Probabilistic Soft Interventions in Conditional Gaussian Networks

Florian Markowetz, Steffen Grossmann, and Rainer Spang
firstname.lastname@molgen.mpg.de
Dept. Computational Molecular Biology
Max Planck Institute for Molecular Genetics
Berlin, Germany

Abstract

We introduce a general concept of probabilistic interventions in Bayesian networks. This generalizes deterministic interventions, which fix nodes to certain states. We propose "pushing" variables in the direction of target states without fixing them. We formalize this idea in a Bayesian framework based on Conditional Gaussian networks.

1 Introduction

In modern biology, the key to infering gene function and regulatory pathways are experiments with interventions into the normal course of action in a cell. A common technique is to perturb a gene of interest experimentally and to study which other genes' activity or phenotypic features are effected. Bayesian networks present a prominent approach to derive a theoretical model from these experiments (Pe'er et al., 2001; Yoo et al., 2002; Friedman, 2004): genes are represented by vertices of a network and the task is to find a topology, which explains dependencies between the genes. When learning from observational data only, groups of Bayesian networks may be statistically indistinguishable (Verma and Pearl, 1990). Information about effects of an intervention helps to resolve such equivalence classes by including causal knowledge into the model (Tian and Pearl, 2001). The final goal is to learn a graph structure which not only represents statistical dependencies, but also causal relations between genes.

Manipulating the expression level of a gene can be done in a variety of ways (Alberts et al., 2002). A gene's expression level can be down-regulated by several techniques including

1. creating animals or cell lines in which the gene is non-functional. This is called a *knockout.*
2. exposing a cell or animal to environmental stress to inhibit the function of certain genes or proteins.
3. partially destroying the RNA transcribed from the gene which itself is left intact. This is the recently introduced method of *RNA interference* (RNAi).

All three examples have in common that the gene's expression level is *pushed* towards a "no expression" state. Only in the first example, however, the intervention leads to a completely unfunctional gene. In RNAi the gene is still active, but silenced. It is less active than normal due to human intervention. Hence, we do not fix the state of the gene, but push it towards lower activities. In addition this pushing is randomized to some extent: the experimentalist knows that he has silenced the gene, but he can not say exactly by how much.

It is crucial that models reflect the way data was generated in the perturbation experiments. In Bayesian structure learning, Tian and Pearl (2001) show that interventions can be modeled by imposing different parameter priors when the gene is actively perturbed or passively observed. They only distinguish between two kinds of interventions: most generally, interventions that change the local probability distribution of the node within a given family of distributions, and as a special case, interventions that fix the state of the variable deterministically. The first is called a *mechanism change*; it does not assume any prior information on *how* the local probability distribution changes. The second type of intervention, which fixes the state of the variable, is called a *do-operation* and is treated in detail in (Pearl, 2000; Spirtes et al., 2000). A do-operation is used in almost all applications of interventional learning in Bayesian networks (e.g. Yoo and Cooper, 2003; Yoo et al., 2002; Steck and Jaakkola, 2002; Tong and Koller, 2001; Pe'er at al., 2001; Murphy, 2001; Cooper, 2000; Cooper and Yoo, 1999).

To model biological experiments as described above we focus on interventions, which specifically concentrate the local distribution at a certain node around some target state. We will call them *pushing interventions*, they are examples of mechanism changes with prior knowledge. The do-operator is a special case of a pushing intervention, which we call a *hard intervention*. In this paper, we generalize hard interventions to *soft interventions*: The local probability distribution only centers more around the target value without being fixed. This generalization is necessary to cope with experiments as in the gene perturbation examples 2 and 3 above. If we treat them as unfocussed mechanism changes we lose valuable information about what kind of intervention was performed. Thus, we need a concept of interventions, which is more directed than general mechanism changes, but still softer than deterministic fixing of variables.

The goal of the paper is to develop a theory for learning a Bayesian network when data from different (hard or soft) pushing interventions of the network is available. We first explain how soft interventions can be modeled by changing the prior distribution in Section 2. A soft intervention can be realized by introducing a "pushing parameter", which captures the pushing strength. We propose a concrete parametrization of the pushing parameter in the classical cases of discrete and Gaussian networks. Hard interventions, which have been formally described by choosing a Dirac prior in (Tian and Pearl, 2001), can then be interpreted as infinite pushing.

Section 3 summarizes the results in the general setting of Conditional Gaussian networks. This extends the existing theory on learning with hard interventions in discrete networks to learning with soft interventions in networks containing discrete and Gaussian variables.

The concluding Section 4 deals with *probabilistic* soft interventions: in this set-up the pushing parameter becomes a random variable and we assign a hyperprior to it. Hence, we account for the experimentalists lack of knowledge on the actual strength of intervention by weighted averaging over all possible values.

2 Pushing interventions in Bayesian networks

A Bayesian network is a graphical representation of the dependency structure between the components of a random vector $\mathbf{X}$. The individual random variables are associated with the vertices of a directed acyclic graph (DAG) D, which describes the dependency structure. Once the states of its parents are given, the probability distribution of a given node is fixed. Thus, the Bayesian network is completely specified by the DAG and the local probability distributions (LPDs). Although this definition is quite general, there are basically three types of Bayesian networks which are used in practice: discrete, Gaussian and Conditional Gaussian (CG) networks. CG networks are a combination of the former two and will be treated in more detail in Section 3, for the rest of this section we focus on discrete and Gaussian networks.

In discrete and Gaussian networks, LPDs are taken from the family of the multinomial and normal distribution, respectively. In the theory of Bayesian structure learning, the parameters of these distributions are not fixed, but instead a prior distribution is assumed (Cooper and Herskovits, 1992; Geiger and Heckerman, 1994; Bøttcher, 2004). The priors usually chosen because of conjugacy are the Dirichlet distribution in the discrete case and the Normal-inverse-χ^2 distribution in the Gaussian case. Averaging the likelihood over these priors yields the marginal likelihood – the key quantity in structure learning (see Section 3).

An intervention at a certain node in the network can in this setting easily be modeled by a change in the LPDs' prior. When focusing on (soft) pushing interventions, this change should result in an increased concentration of the node's LPD around the target value. We model this concentration by introducing a pushing parameter w which is meant to measure the strength of the pushing — a higher value of w results in a stronger concentration of the LPD. We now explain in more detail how this is done for discrete and Gaussian networks. Since the joint distribution $p(\mathbf{x})$ in a Bayesian network factors according to the DAG structure in terms only involving a single node and its parents, it will suffice to concentrate on one such family of nodes.

2.1 Pushing by Dirichlet priors

We denote the set of discrete nodes by Δ and a discrete random variable at node $\delta \in \Delta$ by I_δ. The set of possible states of I_δ is $\mathcal{I}_\delta$. The parametrization of the discrete LPD at node δ is called θ_δ. For every configuration $\mathbf{i}_{pa(\delta)}$ of parents, θ_δ contains a vector of probabilities for each state $i_\delta \in \mathcal{I}_\delta$. Realizations of discrete random variables are multinomially distributed with parameters depending on the state of discrete parents. The conjugate prior is Dirichlet with parameters also depending on the state of discrete parents:

$$\begin{aligned} I_\delta \mid \mathbf{i}_{pa(\delta)}, \theta_\delta &\sim \text{Multin}(1, \theta_{\delta|\mathbf{i}_{pa(\delta)}}), \\ \theta_{\delta|\mathbf{i}_{pa(\delta)}} &\sim \text{Dirichlet}(\alpha_{\delta|\mathbf{i}_{pa(\delta)}}). \end{aligned} \quad (1)$$

We assume that the $\alpha_{\delta|\mathbf{i}_{pa(\delta)}}$ are chosen to respect likelihood equivalence as in (Heckerman et al., 1995). Doing a pushing intervention at node δ amounts to changing the prior parameters such that the multinomial density concentrates at some target value j. We for-

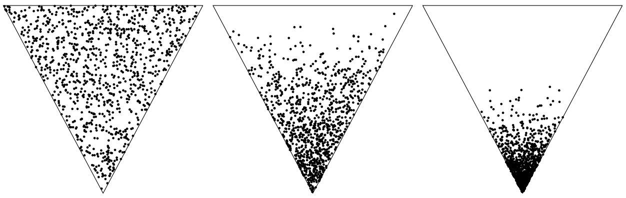

Figure 1: Examples of pushing a discrete variable with three states. Each triangle represents the sample space of the three-dimensional Dirichlet distribution (which is the parameter space of the multinomial likelihood of the node). The left plot shows a uniform distribution with Dirichlet parameter $\alpha = (1, 1, 1)$. The other two plots show effects of pushing with increasing weight: $w = 3$ in the middle and $w = 10$ at the right. In each plot 1000 points were sampled.

malize this by introducing a pushing operator $\mathcal{P}$ defined by

$$\mathcal{P}(\alpha_{\delta|\mathbf{i}_{pa(\delta)}}, w_\delta, j) \;=\; \alpha_{\delta|\mathbf{i}_{pa(\delta)}} + w_\delta \cdot \mathbf{1}_j, \qquad (2)$$

where $\mathbf{1}_j$ is a vector of length $|\mathcal{I}_\delta|$ with all entries zero except for a single 1 at state j. The pushing parameter $w_\delta \in [0, \infty]$ determines the strength of intervention at node δ: if $w_\delta = 0$ the prior remains unchanged, if $w_\delta = \infty$ the Dirichlet prior degenerates to a Dirac distribution and fixes the LPD to the target state j. Figure 1 shows a three-dimensional example of increasing pushing strength w_δ.

2.2 Pushing by Normal-inverse-χ^2 priors

The set of Gaussian nodes will be called Γ and we denote a Gaussian random variable at node $\gamma \in \Gamma$ by Y_γ. In the purely Gaussian case it depends on the values of parents $\mathbf{Y}_{pa(\gamma)}$ via a vector of regression coefficients β_γ. If we assume that β_γ contains a first entry $\beta_\gamma^{(0)}$, the parent-independent contribution of Y_γ, and attach to $\mathbf{Y}_{pa(\gamma)}$ a leading 1, we can write for Y_γ the standard regression model (Bøttcher, 2004):

$$\begin{aligned} Y_\gamma \mid \beta_\gamma, \sigma_\gamma^2 &\sim \mathrm{N}(\mathbf{Y}_{pa(\gamma)}^\top \beta_\gamma,\ \sigma_\gamma^2), \\ \beta_\gamma \mid \sigma_\gamma^2 &\sim \mathrm{N}(\mathbf{m},\ \sigma_\gamma^2 \mathbf{M}^{-1}), \\ \sigma_\gamma^2 &\sim \text{Inv-}\chi^2(\nu,\ s^2). \end{aligned} \qquad (3)$$

We assume that the prior parameters $\mathbf{m}, \mathbf{M}, \nu, s^2$ are chosen as in (Bøttcher, 2004). To push Y_γ to a value k we exchange $\mathbf{m}$ and s^2 by $(\mathbf{m}', s'^2) = \mathcal{P}((\mathbf{m}, s^2), w_\gamma, k)$ defined by

$$\begin{aligned} \mathbf{m}' &= e^{-w_\gamma} \cdot \mathbf{m} + (1 - e^{-w_\gamma}) \cdot k\mathbf{1}_1, \\ s'^2 &= s^2/(w_\gamma + 1), \end{aligned} \qquad (4)$$

where $k\mathbf{1}_1$ is a vector of length $|\mathbf{i}_{pa(\gamma)}| + 1$ with all entries zero except the first, which is k. We use $\mathcal{P}$ for the pushing operator as in the case of discrete nodes; which one to use will be clear from the context. Again $w_\gamma \in [0, \infty]$ represents intervention strength.

The exponential function maps the real valued w into the interval $[0, 1]$. The exponential decay towards 0 ensures that by increasing w interventions quickly gain in strength. The interventional prior mean $\mathbf{m}'$ is a convex combination of the original mean $\mathbf{m}$ with a "pushing" represented by $k\mathbf{1}_1$. If $w = 0$ the mean of the normal prior and the scale of the inverse-χ^2 prior remain unchanged. As $w \to \infty$ the scale s'^2 goes to 0, so the prior for σ^2 tightens at 0. At the same time, the regression coefficients of the parents converge to 0 and β_0 goes to value k. All in all, with increasing w the distribution of Y_γ peaks more and more sharply at $Y_\gamma = k$.

Note that the discrete pushing parameter w_δ and the Gaussian pushing parameter w_γ live on different scales and will need to be calibrated individually.

2.3 Hard pushing

Hard pushing means to make sure that a certain node's LPD produces almost surely a certain target value. It has been proposed by Tian and Pearl (2001) to model this by imposing a Dirac prior on the LPD of the node. Although the Dirac prior is no direct member of neither the Dirichlet nor the Normal-inverse-χ^2 family of distributions it arises for both of them when taking the limit $w \to \infty$ for the pushing strength. Tian and Pearl (2001) give an example for discrete networks, which can easily be extended to Gaussian networks by

$$p(\beta_\gamma, \sigma_\gamma^2 \mid \mathrm{do}(Y_\gamma = k)) = \\ d(\beta_\gamma^{(0)} - k) \prod_{i \in pa(\gamma)} d(\beta_\gamma^{(i)}) \cdot d(\sigma_\gamma^2). \qquad (5)$$

Here, $d(\cdot)$ is the Dirac function. Averaging over this prior sets the variance and the regression coefficients to zero, while $\beta_\gamma^{(0)}$ is set to k. Thus, the marginal distribution of Y_γ is fixed to state k with probability one.

2.4 Modeling interventions by policy variables

Hard interventions can be modeled by introducing a policy variable as an additional parent node of the variable at which the intervention is occuring (Pearl, 2000; Spirtes et al., 2000; Lauritzen, 2000). In the same way we can use policy variables to incorporate soft interventions. For each node v, we introduce an additional parent node F_v ("F" for "force"), which is keeping track of whether an intervention was performed at X_v or not, and if yes, what the target state was. For a discrete variable I_δ, the policy variable F_δ has state space $\mathcal{I}_\delta \cup \emptyset$ and we can write

$$p(\theta_{\delta|\mathbf{i}_{pa(\delta)},f_\delta}) = \\ = \begin{cases} \text{Dirichlet}(\alpha_{\delta|\mathbf{i}_{pa(\delta)}}) & \text{if } F_\delta = \emptyset, \\ \text{Dirichlet}(\alpha'_{\delta|\mathbf{i}_{pa(\delta)}}) & \text{if } F_\delta = j, \end{cases} \quad (6)$$

where $\alpha'_{\delta|\mathbf{i}_{pa(\delta)}} = \mathcal{P}(\alpha_{\delta|\mathbf{i}_{pa(\delta)}}, w_\delta, j)$ is derived from $\alpha_{\delta|\mathbf{i}_{pa(\delta)}}$ as defined in Eq. 2. For a continuous variable Y_γ, the policy variable F_γ has state space $\mathrm{I\!R} \cup \emptyset$ and we can write

$$p(\beta_{\gamma|f_\gamma}, \sigma^2_{\gamma|f_\gamma}) = \\ = \begin{cases} \text{N}(\mathbf{m}, \mathbf{M}) \cdot \text{Inv-}\chi^2(\nu, s^2) & \text{if } F_\gamma = \emptyset, \\ \text{N}(\mathbf{m}', \mathbf{M}) \cdot \text{Inv-}\chi^2(\nu, s'^2) & \text{if } F_\gamma = k, \end{cases} \quad (7)$$

where $(\mathbf{m}', s'^2) = \mathcal{P}((\mathbf{m}, s^2), w_\gamma, k)$ as defined in Eq. 4. Equations 6 and 7 will be used in section 3.2 to compute the marginal likelihood of Conditional Gaussian networks from a mix of interventional and non-interventional data.

3 Pushing in Conditional Gaussian networks

In this section we summarize the results in the general framework of Conditional Gaussian networks and compute a scoring metric for learning from soft interventions.

3.1 Conditional Gaussian networks

Conditional Gaussian (CG) networks are Bayesian networks encoding a joint distribution over discrete and continuous variables. We consider a random vector $\mathbf{X}$ splitting into two subsets: $\mathbf{I}$ containing discrete variables and $\mathbf{Y}$ containing continuous ones. The dependencies between individual variables in $\mathbf{X}$ can be represented by a directed acyclic graph (DAG) D with node set V and edge set E. The node set V is partitioned as $V = \Delta \cup \Gamma$ into nodes of discrete (Δ) and continuous (Γ) type. Each discrete variable corresponds to a node in Δ and each continuous variable to a node in Γ. The distribution of a variable X_v at node v only depends on variables $\mathbf{X}_{pa(v)}$ at parent nodes $pa(v)$. Thus, the joint density $p(\mathbf{x})$ decomposes as

$$p(\mathbf{x}) = p(\mathbf{i}, \mathbf{y}) = p(\mathbf{i})p(\mathbf{y}|\mathbf{i}) \\ = \prod_{\delta \in \Delta} p(i_\delta|\mathbf{i}_{pa(\delta)}) \cdot \prod_{\gamma \in \Gamma} p(y_\gamma|\mathbf{y}_{pa(\gamma)}, \mathbf{i}_{pa(\gamma)}). \quad (8)$$

The discrete part, $p(\mathbf{i})$, is given by an unrestricted discrete distribution. The distribution of continuous random variables given discrete variables, $p(\mathbf{y}|\mathbf{i})$, is multivariate normal with mean and covariance matrix depending on the configuration of discrete variables. Since discrete variables do not depend on continuous variables, the DAG D contains no edges from nodes in Γ to nodes in Δ.

For discrete nodes, the situation in CG networks is exactly the same as in the pure case discussed in Section 2: The distribution of $I_\delta|\mathbf{i}_{pa(\delta)}$ is multinomial and parametrized by θ_δ. Compared to the purely Gaussian case treated in Section 2, we have for Gaussian nodes in CG networks an additional dependency on discrete parents. This dependency shows in the regression coefficients and the variance, which now not only depend on the node, but also on the state of the discrete parents:

$$Y_\gamma \mid \beta_{\mathbf{i}_{pa(\gamma)}}, \sigma^2_{\mathbf{i}_{pa(\gamma)}} \sim \text{N}(\mathbf{Y}^\top_{pa(\gamma)}\beta_{\mathbf{i}_{pa(\gamma)}},\ \sigma^2_{\mathbf{i}_{pa(\gamma)}}). \quad (9)$$

As a prior distribution we again take the conjugate normal-inverse-χ^2 distribution as in Eq. 3. For further details on CG networks we refer to (Lauritzen, 1996; Bøttcher, 2004).

3.2 Learning from interventional and non-interventional data

Assuming an uniform prior over network structures D, the central quantity to be calculated is the *marginal likelihood* $p(d|D)$ (Heckerman et al., 1995). In the case of only one type of data it can be written as

$$p(d|D) = \int_\Theta p(d|D,\theta)p(\theta|D)\ \mathrm{d}\theta. \quad (10)$$

Here $p(\theta|D)$ is the prior on the parameters θ of the LPDs. If the dataset contains both interventional and non-interventional cases, the basic idea is to choose parameter priors locally for each node as in Eq. 6 and 7 according to whether a variable was intervened in a certain case or not. We will see that this strategy effectively leads to a local split of the marginal likelihood into an interventional and a non-interventional part.

3.2.1 A family-wise view of marginal likelihood

To compute the marginal likelihood of CG networks on interventional and non-interventional data, we rewrite Eq. 10 in terms of single nodes such that the theory of (soft) pushing from Section 2 can be used. In the computation we will use the following technical utilities:

1. The dataset d consists of N cases $\mathbf{x}^1, \ldots, \mathbf{x}^N$, which are sampled independently. Thus we can write $p(d|D,\theta)$ as a product over all single case likelihoods $p(\mathbf{x}^c|D,\theta)$, $c = 1, \ldots, N$.

2. The joint density $p(\mathbf{x})$ factors according to the DAG D as in Eq. 8. Thus for each case $\mathbf{x}^c$ we can write $p(\mathbf{x}^c|D,\theta)$ as a product over node contributions $p(x_v^c|\mathbf{x}_{pa(v)}^c,\theta_v)$ for all $v \in V$.

3. We assume *parameter independence*: the parameters associated with one variable are independent of the parameters associated with other variables, and the parameters are independent for each configuration of the discrete parents (Heckerman et al., 1995) This allows us to decompose the prior $p(\theta|D)$ in Eq. 10 into node-wise priors $p(\theta_{v|\mathbf{i}_{pa(v)}}|D)$ for a given parent configuration $\mathbf{i}_{pa(v)}$.

4. All interventions are soft pushing. For a given node, intervention strength and target state stay the same in all cases in the data, but of course different nodes may have different pushing strengths and target values. This constraint just helps us to keep the following formulas simple and can easily be dropped.

These four assumptions allow a family-wise view of the marginal likelihood. Before we present it in a formula, it will be helpful to introduce a *batch notation*. In CG networks, the parameters of the LPD at a certain node depend only on the configuration of discrete parents. This holds for both discrete and Gaussian nodes. Thus, when evaluating the likelihood of data at a certain node, it is reasonable to collect all cases in a batch, which correspond to the same parent configuration:

$$\begin{aligned} & p(d|D,\theta) \\ &= \prod_{c\in d} p(\mathbf{x}^c|D,\theta) \;=\; \prod_{c\in d}\prod_{v\in V} p(x_v^c|\mathbf{x}_{pa(v)}^c,\theta_v) \\ &= \prod_{v\in V}\prod_{\mathbf{i}_{pa(v)}}\prod_{c:\mathbf{i}_{pa(v)}^c=\mathbf{i}_{pa(v)}} p(x_v^c|\mathbf{i}_{pa(v)}^c,\mathbf{y}_{pa(v)},\theta_v) \end{aligned} \tag{11}$$

The last formula is somewhat technical: If the node v is discrete, then $\mathbf{y}_{pa(v)}$ will be empty, and usually not all parent configuration $\mathbf{i}_{pa(v)}$ are found in the data, so some terms of the product will be missing.

For each node we will denote the cases with the same joint parent state by $B_{\mathbf{i}_{pa(v)}}$. When learning with interventional data, we have to distinguish further between observations of a variable which were obtained passively and those that are result of intervention. Thus, for each node v we split the batch $B_{\mathbf{i}_{pa(v)}}$ into one containing all observational cases and one containing the interventional cases:

$$\begin{aligned} B^{obs}_{\mathbf{i}_{pa(v)}} = \{c \in d \;&: \mathbf{i}^c_{pa(v)} = \mathbf{i}_{pa(v)} \\ &\text{and no intervention at } v\}, \\ B^{int}_{\mathbf{i}_{pa(v)}} = \{c \in d \;&: \mathbf{i}^c_{pa(v)} = \mathbf{i}_{pa(v)} \\ &\text{and intervention at } v\}. \end{aligned}$$

If there is more than one type of intervention applied to node v, the batch containing interventional cases has to be split accordingly. Using this notation we can now write down the marginal likelihood for CG networks in terms of single nodes and parents:

$$\begin{aligned} & p(d|D) = \\ & \prod_{v\in V}\prod_{\mathbf{i}_{pa(v)}}\int_\Theta \prod_{B^{obs}_{\mathbf{i}_{pa(v)}}} p(x_v^o|\mathbf{i}_{pa(v)},\mathbf{y}^o_{pa(v)},\theta_v)p'(\theta_v|D)\,\mathrm{d}\theta_v \;\times \\ & \prod_{v\in V}\prod_{\mathbf{i}_{pa(v)}}\int_\Theta \prod_{B^{int}_{\mathbf{i}_{pa(v)}}} p(x_v^e|\mathbf{i}_{pa(v)},\mathbf{y}^e_{pa(v)},\theta_v)p''(\theta_v|D,w_v)\,\mathrm{d}\theta_v. \end{aligned} \tag{12}$$

At each node, we use distributions and priors as defined in Eq. 6 for discrete nodes and Eq. 7 for Gaussian nodes. The non-interventional prior p' corresponds to $F_v = \emptyset$ and the interventional prior p'' corresponds to F_v equalling some target value. We denoted the intervention strength explicitly in the formula, since we will focus on it further when discussing *probabilistic* soft interventions in Section 4.

Equation 12 consists of an observational and an interventional part. Both can further be split into a discrete and a Gaussian part, so we end up with four terms to consider.

3.2.2 Discrete observational part

To write down the marginal likelihood of discrete observational data, we denote by $n_{i_\delta|\mathbf{i}_{pa(\delta)}}$ the number of times we passively observe $I_\delta = i_\delta$ in batch $B^{obs}_{\mathbf{i}_{pa(\delta)}}$, and by $\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}}$ the corresponding pseudo-counts of the Dirichlet prior. Summation of $\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}}$ and $n_{i_\delta|\mathbf{i}_{pa(\delta)}}$ over all $i_\delta \in \mathcal{I}_\delta$ is abbreviated by $\alpha_{\mathbf{i}_{pa(\delta)}}$ and $n_{\mathbf{i}_{pa(\delta)}}$, respectively. Then, the marginal likelihood of the discrete data d_Δ can be written as

$$\begin{aligned} p(d_\Delta \mid D) = \prod_{\delta\in\Delta}\prod_{\mathbf{i}_{pa(\delta)}} \Bigg(& \frac{\Gamma(\alpha_{\mathbf{i}_{pa(\delta)}})}{\Gamma(\alpha_{\mathbf{i}_{pa(\delta)}} + n_{\mathbf{i}_{pa(\delta)}})} \times \\ & \prod_{i_\delta\in\mathcal{I}_\delta} \frac{\Gamma(\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}} + n_{i_\delta|\mathbf{i}_{pa(\delta)}})}{\Gamma(\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}})} \Bigg), \end{aligned} \tag{13}$$

This result was first obtained by Cooper and Herskovits (1992) and is further discussed in (Heckerman et al., 1995).

3.2.3 Discrete interventional part

Since interventions are just changes in the prior, the marginal likelihood of the interventional part of discrete data is of the same form as Eq. 13. The prior parameters $\alpha_{i_\delta|\mathbf{i}_{pa(\delta)}}$ are exchanged by $\alpha'_{i_\delta|\mathbf{i}_{pa(\delta)}} =$

$\mathcal{P}(\alpha_{i\delta|\mathbf{i}_{pa(\delta)}}, w_\delta, j)$ as given by Eq. 2, and the counts $n_{i\delta|\mathbf{i}_{pa(\delta)}}$ are exchanged by $n'_{i\delta|\mathbf{i}_{pa(\delta)}}$ taken from batch $B^{int}_{\mathbf{i}_{pa(\delta)}}$.

In the limit $w_\delta \to \infty$ this part converges to one and vanishs from the overall marginal likelihood $p(d|D)$. This special case was already shown in (Cooper and Yoo, 1999; Tian and Pearl, 2001).

3.2.4 Gaussian observational part

All cases y_γ in batch $B^{obs}_{\mathbf{i}_{pa(\gamma)}}$ are sampled independently from a normal distribution with fixed parameters. If we gather them in a vector $\mathbf{y}_\gamma$ and the corresponding states of continuous parents as rows in a matrix $\mathbf{P}_\gamma$, we yield the standard regression scenario

$$\mathbf{y}_\gamma \sim \mathrm{N}(\mathbf{P}_\gamma \beta_\gamma, \sigma^2_\gamma \mathbf{I}), \tag{14}$$

where $\mathbf{I}$ is the identity matrix. As a prior distribution we choose normal-inverse-χ^2 as shown in Eq. 3. Marginalizing with respect to β_γ and σ^2_γ yields a multivariate t-distribution of dimension $|B^{obs}_{\mathbf{i}_{pa(\gamma)}}|$, with location vector $\mathbf{Pm}$, scale matrix $s(\mathbf{I} + \mathbf{PM}^{-1}\mathbf{P}^\top)$, and ν degrees of freedom. The density function can be found in many textbooks (e.g. Gelman et al., 1996).

When using data from different batches, every parameter above carries an index "$\mathbf{i}_{pa(\gamma)}$" indicating that it depends on the state of the discrete parents of the Gaussian node γ. Multiplying t-densities for all nodes and configurations of discrete parents—the outer double-product in Eq. (12)—yields the marginal likelihood of the Gaussian part. See Bøttcher (2004) for details.

3.2.5 Gaussian interventional part

Here we consider cases y_γ in batch $B^{int}_{\mathbf{i}_{pa(\gamma)}}$. We collect them in a vector and can again write a regression model like in Eq. 14. The difference to the observational Gaussian case lies in the prior parameters. They are now given by Eq. 4. The result of marginalization is again a t-density with parameters as above, just $\mathbf{m}, s$ are exchanged by $(\mathbf{m}', s') = \mathcal{P}((\mathbf{m}, s), w_\gamma, k)$. The Gaussian interventional part is then given by a product of such t-densities over nodes and discrete parent configurations.

If we use the hard intervention prior in Eq. 5 instead, the Gaussian interventional part integrates to one and vanishs from the marginal likelihood in Eq. (12). This is the extension of the results in (Cooper and Yoo, 1999) to Gaussian networks.

4 Probabilistic soft interventions

In Section 2 we introduced the pushing operator $\mathcal{P}(\cdot, w_v, t_v)$ to model a soft intervention at a discrete or Gaussian node v. The intervention strength w_v is a parameter, which has to be chosen before network learning. There are several possibilities, how to do it:

- If there is solid experimental experience on how powerful interventions are, this can be reflected in an appropriate choice of w_v. An obvious problem is that w_v needs to be determined on a scale that is compatible with the Bayesian network model.

- If there is prior knowledge on parts of the network topology, the parameter w_v can be tuned until the result of network learning fits the prior knowledge.

Note again that by the parametrization of pushing given in Section 2, the pushing strengths for discrete and Gaussian nodes live on different scales and have to be calibrated separately.

However, a closer inspection of the biological experiments, which motivated the theory of soft pushing interventions, suggests to treat the intervention strength w_v as a random variable: In gene silencing an inhibiting molecule (a double-stranded RNA in case of RNAi) is introduced into the cell. This usually works in a high percentage of affected cells. In the case of success, the inhibitor still has to spread throughout the cell to silence the target gene. This diffusion process is stochastic and consequently causes experimental variance in the strength of the silencing effect.

These observations suggest to assign a prior distribution $p(w_v)$ to the intervention strength. That is, we drop the assumption of having one intervention strength in all cases, but instead average over possible values of w_v. For simplicity we assume there is only a limited number of possible values of w_v, say, $w_v^{(1)}, \ldots, w_v^{(k)}$, with an arbitrary discrete distribution assigned to them. Then we can express our inability to control the pushing strength in the experiment deterministically by using a mixed prior of the form:

$$p(\theta_v|D) = \sum_{i=1}^{k} q_k \, p(\theta_v|D, w_v^{(k)}). \tag{15}$$

Here, the mixture coefficients $q_k = p(w_v^{(k)})$ are the prior probabilities of each possible pushing strength. The terms $p(\theta_v|D, w_v^{(k)})$ correspond to Dirichlet densities in the discrete case and Normal-inverse-χ^2 densities in the Gaussian case. In RNAi experiments, $w_v^{(1)}, \ldots, w_v^{(k)}$ can be estimated from the empirical distribution of measured RNA degradation efficiencies in repeated assays.

Mixed priors as in Eq. 15 are often used in biological sequence analysis to express prior knowledge which is not easily forced into a single distribution. See (Durbin et al., 1998) for details.

If we substitute the prior $p''(\theta_v|D, w_v)$ in the interventional part of Eq. 12 with the mixture prior in Eq. 15, the marginal likelihood of a family of nodes is a mixture of marginal likelihoods corresponding to certain values $w_v^{(k)}$ weighted by mixture coefficients q_k.

5 Conclusion

Our work extends structure learning from interventional data into two directions: from learning discrete networks to learning mixed networks and from learning with hard interventions to learning with soft interventions.

Soft interventions are focussed on a specific target value of the variable of interest and concentrate the local probability distribution there. We proposed parametrizations for pushing discrete and continuous variables using Dirichlet and Normal-inverse-χ^2 priors, respectively.

We computed the marginal likelihood of CG networks for data containing both observational and (soft) interventional cases. In Bayesian structure learning, the marginal likelihood is the key quantity to compute from data. Using it (and possibly a prior over network structures) as a scoring function, we can start model search over possible network structures. For networks with more than 5 nodes, exhaustive search becomes infeasible; often used search heuristics include hill climbing or MCMC methods. For a survey see Heckerman et al. (1995) and references therein.

Since in biological settings the pushing strength is unknown we proposed using a mixture hyperprior on it, resulting in a mixture marginal likelihood. This makes the score for each network more time-consuming to compute. Searching in the space of DAGs may become infeasible even with quick search heuristics. But in applications there is often a large amount of biological prior knowledge, which limits the number of pathway candidates from the beginning. When learning network structure we usually don't have to optimize the score over the space of all possible DAGs but are limited to a few candidate networks, which are to be compared. This corresponds to a very rigid structure prior.

Due to measurement error or noise inherent in the observed system it may often happen that a variable, at which an intervention took place, is observed in a state different from the target state. In the hard intervention framework, a single observation of this kind results in a marginal likelihood of zero. Modeling interventions as soft pushing mends this problem and makes structure learning more robust against noise. This is a central benefit of our approach.

Acknowledgements

This work was done within the context of the Berlin Center for Genome Based Bioinformatics (BCB), part of the German National Genome Research Network (NGFN) and supported by BMBF grants 031U109C and 03U117. We thank Dennis Kostka as well as the rest of CMB for helpful discussions.

References

B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell.* Garland Science, New York, 4 edition, 2002.

S. G. Bøttcher. *Learning Bayesian Networks with Mixed Variables.* PhD thesis, Aalborg University, Denmark, 2004.

G. F. Cooper. A Bayesian Method for Causal Modeling and Discovery Under Selection. In *Proceedings of UAI 16*, pages 98–106, 2000.

G. F. Cooper and E. Herskovits. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning*, 9:309–347, 1992.

G. F. Cooper and C. Yoo. Causal discovery from a mixture of experimental and observational data. In *Proceedings of UAI 15*, pages 116–125, 1999.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis.* Cambridge University Press, 1998.

N. Friedman. Inferring Cellular Networks Using Probabilistic Graphical Models. *Science*, 303(5659):799–805, 2004.

D. Geiger and D. Heckerman. Learning Gaussian Networks. In *Proceedings of UAI 10*, pages 235–243, 1994.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis.* Chapman and Hall-CRC, 1996.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20(3):197–243, Sep. 1995.

S. L. Lauritzen. *Graphical Models.* Clarendon Press, Oxford, 1996.

S. L. Lauritzen. Causal Inference from Graphical Models. In: O. E. Barndorff-Nielsen, D. R. Cox, and C. Klüppelberg (eds.) *Complex Stochastic Systems.* Chapman and Hall, London, 2000.

K. P. Murphy. Active Learning of Causal Bayes Net Structure, *Tech. Rep., UC Berkeley*, 2001.

J. Pearl. *Causality: Models, Reasoning and Inference.* Cambridge University Press, Cambridge, 2000.

D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(90001):S215–S224, 2001.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* MIT Press, Cambridge, MA, second edition, 2000.

H. Steck and T. S. Jaakkola. Unsupervised active learning in large domains. In *Proceedings of UAI 18*, pages 469–476, 2002.

J. Tian and J. Pearl. Causal discovery from changes: a Bayesian approach. In *Proceedings of UAI 17*, pages 512–521, 2001.

S. Tong and D. Koller. Active Learning for Structure in Bayesian Networks. In *Proceedings of IJCAI 17*, 2001.

T. S. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of UAI 6*, pages 255–268, 1990.

C. Yoo and G. F. Cooper. An evaluation of a system that recommends microarray experiments to perform to discover gene-regulation pathways. *Journal Artificial Intelligence in Medicine*, 2003.

C. Yoo, V. Thorsson, and G. F. Cooper. Discovery of causal relationships in a generegulation pathway from a mixture of experimental and oberservational DNA microarray data. In *Proceedings of PSB 7*, pages 498-509, 2002.