$. This is illustrated in Figure 1.

Consider now the situation where $v \to w$ in D_2. Let ∇_{w_2} be the set of variables, that are parents of w in both D_2 and D_2^*. The Bayes factor is given as

$$\begin{aligned}\frac{p(x|D_2)}{p(x|D_2^*)} &= \frac{p(x_w|x_{v\cup\nabla_{w_2}}, D_2)}{p(x_w|x_{\nabla_{w_2}}, D_2^*)} \\ &= \frac{\int p(x_w|x_{v\cup\nabla_{w_2}}, H_{w|v\cup\nabla_{w_2}}, D_2)p(H_{w|v\cup\nabla_{w_2}}|D_2)dH_{w|v\cup\nabla_{w_2}}}{\int p(x_w|x_{\nabla_{w_2}}, H_{w|\nabla_{w_2}}, D_2^*)p(H_{w|\nabla_{w_2}}|D_2^*)dH_{w|\nabla_{w_2}}}.\end{aligned}$$

Again we see that because of parameter modularity, this Bayes factor is the same as the Bayes factor given in (12), if $\nabla_{v_1} = \nabla_{w_2}$, *i.e.* if w in D_2 has the same parents as v does in D_1, with the exception that v is a parent of w in D_2. For an illustration, see Figure 2.

To show that these situations are the only ones where the Bayes factors always are the same, it is easy to find an example where $\nabla_{v_1} \neq \nabla_{v_2}$ and the Bayes factors are not same.

The above result is summarized in the following theorem.

Figure 2: Equivalence due to property of local master procedure.

Theorem 8.1
The Bayes factor for testing the arrow $v \leftarrow w$ in a DAG D_1 is equivalent to the Bayes factor for testing the same arrow in any other network D_2 if and only if the following two criteria are met:

(1) $v \leftarrow w$ and v in D_2 has the same parents as in D_1.

(2) $v \rightarrow w$ and w in D_2 has the same parents as v does in D_1, with the exception that v is a parent of w in D_2.

Although using the two criteria reduces the number of comparisons, there will still, for large networks, be too many comparisons needed for finding the most likely DAG. Therefore it is still necessary to use some kind of search strategy.

8.3 Greedy search with random restarts

As mentioned earlier, many search strategies use Bayes factors as a way to compare the network score for two different networks. In the following we will describe one such strategy called *greedy search*.

Greedy search is initialized by choosing a network D from which to start the search. Let Δe be the posterior odds between two networks that differ by an arrow. Calculate then Δe for all DAGs D^* that differ from D by a single arrow e, either added, removed or reversed. Make the change e for which Δe is a minimum, that is where $p(D^*|d)$ is a maximum and continue the search from this new network. The search is terminated when there is no e with Δe smaller than 1. As shown in the previous section, the posterior odds is because of decomposability especially simple, as D and D^* only differ by one arrow. Further, it is possible to reduce the time complexity by using the equivalence criteria developed in Section 8.2.

As this search is local in the sense that it only evaluates local changes to the network, there is a chance that the found maximum is only a local maximum. A way to overcome this problem is to randomly perturb the structure of the start network D and restart the greedy search from this new network. This can be repeated a manageable number of times and between the networks found by the search strategy, the network with the highest score is chosen.

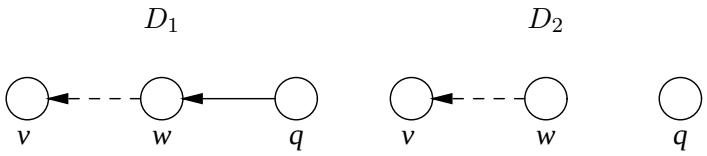


Figure 3: Models for which the Bayes factors are equivalent.

8.4 Priors on DAGs

In this section we will consider how to assign prior probabilities to the possible DAGs in a given problem. As shown in various papers, there are different ways of doing this. The Bayesian way would be to assess the prior belief in each DAG, but as the number of different DAGs grow, this is not manageable. Instead automated methods is being used.

An often used approach is to assume that all DAGs are equally likely, thus letting the prior probability distribution over DAGs be uniform. This approach is mostly used only for simplicity and can be refined in various ways. For example, if we know that some of the DAGs are not possible, then we can assign probability zero to these and equal probabilities to the rest. Because of likelihood equivalence, DAGs within the same equivalence class will, with this approach, be assigned the same network score.

One argument against letting the prior over DAGs be uniform is that the number of different DAGs in an equivalence class varies between equivalence classes. This means that the conditional independencies represented in an equivalence class with many DAGs, a priori are more probable than those represented in an equivalence class with fewer DAGs. When using model averaging, this is a problem because it involves a sum over all the different DAGs. The conditional independencies represented by a large equivalence class, therefore influence the result more than those represented by a small equivalence class. A way to handle this problem is to either include only one DAG from each equivalence class or instead let all equivalence classes be equally likely and assign to each DAG a prior probability inversely proportional to the number of DAGs in the equivalence class it belongs to.

This last approach has, however, an affect on the posterior odds. Consider the following example, illustrated in Figure 3.

According to criteria one in Theorem 8.1, the Bayes factor for testing the presence of the arrow $v \leftarrow w$ in D_1 is equivalent to testing $v \leftarrow w$ in D_2, *i.e.*

$$\frac{p(v|w, D_1)}{p(v|D_1^*)} = \frac{p(v|w, D_2)}{p(v|D_2^*)}.$$

If we assign equal priors to all DAGs, the posterior odds are the same as the Bayes

factors and they will therefore also be equivalent in the above example. However, if we let all equivalence classes be equally likely and assign to each DAG a prior probability inversely proportional to the number of DAGs in the equivalence class it belongs to, the posterior odds are no longer the same as the Bayes factors. In the above example, the number of DAGs in the equivalence classes for D_1, D_1^*, D_2 and D_2^* are respectively 3, 2, 2 and 1. So the prior odds are not equivalent, *i.e.*

$$\frac{p(D_1)}{p(D_1^*)} = \frac{2}{3} \neq \frac{1}{2} = \frac{p(D_2)}{p(D_2^*)},$$

and therefore the posterior odds are not equivalent either. So this approach should not be used if we in a search strategy want to utilize that some of the Bayes factors are equivalent.

9 Example

In the following, some of the methods derived are illustrated by a simple example. This example was constructed by Morrison (1976) and also studied in Edwards (1995).

9.1 The Dataset

The dataset is from a hypothetical drug trial, where the weight losses of male and female rats under three different drug treatments have been measured after one and two weeks. Thus we have the discrete variables I_{sex} and I_{drug} with states

$$I_{sex} = \{\text{male} = 1, \text{female} = 2\}$$
$$I_{drug} = \{1, 2, 3\},$$

and the continuous variables Y_{w1} and Y_{w2} which respectively represents the weight losses after one and two weeks. For every drug, four rats of each sex have been treated, which gives a total of 24 observations. The observations are shown in Table 1.

9.2 Specifying the Prior Network

We start by specifying a prior Bayesian network $(D, \mathcal{P})$. To simplify the specification of the joint parameter prior, we choose to let all the variables be independent, so the local probability distribution for each node only depends on the node itself, and we can specify them as follows.

sex	$drug$	$w1$	$w2$	sex	$drug$	$w1$	$w2$
1	1	5	6	2	1	7	10
1	1	7	6	2	1	8	10
1	1	9	9	2	1	6	6
1	1	5	4	2	1	9	7
1	2	9	12	2	2	7	6
1	2	7	7	2	2	10	13
1	2	7	6	2	2	6	9
1	2	6	8	2	2	8	7
1	3	14	11	2	3	14	9
1	3	21	15	2	3	14	8
1	3	12	10	2	3	16	12
1	3	17	12	2	3	10	5

Table 1: Observations of weight loss of male and female rats under three different drug treatments.

For each discrete variable, we let each state be equally likely, so

$$p(i_{sex} = 1) = p(i_{sex} = 2) = \frac{1}{2}$$

and

$$p(i_{drug} = 1) = p(i_{drug} = 2) = p(i_{drug} = 3) = \frac{1}{3}.$$

This in fact is true by design.

For the continuous variables we use the sample mean and the sample variance as an initial estimate of the mean and the variance. Using this approach, the position and scale of the parameters are determined. We find that

$$p(y_{w1}) = \mathcal{N}(9.6, 17.1)$$

and

$$p(y_{w2}) = \mathcal{N}(8.7, 7.6).$$

So jointly

$$p(i, y) = p(i)\mathcal{N}(m_i, \Sigma_i),$$

with

$$p(i) = \frac{1}{6}, \quad m_i = \begin{pmatrix} 9.6 \\ 8.7 \end{pmatrix} \quad \text{and} \quad \Sigma_i = \begin{pmatrix} 17.1 & 0 \\ 0 & 7.6 \end{pmatrix},$$

for all possible configurations of i.

Be aware that in this way the dataset is used twice, namely both to initially specify the local probability distributions and later to find the posterior parameter distributions. This could result in parameter values that are overfitted to data.

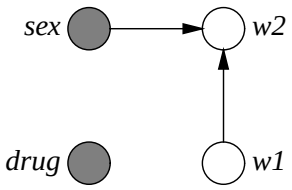


Figure 4: The DAG in the example for specification of local parameter priors.

9.3 Specifying Parameter Priors

In order to specify parameter priors for all possible networks, we use the local master procedure.

First we translate the prior network into an imaginary database. The parameters needed to represent this imaginary database are n, α_i, ν_i, ρ_i, μ_i and Φ_i.

Here we let $\Phi_i = (\nu_i - 1)\Sigma_i$, so ν_i must be larger than 1. This means in this example that n must be larger than 6. We choose $n = 12$ and find that

$$\alpha_i = \nu_i = \rho_i = p(i)n = \frac{1}{6}12 = 2.$$

Further

$$\mu_i = m_i = \begin{pmatrix} 9.6 \\ 8.7 \end{pmatrix} \quad \text{and} \quad \Phi_i = (\nu_i - 1)\Sigma_i = \begin{pmatrix} 17.0 & 0 \\ 0 & 7.6 \end{pmatrix},$$

for all configurations of i.

We can now specify parameter priors for all possible networks. As an illustration, consider the parameter prior for the network in Figure 4.

We need to find the local masters for the following four families

$$\begin{aligned} A_1 &= \{sex\}, \\ A_2 &= \{drug\}, \\ A_3 &= \{w1\}, \\ A_4 &= \{sex, w1, w2\}. \end{aligned}$$

As the variables in A_1, A_2 and A_3 do not have any parents, the local masters for these families are also the local parameter priors. Thus the local parameter prior for I_{sex} is given by

$$\Psi_{sex} \sim \mathcal{D}(\alpha_{sex}),$$

with

$$\alpha_{i_{sex}=1} = \sum_{j:j_{sex}=1} \alpha_j = 6 \text{ and } \alpha_{i_{sex}=2} = \sum_{j:j_{sex}=2} \alpha_j = 6.$$

Similarly the local parameter prior for I_{drug} is

$$\Psi_{drug} \sim \mathcal{D}(\alpha_{drug}),$$

with

$$\alpha_{i_{drug}=1} = \alpha_{i_{drug}=2} = \alpha_{i_{drug}=3} = 4.$$

For Y_{w1} we find the local parameter prior to be

$$\begin{aligned} \Sigma_{w1} &\sim \mathcal{IW}(\rho, \tilde{\Phi}_{w1}), \\ m_{w1}|\Sigma_{w1} &\sim \mathcal{N}(\overline{\mu}_{w1}, \frac{1}{\nu}\Sigma_{w1}), \end{aligned}$$

with

$$\rho = \sum_j \rho_j = 12 \text{ and } \nu = \sum_j \nu_j = 12,$$

and

$$\begin{aligned} \overline{\mu} &= \frac{\sum_i \mu_i \nu_i}{\nu} = \begin{pmatrix} 9.6 \\ 8.7 \end{pmatrix}, \\ \tilde{\Phi} &= \sum_i \Phi_i + \sum_i \nu_i (\mu_i - \overline{\mu})(\mu_i - \overline{\mu})^{\mathrm{T}} = \begin{pmatrix} 102.6 & 0 \\ 0 & 45.6 \end{pmatrix}, \end{aligned}$$

so

$$\overline{\mu}_{w1} = 9.6 \text{ and } \tilde{\Phi}_{w1} = 102.6.$$

The local master for the family A_4 is given as

$$\begin{aligned} (\Sigma_{i_{sex}}) &\sim \mathcal{IW}(\rho_{i_{sex}}, (\tilde{\Phi}_{i_{sex}})), \\ (m_{i_{sex}})|(\Sigma_{i_{sex}}) &\sim \mathcal{N}((\overline{\mu}_{i_{sex}}), \frac{1}{\nu_{i_{sex}}}(\Sigma_{i_{sex}})), \end{aligned}$$

with

$$\rho_{i_{sex}=1} = \sum_{j:j_{sex}=1} \rho_j = 6 \text{ and } \rho_{i_{sex}=2} = \sum_{j:j_{sex}=2} \rho_j = 6.$$

Likewise for $\nu_{i_{sex}}$. Further

$$\overline{\mu}_{i_{sex}=1} = \frac{\sum_{j:j_{sex}=1} \mu_j \nu_j}{\nu_{i_{sex}=1}} = \begin{pmatrix} 9.6 \\ 8.7 \end{pmatrix}$$

and

$$\begin{aligned} \tilde{\Phi}_{i_{sex}=1} &= \sum_{j:j_{sex}=1} \Phi_j + \sum_{j:j_{sex}=1} \nu_j (\mu_j - \overline{\mu}_{i_{sex}=1})(\mu_j - \overline{\mu}_{i_{sex}=1})^{\mathrm{T}} \\ &= \begin{pmatrix} 51.3 & 0 \\ 0 & 22.8 \end{pmatrix} \end{aligned}$$

and the same for $i_{sex} = 2$.

The local parameter prior for Y_{w2} given Y_{w1} and I_{sex} can now be found by conditioning in this local master distribution.

We have now specified the parameters needed to calculate the likelihood of a DAG, $p(d|D)$. To calculate the network score of D, we also need to specify the prior probability of D. In this example we just choose to let all DAGs be equally likely and thus use the likelihood $p(d|D)$ as the network score.

9.4 Result

Using the formula on page 20, we find that for a network with two discrete and two continuous nodes, there are 144 possible DAGs. So in this example, there are no computational problems in calculating the network score for all these DAGs. Further, if we only calculate the score for DAGs that are not independence equivalent, the number of different DAGs are reduced to 88.

Prior network			Imaginary sample size		12
 1	 0.68	 0.30	 0.20	 0.12	 0.12
 0.075	 0.060	 0.051	 0.037	 0.035	 0.028
 0.023	 0.022	 0.018	 0.015	 0.0093	 0.0084
 0.0076	 0.0072	 0.0069	 0.0037	 0.0028	 0.0023
 0.0022	 0.0020	 0.0019	 0.0017	 0.0011	 $9.6 \cdot 10^{-4}$
 $6.0 \cdot 10^{-4}$	 $5.6 \cdot 10^{-4}$	 $5.2 \cdot 10^{-4}$	 $4.5 \cdot 10^{-4}$	 $2.9 \cdot 10^{-4}$	 $1.9 \cdot 10^{-4}$
 $1.7 \cdot 10^{-4}$	 $1.7 \cdot 10^{-4}$	 $1.6 \cdot 10^{-4}$	$1.5 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$

continued on next page

continued from previous page					
Prior network			Imaginary sample size		12
 $1.3 \cdot 10^{-4}$	 $1.1 \cdot 10^{-4}$	 $8.9 \cdot 10^{-5}$	 $8.0 \cdot 10^{-5}$	 $7.2 \cdot 10^{-5}$	 $5.5 \cdot 10^{-5}$
 $5.2 \cdot 10^{-5}$	 $5.0 \cdot 10^{-5}$	 $4.7 \cdot 10^{-5}$	 $4.5 \cdot 10^{-5}$	 $4.2 \cdot 10^{-5}$	 $4.2 \cdot 10^{-5}$
 $3.9 \cdot 10^{-5}$	 $3.8 \cdot 10^{-5}$	 $3.4 \cdot 10^{-5}$	 $3.2 \cdot 10^{-5}$	 $2.7 \cdot 10^{-5}$	 $2.4 \cdot 10^{-5}$
 $2.2 \cdot 10^{-5}$	 $2.0 \cdot 10^{-5}$	 $1.6 \cdot 10^{-5}$	 $1.3 \cdot 10^{-5}$	 $1.1 \cdot 10^{-5}$	 $1.0 \cdot 10^{-5}$
 $5.9 \cdot 10^{-6}$	 $4.9 \cdot 10^{-6}$	 $3.6 \cdot 10^{-6}$	 $3.0 \cdot 10^{-6}$	 $1.1 \cdot 10^{-6}$	 $9.0 \cdot 10^{-7}$
 $3.2 \cdot 10^{-7}$	 $3.0 \cdot 10^{-7}$	 $2.6 \cdot 10^{-7}$	 $2.5 \cdot 10^{-7}$	 $1.5 \cdot 10^{-7}$	 $1.3 \cdot 10^{-7}$
 $9.4 \cdot 10^{-8}$	 $8.9 \cdot 10^{-8}$	 $7.8 \cdot 10^{-8}$	$7.4 \cdot 10^{-8}$	$7.2 \cdot 10^{-8}$	$5.9 \cdot 10^{-8}$
$4.5 \cdot 10^{-8}$	$3.8 \cdot 10^{-8}$	$2.1 \cdot 10^{-8}$	$1.8 \cdot 10^{-8}$		

Table 2: The DAGs in the reduced search space, listed in decreasing order of probability. The number below each DAG is the Bayes factor between the given DAG and the DAG with the highest network score.

In Table 2 the result of the learning procedure is given. The DAGs are listed in decreasing order of probability, and the number below each DAG is the posterior odds between the given DAG and the DAG with the highest network score. This number expresses the relative probability of a DAG, that is, relative to the DAG with the highest network score. As we have chosen a uniform prior over DAGs, the posterior odds is in this example equal to the Bayes factor.

Before analyzing the result, we can discard some of the networks in Table 2. By design, the discrete variables sex and $drug$ are independent, so there should not be an arrow between sex and $drug$. Further, there is a time restriction between $w1$ and $w2$, as $w1$ is observed before $w2$. So if $w1$ and $w2$ are dependent, the arrow between $w1$ and $w2$ must go from $w1$ to $w2$. Taking these restrictions into account, we only consider the 32 different DAGs listed in Table 3.

Prior network			Imaginary sample size		12
 1	 0.68	 0.12	 0.075	 0.051	 0.023
 0.0093	 0.0020	 0.0019	 0.0017	 $9.6 \cdot 10^{-4}$	 $4.5 \cdot 10^{-4}$
 $1.6 \cdot 10^{-4}$	 $1.5 \cdot 10^{-4}$	 $1.4 \cdot 10^{-4}$	 $1.3 \cdot 10^{-4}$	 $1.1 \cdot 10^{-4}$	 $8.9 \cdot 10^{-5}$
 $7.2 \cdot 10^{-5}$	 $3.4 \cdot 10^{-5}$	 $2.0 \cdot 10^{-5}$	 $1.6 \cdot 10^{-5}$	 $3.6 \cdot 10^{-6}$	 $3.0 \cdot 10^{-6}$
 $3.2 \cdot 10^{-7}$	 $3.0 \cdot 10^{-7}$	 $2.6 \cdot 10^{-7}$	 $2.5 \cdot 10^{-7}$	 $1.5 \cdot 10^{-7}$	 $1.3 \cdot 10^{-7}$
 $7.2 \cdot 10^{-8}$	 $5.9 \cdot 10^{-8}$				

Table 3: The DAGs in the reduced search space, listed in decreasing order of probability. The number below each DAG is the Bayes factor between the given DAG and the DAG with the highest network score.

In the most probable DAG, we see that $w2$ depends on $w1$ and $w1$ depends on $drug$. Further $w2$ and $drug$ are conditionally independent given $w1$ and both $w1$ and $w2$ are independent on sex.

Almost the same dependency structure is seen in the second and third best DAG, except that here $w2$ also depends on respectively sex and $drug$.

Generally we see that in the first 12 DAGs, $w1$ depends on $drug$. The first DAG that does not show this dependency relation is only 0.00016 times as probable as the best DAG. Likewise we see that in the first 7 DAGs, $w2$ depends on $w1$ and the first DAG that does not contain this dependency relation is only 0.0020 as probable as the best DAG. Therefore we should not consider any model that does not include these dependencies.

It is not clear which independencies should be included in the model, except for those introduced when we reduced the search space. The second DAG is for example 0.68 times as probable as the first DAG, and the third to the sixth DAG is between 0.12 and 0.023 as probable as the best DAG. This suggest that there is some unexplained variation not accounted for in the best DAG and it might therefore be more accurate to select *e.g.* the first six models and use model averaging.

In Edwards (1995) the dataset is analyzed using undirected graphical models. He uses the software MIM for maximum likelihood estimation and likelihood ratio test. The result is displayed in Figure 5 and we see that it is not in conflict with our result.

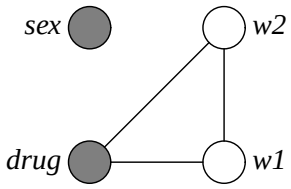


Figure 5: Previous result.

9.5 Sensitivity to Prior Information

In this section we will explore how the size of the imaginary database and the choice of the prior network influences the result. The findings agree with findings for a purely discrete case described in Steck & Jaakkola (2002).

Prior network			Imaginary sample size		2000
 1	 0.99	 0.97	 0.96	 0.96	 0.95
 0.94	 0.93	 0.89	 0.89	 0.89	 0.89
 0.88	 0.88	 0.88	 0.87	 0.87	 0.87
 0.86	 0.86	 0.85	 0.85	 0.84	 0.83
 0.79	 0.79	 0.79	 0.79	 0.78	 0.78
 0.78	 0.78				

Table 4: The revised result with the prior network and the imaginary sample size specified as in the first line of this table.

Recall that the prior network ideally expresses which dependency structure we

believe there is between the variables in the network and the size of the imaginary database expresses how much confidence we have in this dependency structure.

In the previous section we used the empty network as the prior network and set the size n of the imaginary database to 12. This is less than the number of real observations in the example, which is 24. We will therefore also learn the networks using a larger value of n and to see the difference clearly, we use $n = 2000$. The result is given in Table 4.

If we look at the three best networks from the previous result, we see that the relative probabilities for these networks in this result, are between 0.94 and 0.97. They are no longer the most probable networks, but they are still very probable. Actually all the networks are very probable and the relative probability of the least probable network is as much as 0.78.

The reason for this is that the prior network is the empty network, which represents that all the variables are independent. This model is therefore a submodel of all other models. When n is large, we have much confidence in these independencies, so all networks will a priori be very probable. As the real database only contains few observations, we have not enough information to differentiate between these networks and all the networks are therefore almost equally likely.

Prior network			Imaginary sample size		12
 1	 0.59	 0.38	 0.34	 0.17	 0.10
 0.064	 0.056	 $2.7 \cdot 10^{-4}$	 $1.3 \cdot 10^{-4}$	 $1.2 \cdot 10^{-4}$	 $4.8 \cdot 10^{-5}$
 $4.5 \cdot 10^{-5}$	 $2.2 \cdot 10^{-5}$	 $1.9 \cdot 10^{-5}$	 $7.9 \cdot 10^{-6}$	 $7.5 \cdot 10^{-8}$	 $5.0 \cdot 10^{-8}$
 $4.4 \cdot 10^{-8}$	 $2.9 \cdot 10^{-8}$	 $2.9 \cdot 10^{-8}$	 $2.6 \cdot 10^{-8}$	 $1.9 \cdot 10^{-8}$	 $1.7 \cdot 10^{-8}$
 $2.1 \cdot 10^{-11}$	 $1.4 \cdot 10^{-11}$	 $9.9 \cdot 10^{-12}$	 $8.9 \cdot 10^{-12}$	 $6.5 \cdot 10^{-12}$	 $5.8 \cdot 10^{-12}$
 $3.6 \cdot 10^{-12}$	 $2.4 \cdot 10^{-12}$				

Table 5: The revised result with the prior network and the imaginary sample size specified as in the first line of this table.

Prior network			Imaginary sample size		2000
 1	 1	 0.99	 0.98	 0.98	 0.98
 0.97	 0.96	 $6.5 \cdot 10^{-4}$	 $6.4 \cdot 10^{-4}$	 $6.4 \cdot 10^{-4}$	 $6.3 \cdot 10^{-4}$
 $1.9 \cdot 10^{-4}$	 $1.8 \cdot 10^{-4}$	 $1.8 \cdot 10^{-4}$	 $1.8 \cdot 10^{-4}$	 $3.5 \cdot 10^{-9}$	 $3.5 \cdot 10^{-9}$
 $3.5 \cdot 10^{-9}$	 $3.5 \cdot 10^{-9}$	 $3.4 \cdot 10^{-9}$	 $3.4 \cdot 10^{-9}$	 $3.4 \cdot 10^{-9}$	 $3.4 \cdot 10^{-9}$
 $2.2 \cdot 10^{-12}$	 $2.2 \cdot 10^{-12}$	 $2.2 \cdot 10^{-12}$	 $2.2 \cdot 10^{-12}$	 $6.4 \cdot 10^{-13}$	 $6.4 \cdot 10^{-13}$
 $6.4 \cdot 10^{-13}$	 $6.4 \cdot 10^{-13}$				

Table 6: The revised result with the prior network and the imaginary sample size specified as in the first line of this table.

We will now explore what happens if we change the prior network. First we will learn the structure using the most probable structure from Table 3 as the prior network. The results with $n = 12$ and $n = 2000$ are given in respectively Table 5 and Table 6.

For $n = 12$ we see almost the same result as when using the empty network. The best networks are, not surprisingly, the same, only the order between them are a little different. To some extent, this also applies for $n = 2000$.

Further we see that for both $n = 12$ and $n = 2000$, the 32 networks categorize as follows. The 8 networks with both arrows $drug \rightarrow w1$ and $w1 \rightarrow w2$ are the 8 most probable networks. In the succeeding 8 networks we have $drug \rightarrow w1$ and $w1 \nrightarrow w2$, after that the 8 networks with $drug \nrightarrow w1$ and $w1 \rightarrow w2$. In the last 8 networks we have $drug \nrightarrow w1$ and $w1 \nrightarrow w2$. Also we see that within each category, the networks are almost equally likely, mostly pronounced for $n = 2000$. These finding are what we expected. The arrows included in the prior network are all represented in the most probable networks and these networks are all almost equally likely, as the prior network is a submodel of these. Further there is a large difference in relative score between the different categories, which shows that networks which include the arrows $drug \rightarrow w1$ and $w1 \rightarrow w2$, are

much more likely than those that do not. As this is valid for both $n = 12$ and $n = 2000$, it is not only due to the influence of the prior network, but also because the dataset supports these dependencies.

We will now explore what happens if we choose the prior network to be the least probable network from Table 3. The results are for $n = 12$ and $n = 2000$ given in respectively Table 7 and Table 8.

Prior network			Imaginary sample size		12
 1	 0.25	 0.13	 0.094	 0.023	 0.012
 0.0079	 0.0020	 0.0018	 $9.6 \cdot 10^{-4}$	 $7.4 \cdot 10^{-4}$	 $3.9 \cdot 10^{-4}$
 $1.8 \cdot 10^{-4}$	 $1.7 \cdot 10^{-4}$	 $1.6 \cdot 10^{-4}$	 $1.4 \cdot 10^{-4}$	 $9.0 \cdot 10^{-5}$	 $3.9 \cdot 10^{-5}$
 $3.7 \cdot 10^{-5}$	 $3.5 \cdot 10^{-5}$	 $2.0 \cdot 10^{-5}$	 $1.8 \cdot 10^{-5}$	 $1.2 \cdot 10^{-6}$	 $1.1 \cdot 10^{-6}$
 $3.1 \cdot 10^{-7}$	 $2.9 \cdot 10^{-7}$	 $2.8 \cdot 10^{-7}$	 $2.6 \cdot 10^{-7}$	 $1.5 \cdot 10^{-7}$	 $1.4 \cdot 10^{-7}$
 $6.2 \cdot 10^{-8}$	 $5.5 \cdot 10^{-8}$				

Table 7: The revised result with the prior network and the imaginary sample size specified as in the first line of this table.

For $n = 12$ we see almost the same result as with the other prior networks. For $n = 2000$ we see that the 8 most probable models actually are the 8 models that are possible with both the arrows $sex \rightarrow w1$ and $sex \rightarrow w2$. Further we see that all networks are almost equally likely and there is not, as would be expected, a large difference in score between networks with both arrows and the others. Actually for both $n = 12$ and $n = 2000$ the result is very similar to the result with the empty network as the prior networks. The reason for this is that the probability distribution of the prior network is estimated from data, *i.e.* we use the sample mean and sample variance as the mean and variance in the prior network. If data does not support a dependence between sex and respectively $w1$ and $w2$, then this prior network will be almost the same as the empty prior network and so will the result of the learning procedure. However, it can be seen that even small differences from

Prior network			Imaginary sample size		2000
 1	 0.99	 0.94	 0.94	 0.91	 0.90
 0.86	 0.86	 0.67	 0.65	 0.61	 0.59
 0.59	 0.59	 0.54	 0.53	 0.38	 0.37
 0.35	 0.35	 0.34	 0.33	 0.32	 0.31
 0.25	 0.25	 0.22	 0.22	 0.22	 0.22
 0.20	 0.20				

Table 8: The revised result with the prior network and the imaginary sample size specified as in the first line of this table.

the empty prior network have an impact when n is large, as the 8 most probable networks actually are the ones with both $sex \rightarrow w1$ and $sex \rightarrow w2$.

Acknowledgements

This research was supported by the ESPRIT project P29105 (BaKE). Also I would like to thank my supervisor Steffen L. Lauritzen for his dedicated help and support.

References

Bernardo, J. M. & Smith, A. F. M. (1994). *Bayesian Theory*, John Wiley & Sons, Chichester.

Bøttcher, S. G. (2001). Learning Bayesian Networks with Mixed Variables, *Artificial Intelligence and Statistics 2001*, Morgan Kaufmann, San Francisco, CA, USA, pp. 149–156.

Chickering, D. M. (1995). A transformational characterization of equivalent Bayesian-network structures, *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, CA, USA, pp. 87–98.

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

Dawid, A. P. & Lauritzen, S. L. (1993). Hyper Markov laws in the statistical analysis of decomposable graphical models, *The Annals of Statistics* **21**(3): 1272–1317.

DeGroot, M. H. (1970). *Optimal Statistical Decisions*, McGraw-Hill, New York.

Edwards, D. (1995). *Introduction to Graphical Modelling*, Springer-Verlag, New York.

Frydenberg, M. (1990). Marginalization and collapsibility in graphical interaction models, *Annals of Statistics* **18**: 790–805.

Geiger, D. & Heckerman, D. (1994). Learning Gaussian Networks, *Proceedings of Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, CA, USA, pp. 235–243.

Heckerman, D., Geiger, D. & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data, *Machine Learning* **20**: 197–243.

Lauritzen, S. L. (1992). Propagation of probabilities, means and variances in mixed graphical association models, *Journal of the American Statistical Association* **87**(420): 1098–1108.

Lauritzen, S. L. (1996). *Graphical Models*, Clarendon press, Oxford, New York.

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

Morrison, D. F. (1976). *Multivariate Statistical Methods*, McGraw-Hill, USA.

Robinson, R. W. (1977). Counting unlabeled acyclic digraphs, *Lecture Notes in Mathematics, 622: Combinatorial Mathematics V* pp. 239–273.

Seber, G. A. F. (1984). *Multivariate Observations*, John Wiley and Sons, New York.

Shachter, R. D. & Kenley, C. R. (1989). Gaussian influence diagrams, *Management Science* **35**: 527–550.

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

Steck, H. & Jaakkola, T. S. (2002). On the Dirichlet Prior and Bayesian Regularization, *Conference on Advances in Neural Information Processing Systems*, Vol. 15, MIT Press, Cambridge, USA.