BAYESIAN STATISTICS 9,
J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid,
D. Heckerman, A. F. M. Smith and M. West (Eds.)
© Oxford University Press, 2010

Nonparametric Bayesian Networks

Katja Ickstadt[1], Björn Bornkamp[1], Marco Grzegorczyk[1], Jakob Wieczorek[1]
M. Rahuman Sheriff[2], Hernán E. Grecco[2] & Eli Zamir[2]
[1] *TU Dortmund University, Germany*
[2] *Max-Planck Institute of Molecular Physiology, Dortmund, Germany*
ickstadt@statistik.tu-dortmund.de

Summary

A convenient way of modelling complex interactions is by employing graphs or networks which correspond to conditional independence structures in an underlying statistical model. One main class of models in this regard are Bayesian networks, which have the drawback of making parametric assumptions. Bayesian nonparametric mixture models offer a possibility to overcome this limitation, but have hardly been used in combination with networks. This manuscript brigdes this gap by introducing nonparametric Bayesian network models. We review (parametric) Bayesian networks, in particular Gaussian Bayesian networks, from a Bayesian perspective as well as nonparametric Bayesian mixture models. Afterwards these two modelling approaches are combined into nonparametric Bayesian networks. The new models are compared both to Gaussian Bayesian networks and to mixture models in a simulation study, where it turns out that the nonparametric network models perform favorably in non Gaussian situations. The new models are also applied to an example from systems biology.

Keywords and Phrases: Gaussian Bayesian networks; Systems Biology; Nonparametric Mixture Models; Species Sampling Models

1. INTRODUCTION

Complex interactions are of increasing importance in many research areas like information retrieval, engineering, decision support systems and systems biology. A convenient way of modelling such complex interactions are graphs, which correspond to conditional independence structures in the underlying statistical model. In this context graphs appear in two main flavors: graphs containing only undirected or only

This work has been supported by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung) in the project "Spatially resolved reverse engineering of protein networks in cell-matrix adhesions" as well as by the Research Training Group "Statistical modelling" of the German Research Foundation (Deutsche Forschungsgemeinschaft).

directed edges. The most prominent Bayesian statistical models based on undirected graph structures are Gaussian graphical models (see, for example, Giudici (1996) or more recently Carvalho and Scott (2010)). A limitation of undirected models is the fact that is not possible to learn the direction of dependencies (*i.e.* causal dependencies), which is of major importance, for example, in systems biology.

Prominent statistical models based on directed graphs are Bayesian networks. The underlying graph is a so-called directed acyclic graph (DAG) with nodes representing random variables and edges coding the conditional independence structure. Bayesian network methodology was proposed and developed by Pearl (1985), and following Pearl's book (Pearl (1988)) Bayesian networks have been used for modelling complex conditional (in-)dependencies among variables in various fields of research. Bayesian networks are interpretable and fairly flexible models for representing probabilistic relationships among interacting variables. In the seminal paper by Friedman et al. (2000) Bayesian networks were applied to infer gene regulatory networks from gene expression data in systems biology research. Since then Bayesian network models have been developed further, and nowadays Bayesian networks can be seen as one of the most popular tools in systems biology research for reverse engineering regulatory networks and cellular signalling pathways from a variety of types of postgenomic data. Fast Markov Chain Monte Carlo (MCMC) algorithms, like those developed in Friedman and Koller (2003) or Grzegorczyk and Husmeier (2008), can be applied to systematically search the space of network structures for those that are most consistent with the data. A closed-form expression of the marginal likelihood can be obtained for two probabilistic models with their respective conjugate prior distributions: the multinomial distribution with the Dirichlet prior (BDe) (Cooper and Herskovits (1992)) and the linear Gaussian distribution with the normal-Wishart prior (BGe) (Geiger and Heckerman, 1994)). However these two standard approaches are restricted in that they either require the data to be discretized (BDe) or can only capture linear regulatory relationships (BGe). The BGe model makes an implicit assumption of multivariate normality for the data and in real-world applications this assumption is often violated. On the other hand, data discretisation always incurs an information loss so that the discrete BDe model cannot be seen a sufficient remedy. One extension to overcome these limitations of the BGe model is the mixture model of Grzegorczyk et al. (2008). In this paper we generalize this model and consider it in a broader framework of Bayesian nonparametric mixture models.

Interest in Bayesian nonparametric mixture models started with the publication of Ferguson (1973) on the Dirichlet process. While early literature was mainly confined to relatively simple conjugate models, the advent of MCMC (see, among others, Escobar and West (1995)) and positive asymptotic properties (Ghosh and Ramamoorthi, 2003), renewed practical and theoretical interest in the field. Nonparametric, *i.e.* infinite, mixture models employ discrete random probability measures (*i.e.* stochastic processes) for the mixing distribution, see, for example, Ongaro and Cattaneo (2004) or James, Lijoi and Prünster (2009). When interest does not focus on probability measures, random measures, for example Lévy processes, are often used as a prior for the mixing distribution. These priors are also employed for nonparametric regression, see among others, Clyde and Wolpert (2007) or Bornkamp and Ickstadt (2009). However, graphical model structures are hardly used up to now in the context of nonparametric mixture modelling with the exception of the recent manuscript by Rodriguez, Lenkoski and Dobra (2010), which focusses on undirected graph models.

However, these models could be useful for applications in which graphs or more generally network inference is of interest, like *e.g.* systems biology. Since causal dependencies are of main importance to biologists, Bayesian networks are preferred over Gaussian graphical models in this field. We suggest to model such systems using nonparametric Bayesian networks and the main goal of our analysis is to find modules, i.e. a subset of components strongly connected within itself but only loosely connected to the rest of a system. Modules might refer to specific functions of the system, whereas the connectivity between them is important to understand higher order functions of the system.

Bayesian networks were developed and are applied mainly by researchers in artificial intelligence and machine learning, while certainly Bayesians should also be interested in this type of model. On the other hand Bayesian nonparametrics might have an important contribution to make in the field of network inference. One goal of this paper is to bring closer together the research communities of Bayesian networks and nonparametric Bayesian statistics.

We begin our paper in Section 2.1 with a wrap of the Bayesian network literature both on directed acyclic graphs and the Gaussian Bayesian network. Section 2.2 then discusses Bayesian nonparametric mixture models based on random probability measures and Section 3 then extends the Gaussian Bayesian network model by using a nonparametric mixture model. In Section 4 we use data simulated from a small biochemical system to test our nonparametric Bayesian network methodology. We further investigate the suitability of our approach for a realistic biological system, the widely studied MAPK (mitogen-activated protein kinase) cascade in section 5 (Kholodenko (2000)). This consists of eight species suggested to be organized in three modules, that we want to confirm in our analysis.

2. METHODS

2.1. *Bayesian Networks*

This section briefly introduces the necessary graph theory and notations; for details or additional material see Jordan (1999), Koller and Friedmann (2009) and Koski and Noble (2009). A graph $\mathcal{G} = (V, E)$ consists of a finite set of nodes V corresponding to random variables $x_1, ..., x_d$, i.e. $V = \{x_1, ..., x_d\}$, and an edge set $E \subset V \times V$. If $\alpha, \beta \in V$ are two distinct nodes, the ordered pair $(\alpha, \beta) \in E$ denotes a directed edge from α to β and D the set of all directed edges. $\langle \alpha, \beta \rangle \in E$ is an undirected edge and U the corresponding set of undirected edges with $E = D \cup U$. If all edges of $\mathcal{G}$ are directed (undirected) then $\mathcal{G}$ is said to be directed (undirected). The undirected version of $\mathcal{G}$ is obtained by replacing all directed edges of $\mathcal{G}$ by undirected ones and is called skeleton. Moreover, for any node $\alpha \in V$ of a given graph $\mathcal{G}$ the set $pa_{\mathcal{G}}(\alpha) = \{\beta \in V | (\beta, \alpha) \in D\}$ defines the set of parents.

Definition 1
A graph $\mathcal{G} = (V, E)$ is called a directed acyclic graph (DAG) if each edge is directed and for any node $\alpha \in V$ there are no cycles, i.e. there does not exist any set of distinct nodes $\tau_1, ..., \tau_m$ such that $\alpha \neq \tau_j$, $j = 1, ..., m$ and $(\alpha, \tau_1, ..., \tau_m, \alpha)$ forms a directed path.

In general, we can represent the joint distribution of the $x_1, ..., x_d$ by

$$p(x_1, ..., x_d) = p(x_1) \cdot p(x_2|x_1) \cdot ... \cdot p(x_d|x_1, ..., x_{d-1}).$$

For any ordering σ of $(1, ..., d)$ we can replace this expression by

$$p(x_1, ..., x_d) = p(x_{\sigma(1)}) \cdot p(x_{\sigma(2)}|x_{\sigma(1)}) \cdot ... \cdot p(x_{\sigma(d)}|x_{\sigma(1)}, ..., x_{\sigma(d-1)});$$

this representation is called factorization.
For a DAG the factorization can be simplified in the following way. A probability distribution p over $x_1, ..., x_d$ factorizes according to a DAG $\mathcal{G}$ if there exists an ordering with $pa_{\mathcal{G}}(x_{\sigma(1)}) = \emptyset$, i.e. $x_{\sigma(1)}$ has no parents, $pa_{\mathcal{G}}(x_{\sigma(j)}) \subseteq \{x_{\sigma(1)}, ..., x_{\sigma(j-1)}\}$ and

$$p(x_1, ..., x_d) = \prod_{j=1}^{d} p(x_{\sigma(j)}|pa_{\mathcal{G}}(x_{\sigma(j)})).$$

The individual $p(x_{\sigma(j)}|pa_{\mathcal{G}}(x_{\sigma(j)}))$ are called conditional probability distributions (CPDs).

Definition 2
A Bayesian network (BN) is a pair $(\mathcal{G}, p)$ where p factorizes according to $\mathcal{G}$ and p is specified as a set of CPDs associated with the nodes of $\mathcal{G}$. The factorization is minimal in the sense that for an ordering of $x_1, ..., x_d$ the parent set $pa_{\mathcal{G}}(x_{\sigma(j)})$ is the smallest set of variables such that $x_{\sigma(j)} \perp pa^c(x_{\sigma(j)})|pa_{\mathcal{G}}(x_{\sigma(j)})$ where "$\perp$" denotes conditional independence.

To simplify notation we assume in the following that the DAGs (and Bayesian networks) are ordered.

For a given set of variables $V = \{x_1, ..., x_d\}$ different DAGs may exist that represent the same independence structure. Two such DAGs are called Markov equivalent. Necessary and sufficient features of a DAG that determine its Markov structure are its skeleton and its immoralities (or v-structures), where an immorality in a graph $\mathcal{G}$ with $E = D \cup U$ is defined as a triple of nodes (α, β, γ) such that $(\alpha, \beta) \in D$ and $(\gamma, \beta) \in D$, but $(\alpha, \gamma) \notin D$, $(\gamma, \alpha) \notin D$ and $\langle \alpha, \gamma \rangle \notin U$.

Theorem 1
Two DAGs are Markov equivalent if and only if they have the same skeleton and the same immoralities. For a proof see Verma and Pearl (1992).

When a Bayesian network is inferred from data, all Markov equivalent DAGs should fit the data equally well as they imply the same conditional independence statements. If additional causal (directional) information exists, only those DAGs from the equivalence class that reflect the causal dependencies should be chosen.

When inferring a Bayesian network from data, it is convenient to assume a parametric model for the CPDs. In the Bayesian networks literature there are two dominant approaches: The first, based on the multinomial distribution with Dirichlet prior, has the advantage that only few assumptions about the form of the dependence structure are made (Koller and Friedmann, 2009), however one disadvantage is that continuous variables can only be handled by discretization (this model is typically called BDe in the Bayesian network literature). The second approach, which we will describe in more detail, is based on the multivariate Gaussian distribution with a normal Wishart prior (typically abbreviated BGe). This approach is relatively restrictive, as it makes a strong parametric assumption. We will, however, present a generalization based on nonparametric mixture models later.

We start with a model for the CPDs $p(x_j|pa_{\mathcal{G}}(x_j))$ for a given $\mathcal{G}$ and generalize this to inference about the DAG $\mathcal{G}$ itself later.

Definition 3
A Bayesian network $(\mathcal{G}, p)$ is called a Gaussian Bayesian network, when the conditional distributions $p(x_j|pa_\mathcal{G}(x_j))$ are given by normal distributions of the form: $x_j|pa_\mathcal{G}(x_j) \sim N(\mu_j + \sum_{\mathcal{K}_j} \beta_{j,k}(x_k - \mu_k), \sigma_j^2)$, where $\mathcal{K}_j = \{k|x_k \in pa_\mathcal{G}(x_j)\}$, the μ_j are the unconditional means of x_j and $\beta_{j,k}$ are real coefficients determining the influence of x_k on x_j.

In a Gaussian Bayesian network, the variable x_j is hence modelled as a linear function of its parents plus normally distributed random noise. Due to the properties of the normal distribution the joint distribution, specified by the CPDs is multivariate Gaussian: Shachter and Kenley (1989) describe an algorithm that extracts the underlying multivariate normal distribution with mean $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)$ and precision matrix $\boldsymbol{M}$ from the specification of the CPDs. Hence the parameters $\boldsymbol{\mu}$, $\boldsymbol{\sigma} = (\sigma_1^2, \ldots, \sigma_d^2)'$ and $\boldsymbol{B} = (\boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_d)$ with $\boldsymbol{\beta}_k = (\beta_{j,1}, \ldots, \beta_{j,k})$, $j < k$ coding the conditional independencies, are an alternative parametrization of the multivariate Gaussian distribution.
Of main interest in inferring a Bayesian network from data is the underlying DAG structure rather than the posterior distributions of $\boldsymbol{\mu}, \boldsymbol{\sigma}$ and $\boldsymbol{B}$. For computational reasons it is hence desirable to integrate out these parameters analytically. One typically chooses the conjugate prior for the multivariate normal distribution, the normal Wishart distribution given by $p(\boldsymbol{\mu}|\boldsymbol{M})p(\boldsymbol{M})$, where $p(\boldsymbol{\mu}|\boldsymbol{M})$ is a multivariate normal distribution and $p(\boldsymbol{M})$ is the Wishart distribution. The distribution $p(\boldsymbol{M})$ can also be transformed to the parametrization in terms of $\boldsymbol{\sigma}$ and $\boldsymbol{B}$, $p(\boldsymbol{\sigma}, \boldsymbol{B})$. A convenient feature of the Wishart distribution is that it factorizes in the same way as the distribution for $x_1, \ldots, x_d$ under a given DAG $\mathcal{G}$, *i.e.* $p(\boldsymbol{\sigma}, \boldsymbol{B}) = \prod_{j=1}^d p(\sigma_j^2, \beta_j)$ (this property is called parameter independence in the Bayesian networks literature, see Geiger and Heckerman (1994)).

With $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})'$, the likelihood for an iid sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ of a multivariate Gaussian distribution with underlying DAG $\mathcal{G}$ is hence given by

$$L(\boldsymbol{\mu}, \boldsymbol{M}_\mathcal{G}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \prod_{i=1}^n p(\boldsymbol{x}_i|\boldsymbol{\mu}, \boldsymbol{M}_\mathcal{G}),$$

where $\boldsymbol{M}_\mathcal{G}$ is chosen so that the conditional independence statements under $\mathcal{G}$ hold. The prior distribution is given by $p(\boldsymbol{\mu}|\boldsymbol{M})p(\boldsymbol{M})$. Now one can first perform the integration with respect to $\boldsymbol{\mu}$, $\int L(\boldsymbol{\mu}, \boldsymbol{M}_\mathcal{G}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)p(\boldsymbol{\mu}|\boldsymbol{M}_\mathcal{G})p(\boldsymbol{M}_\mathcal{G})d\boldsymbol{\mu}$, resulting in the integrated likelihood $L(\boldsymbol{M}_\mathcal{G}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. Now let $\boldsymbol{X}$ be the matrix with rows $\boldsymbol{x}_1', \ldots, \boldsymbol{x}_n'$, and $\boldsymbol{X}^{(\mathcal{I})}$ denote the columns of $\boldsymbol{X}$ with indices in $\mathcal{I}$.

Geiger and Heckerman (1994) (Theorem 2) show that $L(\boldsymbol{M}_\mathcal{G}|\boldsymbol{X})$ factorizes according to the DAG $\mathcal{G}$, when switching to the alternative parameterization so that $L(\boldsymbol{M}_\mathcal{G}|\boldsymbol{X}) = L(\boldsymbol{\sigma}, \boldsymbol{B}|\boldsymbol{X}) = \prod_{j=1}^d L(\sigma_j^2, \boldsymbol{\beta}_j|\boldsymbol{X}^{(j \cup \mathcal{K}_j)})$. In addition the same factorization holds for the Wishart prior distribution, so that the marginal (or integrated) likelihood for $\mathcal{G}$ can be calculated as

$$\begin{aligned} L(\mathcal{G}|\boldsymbol{X}) &= \int L(\boldsymbol{\sigma}, \boldsymbol{B}|\boldsymbol{X})p(\boldsymbol{\sigma}, \boldsymbol{B})d\boldsymbol{\sigma}d\boldsymbol{B} \\ &= \prod_{j=1}^d \int L(\sigma_j^2, \boldsymbol{\beta}_j|\boldsymbol{X}^{(j \cup \mathcal{K}_j)})p(\sigma_j^2, \boldsymbol{\beta}_j)d\sigma_j d\boldsymbol{\beta}_j. \qquad (1) \end{aligned}$$

After performing each of the d integrations in (1) each factor is thus the likelihood of the $j-$th variable given its parents, which we will write as $\rho(\boldsymbol{X}^{(j)}|\boldsymbol{X}^{(\mathcal{K}_j)})$ so that $L(\mathcal{G}|\boldsymbol{X}) =: \prod_{j=1}^{d} \rho(\boldsymbol{X}^{(j)}|\boldsymbol{X}^{(\mathcal{K}_j)})$. By the product rule this is equal to $\prod_{j=1}^{d} \frac{\rho(\boldsymbol{X}^{(j\cup\mathcal{K}_j)})}{\rho(\boldsymbol{X}^{(\mathcal{K}_j)})}$ and the numerator and denominator of each of these terms can be calculated explicitly as the involved integrals are over multivariate t distribution kernels. In addition Geiger and Heckerman (1994) (Theorem 3) show that Markov equivalent graphs receive the same integrated likelihood $L(\mathcal{G}|\boldsymbol{X})$, so that a major requirement from graph theory is met.

Combining expression $L(\mathcal{G}|\boldsymbol{X})$ with a prior distribution $p(\mathcal{G})$ on DAG space then determines the posterior probability $p(\mathcal{G}|\boldsymbol{X})$ for the DAG up to proportionality, *i.e.*

$$p(\mathcal{G}|\boldsymbol{X}) \propto L(\mathcal{G}|\boldsymbol{X})p(\mathcal{G}). \tag{2}$$

In the absence of prior information, the prior distribution for the DAG is often chosen as a uniform distribution, although alternative prior distributions are possible. Friedman and Koller (2003), for example, describe a prior that is uniform over the cardinalities of parent sets, so that complex DAGs are penalized; Mukherjee and Speed (2008) describe an approach for informative prior selection. Inference on the DAG $\mathcal{G}$, that determines the conditional independence statements can in theory be performed analytically as the normalization constant can be obtained by summing up $L(\mathcal{G}|\boldsymbol{X})p(\mathcal{G})$ for all possible DAGs. As the space of DAGs increases exponentially with the number of variables d, analytic inference is, however, practically infeasible. A way out of this situation is to run a Markov chain Monte Carlo algorithm in DAG space based on the posterior given above, see *e.g.* Madigan and York (1995) or Grzegorczyk and Husmeier (2008) for details.

Gaussian Bayesian networks hence have the advantage of being computationally tractable as the involved integrations can be performed analytically. However, a Gaussian Bayesian network also involves two crucial assumptions: (i) the CPDs are all normal distributions, and (ii) the relationships between the variables are given by linear functions. In the following section we present nonparametric mixture models as a generic tool to extend general parametric models to obtain more flexible models, while still being able to exploit some of the analytic tractability of parametric models.

2.2. *Nonparametric Mixture Models*

Suppose the data model is $p(\boldsymbol{x}|\boldsymbol{\theta})$, where $p(\boldsymbol{x}|\boldsymbol{\theta})$ is a probability density, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is an unknown parameter and $\boldsymbol{\Theta}$ a general space. In some cases the modelling situation suggests that there is heterogeneity in the data with respect to $\boldsymbol{\theta}$, so that one value for $\boldsymbol{\theta}$ is not adequate for the full data set, but there are groups in the data for which different values of $\boldsymbol{\theta}$ are adequate.

This leads to the idea of (discrete) mixture models that model the data as

$$\sum w_h p(\boldsymbol{x}|\boldsymbol{\theta}_h), \tag{3}$$

where $\boldsymbol{\theta}_h \in \boldsymbol{\Theta}$, $w_h \geq 0$ and $\sum w_h = 1$. The probability distributions generated by (3) allow for multiple $\boldsymbol{\theta}_h$ and are considerably more flexible than just one $p(\boldsymbol{x}|\boldsymbol{\theta}_h)$ alone.

For what follows, it is useful to note that the parameters w_h and $\boldsymbol{\theta}_h$ in (3) describe a discrete probability distribution P, so that the mixture model can be written as $\int p(\boldsymbol{x}|\boldsymbol{\theta})dP(\boldsymbol{\theta})$. Statistical inference hence focuses on the discrete probability measure P. If the prior for P is chosen with support over an infinite dimensional space (for example, the space of continuous probability densities on $\mathbb{R}$) the name nonparametric mixture model is justified. This situation appears when the mixture model is flexible enough to approximate any probability density on the underlying space, see Ghosh and Ramamoorthi (2003) or Wu and Ghosal (2008) for details regarding the support of nonparametric mixture priors.

In the last decades a variety of distributions, called discrete random probability measures have been developed, which can be used as nonparametric priors for P. A unifying class is given by Ongaro and Cattaneo (2004), which we will describe from two different viewpoints. We will start with a definition.

Definition 4
A random probability measure $\mathbb{P}$ belongs to the Ongaro-Cattaneo class when its realizations can be represented as

$$P(\boldsymbol{\theta}) = \sum_{h=1}^{N} w_h \delta_{\boldsymbol{\theta}_h}(\boldsymbol{\theta}), \tag{4}$$

where $\boldsymbol{\theta}_h, w_h$ and N are random variables specified as follows: The $\boldsymbol{\theta}_h$ are independent and identically distributed realizations of a non-atomic distribution P_0 on $\boldsymbol{\Theta}$ (i.e. $P_0(\{\boldsymbol{\theta}\}) = 0, \forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$) and are independent from w_h, $h = 1, \ldots, N$ and N. The weights $w_1, \ldots, w_N$ conditional on N have a distribution Q_N on the $N-1$ dimensional probability simplex $\{(w_1, w_2, \ldots, w_N)' \in \mathbb{R}_+^N : \sum_{h=1}^N w_h = 1\}$ and N is a random variable with support $\{\mathbb{N}_+ \cup \infty\}$. When $N = \infty$ the weights have a distribution on $\{(w_1, w_2, \ldots) : w_h \in \mathbb{R}_+, \sum w_h = 1\}$.

Several random probability measures in the literature can be identified as special cases of this framework. Stick-breaking priors, described in the work by Ishwaran and James (2001) can be obtained by having $N = \infty$ or $N = N_{max}$ and weights $w_h = v_h \prod_{l<h}(1-v_l)$ with $v_h \overset{iid}{\sim} Beta(a_h, b_h)$. To ensure $\sum_h w_h = 1$, one imposes $v_{N_{max}} = 1$ (when $N = N_{max}$) or $\sum_{h=1}^{\infty} \log(1 + a_h/b_h) = \infty$ (when $N = \infty$) (Ishwaran and James, 2001). The stick-breaking class covers, for example, the Dirichlet process (with $a_h = 1$ and $b_h = M$, where M is the mass parameter of the Dirichlet process) and the Poisson-Dirichlet (or Pitman-Yor) process (with $a_h = 1 - a$ and $b_h = b + ha$ with $a \in [0, 1)$ and $b \geq -a$). Another famous subclass of models are finite mixture models (Frühwirth-Schnatter, 2006). Here one typically fixes N or uses a prior distribution on $\mathbb{N}_+$ for N that has positive support on all integers and the prior for the weights w_h is typically chosen as a symmetric Dirichlet distribution. The general class of James, Lijoi and Prünster (2009) obtained by normalizing random measures with independent increments, is a special case of the above class, when the corresponding intensity of the random measure is homogeneous (*i.e.* the w_h are independent of the $\boldsymbol{\theta}_h$).

From a practical viewpoint it is difficult to decide, which of the prior models in Definition 4 is suitable for the particular modelling situation at hand. A first step would be to calculate the prior mean of $\mathbb{P}$, and adjust the parameters in the prior distribution so that a particular prior mean is achieved with a suitable variability around this mean. The prior mean for the probability of an event A is $E(P(A)) = P_0(A)$ and the covariance of the probability between two events A_1 and A_2 is given by $Cov(P(A_1), P(A_2)) = k_0(P_0(A_1 \cap A_2) - P_0(A_1)P_0(A_2))$, where

$k_0 = E(\sum w_h^2)$ is the expected value of the squared weights (Ongaro and Cattaneo, 2004). The distribution P_0 hence determines prior mean and prior correlation of the random probability measure, while the prior distribution for the w_h mainly determines its variability. When focusing only on the first two moments of the random probability measure, the prior for the weights hence only enters into the calculation of the covariance (via k_0). However, the prior for the weights also contains information about how total probability is distributed to the different atoms and thus makes important assumptions about the clustering structure. The following second viewpoint on random probability measures of form (4) makes these clustering assumptions underlying a random probability measure more apparent.

Suppose you observe an exchangeable sequence $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots$ and this evolves according to the rule

$$\boldsymbol{\theta}_1 \sim P_0, \quad \boldsymbol{\theta}_{n+1} | \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n \sim \sum_{h=1}^{k} p_h(\boldsymbol{n}) \delta_{\tilde{\boldsymbol{\theta}}_h} + p_{k+1}(\boldsymbol{n}) P_0, \tag{5}$$

where $\tilde{\boldsymbol{\theta}}_1, \tilde{\boldsymbol{\theta}}_2, \ldots, \tilde{\boldsymbol{\theta}}_k$ are the $k = k(\boldsymbol{n})$ unique values in the sequence $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_n$ and $\boldsymbol{n} = (n_1, n_2, \ldots, n_k)$ are the number of allocations to the unique values in the sequence. The $p_h(\boldsymbol{n})$ are the probabilities (conditional on $\boldsymbol{n}$) of allocating $\boldsymbol{\theta}_{n+1}$ to $\tilde{\boldsymbol{\theta}}_h$, $h = 1, \ldots, k$, or to a new value simulated from P_0 (for $h = k+1$).

The conditional probabilities $p_h(.)$ are called predictive probability function (PPF). The probability distribution $p(.)$ of $\boldsymbol{n}$, from which the PPF can be calculated, is called the exchangeable probability function (EPPF), and is defined on $\mathbb{N}^* = \bigcup_{k=1}^{\infty} \mathbb{N}^k$, where $\mathbb{N}^k$ is the k-fold Cartesian product of $\mathbb{N}$. Due to exchangeability $p(.)$ needs to be symmetric in its arguments and additionally needs to fulfill $p(1) = 1$ and $p(\boldsymbol{n}) = \sum_{h=1}^{k+1} p(\boldsymbol{n}^{(h+)})$, where $\boldsymbol{n}^{(h+)} = (n_1, \ldots, n_h + 1, \ldots, n_k)$ and $\boldsymbol{n}^{((k+1)+)} = (n_1, \ldots, n_k, 1)$. This ensures a sum of 1 for a given total sample size $\sum_{h=1}^{k} n_h$. The PPF can be recovered from the EPPF via $p_h(\boldsymbol{n}) = p(\boldsymbol{n}^{(h+)})/p(\boldsymbol{n})$.

(Pitman, 1996, Section 3) called exchangeable sequences generated according to (5) a species sampling sequence (due to the analogy of collecting species, for example, in ecology or population genetics). He showed that a sequence is a species sampling sequence if and only if it is a sample from a random distribution of form

$$\sum_h w_h \delta_{\boldsymbol{\theta}_h}(\boldsymbol{\theta}) + (1 - \sum_h w_h) dP_0(\boldsymbol{\theta}),$$

where $\sum_h w_h \leq 1$, $w_h \geq 0$, the $\boldsymbol{\theta}_h$ are iid from a non-atomic P_0 and the w_h are distributed independently of the $\boldsymbol{\theta}_h$. When $\sum_h w_h = 1$, which is the case we are interested in, Pitman (1996) called the sequence *proper* species sampling sequence, which thus coincides with the Ongaro-Cattaneo class from Definition 4. In fact (5) can be seen as a generalization of the Polya urn (or Blackwell-MacQueen) scheme, underlying the Dirichlet process. Species sampling models hence provide an equivalent but very different viewpoint on discrete random probability measures (see Ishwaran and James (2003) for more on the species sampling viewpoint on nonparametric mixture models).

Of particular use is the PPF, as it intuitively describes how the random probability measure allocates its probability mass. For example the Dirichlet process with mass parameter M has the PPF $p_h(\boldsymbol{n}) = \frac{n_h}{\sum_{h=1}^{k} n_h + M}$ for $h = 1, \ldots, k$ and

$p_{k+1}(\boldsymbol{n}) = \frac{M}{\sum_{h=1}^{k} n_h + M}$ leading to the Polya urn scheme. This shows that the probability of attaching $\boldsymbol{\theta}_{n+1}$ to a particular value $\tilde{\boldsymbol{\theta}}_h$ grows linearly with n_h, and thus often results in a relatively small number of large clusters and a large number of small clusters. This is undesirable in some situations, see Lee et al. (2008) for a detailed discussion of this topic. Lee et al. (2008) also propose a Monte Carlo technique to derive the PPF from the information given in Definition 4, which potentially result in PPFs, where the increase is slower than linear. An alternative way of calculating the PPF from a random probability measure is via the EPPF. (Pitman, 2002, p. 44) derives the EPPF for a proper species sampling sequence

$$p(\boldsymbol{n}) = \sum_{(j_1,\ldots,j_k)} E\left(\prod_{h=1}^{k} w_{j_h}^{n_h}\right), \tag{6}$$

where $(j_1, \ldots, j_k)$ ranges over all ordered k-tuples of distinct positive integers, and the expectation is with respect to the distribution of the weights. An alternative representation, from which one can also obtain the PPF and which is better suited for Monte Carlo computation is given by

$$p(\boldsymbol{n}) = E\left[\prod_{h=1}^{k} w_h^{n_h - 1} \prod_{h=1}^{k-1}\left(1 - \sum_{j=1}^{h} w_j\right)\right],$$

see (Pitman, 2002, Theorem 3.1).

PPF and EPPF hence more clearly display the assumptions about the clustering behaviour imposed by the random probability measure. This can be used for setting up the prior distribution for the weights. When one focus of the analysis is to infer a complex clustering structure from the data, as in graph-based problems, one would typically use a model with a flexible EPPF, in which more parameters can be adjusted to the data, while simpler structures (such as the Dirichlet process, where only one parameter determines the clustering structure) may be adequate in other situations.

3. NONPARAMETRIC BAYESIAN NETWORK MODELS

In this section we will combine ideas from graphical and general nonparametric mixture modelling to extend the Gaussian Bayesian network model described in 2.1. For undirected graph modelling a similar approach has been taken recently in the preprint Rodriguez, Lenkoski and Dobra (2010).

From the mixture modelling perspective it is important to decide for which aspects of the graphical model we would like to allow for heterogeneity modelled through a nonparametric mixture model. The Gaussian Bayesian network described in Section 2.1 depends on the unknown parameters $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ and $\boldsymbol{B}$ of the multivariate normal distribution as well as on the DAG $\mathcal{G}$, so that the parameter $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{B}, \mathcal{G})$ in the notation of the last section. When taking the mixture with respect to all components of $\boldsymbol{\theta}$, the base measure P_0 described in the last section is built on the product space for $(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{B}, \mathcal{G})$, and the model for the data is hence $p(\boldsymbol{x}) = \int p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{B}, \mathcal{G})dP(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{B}, \mathcal{G})$ with $P \sim \mathbb{P}$, where P is a discrete mixing measure, $\mathbb{P}$ a random probability measure and $p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{B}, \mathcal{G})$ a multivariate normal distribution that fulfills the conditional independence statements made by

$\mathcal{G}$. As P is a discrete probability measure with support points $\boldsymbol{\mu}_h, \boldsymbol{\sigma}_h, \boldsymbol{B}_h, \mathcal{G}_h$ and probabilities w_h, this can be rewritten as

$$p(\boldsymbol{x}) = \sum w_h p(\boldsymbol{x}|\boldsymbol{\mu}_h, \boldsymbol{\sigma}_h, \boldsymbol{B}_h, \mathcal{G}_h), \tag{7}$$

where the prior distribution for the mixing weights w_h is determined by $\mathbb{P}$ and the prior for $\boldsymbol{\mu}_h, \boldsymbol{\sigma}_h, \boldsymbol{B}_h, \mathcal{G}_h$ is, given by the base measure P_0 of $\mathbb{P}$, for all h. The data are hence modelled to come from a number of different Gaussian Bayesian networks, rather than just one. This overcomes two of the limitations of Gaussian Bayesian network models: (i) We no longer make a normality assumption for the underlying data, but assume a mixture of multivariate normal distributions for the density. It is well known that mixtures of multivariate Gaussians can approximate any density on $\mathbb{R}^d$, provided the number of components can get arbitrarily large (see *e.g.* Wu and Ghosal (2008)). (ii) We no longer assume that the variables x_j are in linear relationships, which is the assumption underlying multivariate normality (see Definition 3). Instead a mixture of multivariate normals leads to a mixture of linear relationships, which is considerably more general.

By assuming a mixture model we split the population into a number of clusters, where each cluster has a weight w_h and a DAG $\mathcal{G}_h$ with network parameters $\boldsymbol{\mu}_h, \boldsymbol{\sigma}_h, \boldsymbol{B}_h$. All clusters share the same prior distribution P_0 for these parameters. When the clusters are assumed to be similar in some aspects, one can also assume hyperprior distributions for hyper-parameters in P_0, so that a shrinkage between clusters can be exploited. An even stronger restriction would be to exclude part of the parameters from the mixture, when the population is not heterogeneous with respect to these parameters. In what follows we will constrain our focus on mixture modelling with respect to $\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{B}$, while one DAG $\mathcal{G}$ will be assumed for the whole population, so that we model

$$p(\boldsymbol{x}|\mathcal{G}) = \int p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{B}, \mathcal{G}) dP(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{B}) \text{ with } P \sim \mathbb{P}. \tag{8}$$

It would not pose a serious problem to also include the graph into the mixture: Computations would get slightly more involved, and the implementation would be different from the one described below. However, in the application we consider in this paper it is of interest to learn one DAG with different network parameters in different components for the whole population of observations, rather than completely different DAGs in the subgroups.

In addition, main interest is in the DAG structure and the clustering structure of the population rather than the network parameters $\boldsymbol{\mu}, \boldsymbol{\sigma}$ and $\boldsymbol{B}$. Hence as suggested in Section 2.1, we integrate out these parameters from the likelihood. A way of writing the integrated likelihood for a mixture model is by introducing latent indicator variables $\boldsymbol{l} = (l_1, \ldots, l_n)'$ for each observation $\boldsymbol{x}_i$, with values $l_i \in \{1, 2, 3, \ldots, k\}$ corresponding to the k mixture components and probabilities $w_1, w_2, w_3, \ldots, w_k$. So that for a data set $\boldsymbol{X}$ we obtain the integrated likelihood

$$L(\boldsymbol{w}, \boldsymbol{l}, \mathcal{G}|\boldsymbol{X}) = \prod_h L(\mathcal{G}|\boldsymbol{X}_{(\mathcal{I}_h)}) \prod_h w_h^{n_h}, \tag{9}$$

where $L(\mathcal{G}|\boldsymbol{X})$ is as defined in (1), $\mathcal{I}_h = \{i \in \{1, \ldots, n\} | l_i = h\}$ and $\boldsymbol{X}_{(\mathcal{I}_h)}$ is the matrix consisting of the subset of rows of $\boldsymbol{X}$ corresponding to $\mathcal{I}_h$. Here n_h

denotes the cardinality of $\mathcal{I}_h$. Now integrating $\prod_h w_h^{n_h}$ with respect to the prior distribution for $\boldsymbol{w}$ implicit in $\mathbb{P}$ one obtains a function depending only on the prior distribution and $\boldsymbol{n} = (n_1, \dots, n_k)$. From the discussion in Section 2.2 it follows that this is proportional to the EPPF associated with the random measure $\mathbb{P}$. A table of EPPFs for different choices of the random probability measure $\mathbb{P}$ is given for example in Lau and Green (2007). Hence we obtain a once more integrated likelihood $\prod_h L(\mathcal{G}|\boldsymbol{X}_{(\mathcal{I}_h)})p(\boldsymbol{n})$, where $p(\boldsymbol{n})$ is the EPPF corresponding to the underlying random measure $\mathbb{P}$.

The computational implementation of the proposed model hence needs to be run only on the space of DAGs $\mathcal{G}$ and the latent allocation vector $\boldsymbol{l}$. The marginal posterior distribution for these quantities is given by

$$p(\boldsymbol{l}, \mathcal{G}|\boldsymbol{X}) = \prod_h L(\mathcal{G}|\boldsymbol{X}_{(\mathcal{I}_h)})p(\boldsymbol{n})p(\mathcal{G}). \tag{10}$$

The MCMC scheme can thus alter between updating the DAG given the allocation and updating the allocation given the DAG. Well developed algorithms exist for updating the DAG, where for the allocation vector one can use algorithms in which the random probability measure is marginalized out. One example of such an algorithm is described by Nobile and Fearnside (2007) (see also Grzegorczyk et al. (2008)), who describe different Gibbs or Metropolis Hastings moves for the allocations. A variety of other samplers primarily run on the space of allocations, see for example Neal (2000) for an earlier reference with focus on the Dirichlet process. When the EPPF contains unknown parameters so that $p(\boldsymbol{n}) = p_{\boldsymbol{\xi}}(\boldsymbol{n})$ one can use an additional prior $p(\boldsymbol{\xi})$ and introduce additional MCMC moves to update $\boldsymbol{\xi}$.

A recent alternative MCMC approach for (rather general) random probability measures is described by Kalli, Griffin and Walker (2010), based on earlier work on the blocked Gibbs sampler by Ishwaran and James (2001). This type of algorithm has become quite popular recently and does not marginalize out parameters but simulates from the corresponding conditionals and is therefore more closely related to the traditional data augmentation algorithm for finite mixture models (Frühwirth-Schnatter, 2006), with an adaption to deal with potentially infinitely many components. In our situation, there is no need to use these algorithms, since component specific parameters are not of main interest. Determining, whether conditional algorithms improve upon marginal algorithms for network models in terms of computational efficiency for general models is a question of future research.

4. SIMULATIONS

In order to evaluate the performance of the nonparametric Bayesian network model (NPBN) from Section 3 we compared it in a simulation study with two alternative models. For this purpose we used the Gaussian Bayesian network (BGe), which does not include a mixture component and a nonparametric mixture model (NPM) without a network structure. Specifically we compare the posterior predictive probability for all models on the test data set and the quality of the estimated graph for the network based BGe and NPBN. We will consider an example from systems biology.

For generating a controllable reference data set corresponding to a realistic biochemical system, we simulated a mixture of four proteins A, B, C and D. In this system, proteins A and B can bind each other, forming the complex AB, and C and

D can bind forming the complex CD

$$A + B \underset{k_{AB}^{off}}{\overset{k_{AB}^{on}}{\rightleftarrows}} AB \quad \text{and} \quad C + D \underset{k_{CD}^{off}}{\overset{k_{CD}^{on}}{\rightleftarrows}} CD.$$

These reversible processes can be described by mass-action kinetics with corresponding association and dissociation rate constants k^{on} and k^{off}. The resulting system of differential equations describing the rate of change in the concentration ([.]) of each component is:

$$\begin{aligned}
\frac{d[A]}{dt} = \frac{d[B]}{dt} &= -k_{AB}^{on}[A][B] + k_{AB}^{off}[AB] \\
\frac{d[AB]}{dt} &= k_{AB}^{on}[AB] - k_{AB}^{off}[A][B] \\
\frac{d[C]}{dt} = \frac{d[D]}{dt} &= -k_{CD}^{on}[C][D] + k_{CD}^{off}[CD] \\
\frac{d[CD]}{dt} &= k_{CD}^{on}[CD] - k_{CD}^{off}[CD]
\end{aligned}$$

from which it can be also observed that the total concentration of each protein (e.g. $[A] + [AB]$ for protein A) is a conserved quantity.

In steady state, the concentrations of all species are constant, implying that the binding and dissociation rates of each interaction are equal:

$$k_{AB}^{on}[A][B] = k_{AB}^{off}[AB] \tag{11a}$$

$$k_{CD}^{on}[C][D] = k_{CD}^{off}[CD]. \tag{11b}$$

In order to reveal the correlations between all species, we independently sampled their total concentrations and calculated the steady state using Equation (11). In our simulation, all quantities are considered dimensionless as only their relation and not their absolute value is revealed. The values for the initial total concentrations were drawn from a normal $N(3.5, 1)$ distribution. Such variability in total protein concentration simulates, for example, the typically observed stochastic cell-to-cell variations in the expression levels of proteins. The values for the rate constants were chosen to be $k_{AB}^{on} = 10$, $k_{AB}^{off} = 1$, $k_{CD}^{on} = 1$, $k_{CD}^{off} = 1$ to simulate binding reactions with different bias towards the bound state. Our final data set consisted of 1000 concentrations of the six species. In systems biology such simulated data generation processes are commonly used, see for example, Kholodenko (2000).

Since sample sizes in experimental data are often limited we consider only samples of 50 and 100 observations. The rest is used for test/validation. Figure 1 shows a representative subsample of the data; the nonlinear, hyperbolic pattern of the relationships is clearly visible, for example, the relationship of A and B. Data simulation was done with Mathematica 7.0 (Research, 2008).

For specifying the NPBN model, we applied the general methodology described in Section 3, by using a random probability measure specified as follows. We used a Poisson distribution with parameter $\lambda = 1$ for the number of components N; conditional on N, a symmetric Dirichlet distribution was used for the weights w_h with an N dimensional parameter vector $(\delta, \dots, \delta)$, where we chose $\delta = 1$. The EPPF of

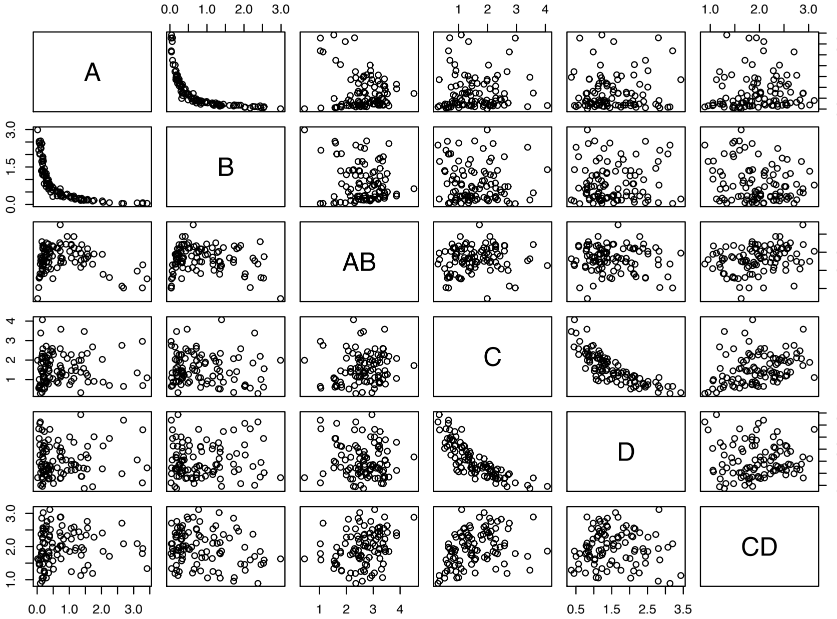

Figure 1: Scatterplots of the generated data, representative subsample of size 100.

such a random probability measure is proportional to $\frac{N!}{(N-k(\boldsymbol{n}))!}\prod_{h=1}^{k(\boldsymbol{n})}\frac{\Gamma(\delta+n_h)}{\Gamma(\delta)}$ (Lau and Green, 2007). Note that the EPPF depends on both the unknown parameter N and δ, so that essentially two parameters control the flexibility of the clustering behavior. While we fixed δ in the simulations, we used a prior distribution for N. For the normal Wishart prior distribution we used the identity matrix for the prior precision matrix and chose the degrees of freedom parameter equal to $d+2$ to ensure propriety of the prior distribution. The mean vector of the multivariate normal distribution was chosen as a vector of zeros. The prior distribution on the space of DAGs was chosen as the prior by Friedman and Koller (2003), which is uniform over the cardinalities of parent sets. The overall posterior distribution for the allocation vector and the target for MCMC simulations is hence given by

$$p(\boldsymbol{l},\mathcal{G},N|\boldsymbol{X}) = \prod_h L(\mathcal{G}|\boldsymbol{X}_{(\mathcal{I}_h)})p_N(\boldsymbol{n})p(N)p(\mathcal{G}), \tag{12}$$

where $p(N)$ is a Poisson distribution with parameter 1.

The BGe algorithm was applied using the same normal Wishart prior distribution, while the NPM algorithm was applied using the same specification for the random probability measure, with the DAG assumed to be fixed and completely connected.

To analyze the data we used the MCMC algorithm outlined in Section 3 and described in more detail in the Appendix. We conducted several runs for the NPBN model and the reference models BGe and NPM, for both sample sizes 50 and 100. We present in detail a run with $4\cdot 10^6$ iterations with thinning of 2000 and a burn in of $1\cdot 10^6$ iterations. We initialized the allocation vector with allocations obtained from the k-means algorithm with 10 components. This has two advantages: (i) The algorithm starts in a region of the posterior distribution with potentially large posterior mass and (ii) using a larger number of components as initialization is beneficial as the merge step of the algorithm is more effective (see Appendix). For both NPBN and NPM the same clusterings were used.

In order to compare the performance of the three different approaches we computed the posterior predictive probability (ppp) for the simulated data which has not been used to train the system. For one data point $\boldsymbol{x}^{test}$ the ppp is calculated by

$$p(\boldsymbol{x}^{test}) = \int \underbrace{p(\boldsymbol{x}^{test}|\boldsymbol{\theta}_m)}_{likelihood} p(\boldsymbol{\theta}_m|\boldsymbol{x}_1^{train},\ldots,\boldsymbol{x}_n^{train})d\boldsymbol{\theta}_m$$

with $m \in \{\text{BGe, NPM, NPBN}\}$. The overall ppp on log scale for all test data equals

$$\log\left(\prod_{i=1}^{n^{test}} p(\boldsymbol{x}_i^{test})\right) = \sum_{i=1}^{n^{test}} \log p(\boldsymbol{x}_i^{test})$$

with higher values corresponding to a better model.

Figure 2 shows the results of the log ppp for the test data. The training data consisted of 100 observations. It can be seen that the NPM and NPBN perform better than the BGe model. This is possibly due to the non-linearity in the relationship between the variables (see also Figure 1). Both the mean of the log ppp in Table 1 and the quantiles visible in Figure 2 are larger for NPBN and NPM. This is also indicated by the probabilities in Table 1, which can be interpreted as the

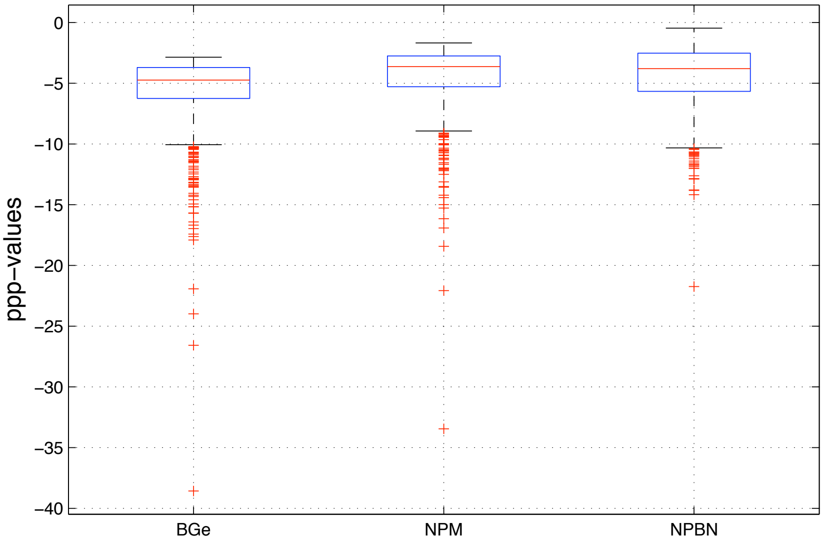

Figure 2: Boxplot of log posterior predictive probabilities for the 900 test data points, based on a training set of size 100.

probability that the test data stem from the corresponding model. The comparison between NPM and NPBN is less clear: There are less surprising observations in the test data set for the NPBN, however the interquartile range for the log ppps is a bit smaller for the NPM. Note however, that the NPBN which infers a sparse network compared to the fully connected one underlying the NPM model, is performing similarly. Moreover the inferred network structure of the NPBN model reflects the true interactions.

Another possibility to compare the two models that infer a network (BGe and NPBN) is to consider the marginal posterior probabilities of the network edges. Figures 3 (i) and 3 (ii) show the resulting posterior probabilities for the network nodes A, B, AB, C, D, CD (see also Equation (11)). The probabilities for a connection are coded in a grey scale, white corresponds to zero and black corresponds to one. In our simulated data example the true underlying graph topology consists of two blocks of fully connected nodes, namely, $\{A, B, AB\}$ and $\{C, D, CD\}$ while there are no edge connections between the two blocks. Note that the interactions of the nodes within each block are implemented according to Equation (11). Since we do not know the true edge directions, we assess the network reconstruction accuracy in terms of undirected edges. The (marginal) edge posterior probabilities of an (undirected) edge connection between two nodes can be estimated by the fraction of graphs in the sample that contain an edge between the two nodes pointing in

Sample Size		BGe	NPM	NPBN
50	mean	-5.5943	-5.0128	-5.0245
	model probability	0.22	0.39	0.39
100	mean	-5.5512	-4.4677	-4.3971
	model probability	0.13	0.41	0.46

Table 1: Predictive probabilities for both samples (50 and 100 observations).

either direction. For our 6-node network example the posterior probabilities of all possible undirected edge connections leads to a symmetric 6 × 6 matrix. Figure 3 shows heatmaps for this matrix for BGe (panel (i)) and NPBN (panel (ii)). It can be seen that the NPBN model, overall, assigns higher posterior probabilities to the edges within the two blocks than the BGe model. For the standard BGe model the node AB is neither connected with node A nor with node B. Moreover, the posterior probability of the edge connection $D - CD$ is only of moderate size (medium grey).The more sophisticated NPBN model assigns the highest posterior probability to four of the six true gold standard edge connections (black elements in Figure 3). Furthermore, the true edge $A - AB$ at least appears in medium grey. Its posterior probabiliy is comparable to the posterior probability of two falses edge connections: $C - AB$ and $D - AB$. Overall, the heatmaps indicate that NPBN gives a better network reconstruction accuracy than the standard BGe model.

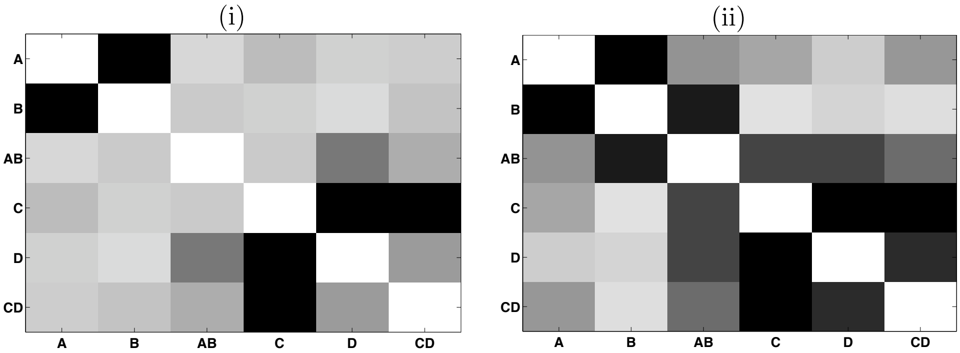

Figure 3: Heatmap inferred from the data set with 50 observations; representations of the (marginal) posterior probabilities of undirected edges, panel (i) BGe and panel (ii) NPBN. In both panels columns and rows represent the nodes A, B, AB, C, D, and CD, and a grey shading is used to indicate the posterior probabilities (black corresponds to 1, and white corresponds to 0).

REFERENCES

Bornkamp, B. and Ickstadt, K. (2009) Bayesian nonparametric estimation of continuous monotone functions with applications to dose-response analysis, *Biometrics* **65**, 198–205.

Carvalho, C. M. and Scott, J. G. (2010) Objective Bayesian model selection in Gaussian graphical models, *Biometrika* **xx**, xx–xx.

Clyde, M. A. and Wolpert, R. L. (2007) Nonparametric function estimation using overcomplete dictionaries, *in* J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith and M. West (eds.), *Bayesian Statistics 8*, Oxford University Press, pp. 91–114.

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

Escobar, M. D. and West, M. (1995) Bayesian density estimation using mixtures, *Journal of the American Statistical Association* **90**, 577–588.

Ferguson, T. S. (1973) A Bayesian analysis of some nonparametric problems, *Annals of Statistics* **1**, 209–230.

Friedman, N. and Koller, D. (2003) Being Bayesian about network structure, *Machine Learning* **50**, 95–126.

Friedman, N., Linial, M., Nachman, I. and Pe'er, D. (2000) Using Bayesian networks to analyze expression data, *Journal of Computational Biology* **7**, 601–620.

Frühwirth-Schnatter, S. (2006) *Finite Mixture and Markov Switching Models*, Springer, Berlin.

Geiger, D. and Heckerman, D. (1994) Learning Gaussian networks, *in* R. L. de Mántaras and D. Poole (eds.), *Uncertainty in Artificial Intelligence Proceedings of the Tenth Conference*, pp. 235–243.

Ghosh, J. K. and Ramamoorthi, R. V. (2003) *Bayesian Nonparametrics*, Springer, New York.

Giudici, P. (1996) Learning in graphical Gaussian models, *in* J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith (eds.), *Bayesian Statistics 5*, Oxford University Press, Oxford, pp. 621–628.

Grzegorczyk, M. and Husmeier, D. (2008) Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move, *Machine Learning* **71**, 265–305.

Grzegorczyk, M., Husmeier, D., Edwards, K., Ghazal, P. and Millar, A. (2008) Modelling non-stationary gene regulatory processes with a non-homogeneous Bayesian network and the allocation sampler, *Bioinformatics* **24**(18), 2071–2078.

Ishwaran, H. and James, L. F. (2001) Gibbs sampling methods for stick-breaking priors, *Journal of the American Statistical Association* **96**, 161–173.

Ishwaran, H. and James, L. F. (2003) Generalized weighted Chinese restaurant processes for species sampling mixture models, *Statistica Sinica* **13**, 1211–1235.

James, L. F., Lijoi, A. and Prünster, I. (2009) Posterior analysis for normalized random measures with independent increments, *Scandinavian Journal of Statistics* **36**, 76–97.

Jordan, M. I. (1999) *Learning in Graphical Models*, MIT Press.

Kalli, M., Griffin, J. E. and Walker, S. G. (2010) Slice sampling mixture models, *Statistics and Computing* **00**, 00–00.

Kholodenko, B. (2000) Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades, *European Journal of Biochemistry* **267**, 1583–1588.

Koller, D. and Friedmann, N. (2009) *Probabilistic Graphical Models - Principles and Techniques*, MIT press.

Koski, T. and Noble, J. M. (2009) *Bayesian Networks - An Introduction*, Wiley.

Lau, J. W. and Green, P. J. (2007) Bayesian model based clustering procedures, *Journal of Computational and Graphical Statistics* **16**, 526–558.

Lee, J., Quintana, F. A., Müller, P. and Trippa, L. (2008) Defining predictive probability functions for species sampling models, Technical report, MD Anderson Cancer Center.

Madigan, D. and York, J. (1995) Bayesian graphical models for discrete data, *International Statistical Review* **63**, 215–232.

Mukherjee, S. and Speed, T. P. (2008) Network inference using informative priors, *Proceedings of the National Academy of Sciences* **105**, 14313–14318.

Neal, R. (2000) Markov chain sampling methods for Dirichlet process mixture models, *Journal of Computational and Graphical Statistics* **9**, 249–265.

Nobile, A. and Fearnside, A. T. (2007) Bayesian finite mixtures with an unknown number of components, *Statistics and Computing* **17**, 147–162.

Ongaro, A. and Cattaneo, C. (2004) Discrete random probability measures: A general framework for nonparametric Bayesian inference, *Statistics and Probability Letters* **67**, 33–45.

Pearl, J. (1985) A model of self-activated memory for evidential reasoning, *in Proceedings of the 7th Conference of the Cognitive Science Society*, University of California, Irvine, CA, pp. 329–334.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Francisco, CA, USA.

Pitman, J. (1996) Some developments of the Blackwell-MacQueen urn scheme, *in* T. S. Ferguson, L. S. Shapley and J. B. MacQueen (eds.), *Statistics, Probability and Game Theory: Papers in Honor of David Blackwell*, Institute of Mathematical Statistics, pp. 245–268.

Pitman, J. (2002) *Combinatorial Stochastic Processes*, Springer.

Research, W. (2008) *Mathematica Edition: Version 7.0*.

Rodriguez, A., Lenkoski, A. and Dobra, A. (2010) Sparse covariance estimation in heterogeneous samples.
URL: *http://arxiv.org/abs/1001.4208*

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

Verma, P. and Pearl, J. (1992) An algorithm for deciding if a set of observed independencies has a causal explanation, *in* D. Dubois, M. Welman, B. D'Ambrosio and P. Smets (eds.), *Uncertainty in Artificial Intelligence Proceedings of the Eighth Conference*, pp. 323–330.

Wu, Y. and Ghosal, S. (2008) Kullback Leibler property of kernel mixture priors in Bayesian density estimation, *Electronic Journal of Statistics* **2**, 298–331.

APPENDIX

Appendix: MCMC-Sampler

Here we describe the MCMC sampler used for analysing the NPBN model proposed in this paper. The BGe and the NPM model are analysed with the same algorithm, by only updating the graph (with all observations allocated to one component) or only updating the allocations (with a completely connected DAG). The Appendix is based on Grzegorczyk et al. (2008) and Nobile and Fearnside (2007), where a more detailed description can be found.

The MCMC sampler generates a sample from the joint posterior distribution of $\boldsymbol{l}, \mathcal{G}, N$ given in Equation (12) and comprises six different types of moves in the state-space $[\boldsymbol{l}, \mathcal{G}, N]$. Before the MCMC simulation is started, probabilities p_i $(i = 1, \dots, 6)$ with $p_1 + \dots + p_6 = 1$ must be predefined with which one of these move types is selected. The moves consist of a structure move, that proposes a change in the graph (abbreviated by DAG move) and five moves that change the allocations (abbreviated by Gibbs, M1, M2 , split and merge). Below we will describe these different move types in some detail.

DAG move

The first move type is a classical structure MCMC single edge operation on the graph $\mathcal{G}$ while the number of components N and the allocation vector $\boldsymbol{l}$ are left unchanged (Madigan and York, 1995). According to the transition probability distribution

$$q(\tilde{\mathcal{G}}|\mathcal{G}) = \begin{cases} \frac{1}{|\mathcal{N}(\mathcal{G})|} & , \ \tilde{\mathcal{G}} \in \mathcal{N}(\mathcal{G}) \\ 0 & , \ \tilde{\mathcal{G}} \notin \mathcal{N}(\mathcal{G}) \end{cases} \tag{13}$$

a new graph $\tilde{\mathcal{G}}$ is proposed, and the new state $[\tilde{\mathcal{G}}, N, \boldsymbol{l}]$ is accepted according to

$$A(\tilde{\mathcal{G}}|\mathcal{G}) \quad = \quad \frac{p(\tilde{\mathcal{G}}|\boldsymbol{X})}{p(\mathcal{G}|\boldsymbol{X})} \cdot \frac{q(\mathcal{G}|\tilde{\mathcal{G}})}{q(\tilde{\mathcal{G}}|\mathcal{G})}$$

where $|\mathcal{N}(\mathcal{G})|$ is the number of neighbors of the DAG $\mathcal{G}$, that can be reached from the current graph by one single edge operation and $p(\mathcal{G}|\boldsymbol{X})$ is defined in (2) for the BGe model and by (12) for the NPBN model.

Allocation moves
The five other move types are adapted from Nobile and Fearnside (2007) and operate on $\boldsymbol{l}$ or on N and $\boldsymbol{l}$. If there are $N > 2$ mixture components, then moves of the type M1 and M2 can be used to re-allocate some observations from one component h to another one $\tilde{h}$. That is, a new allocation vector $\tilde{\boldsymbol{l}}$ is proposed while $\mathcal{G}$ and N are left unchanged. The split and merge moves change N and $\boldsymbol{l}$. A split move proposes to increase the number of mixture components by 1 and simultaneously tries to re-allocate some observations to fill the new component. The merge move is complementary to the split move and decreases the number of mixture components by 1. The acceptance probabilities for M1, M2, split and merge are of the same functional form

$$A(\tilde{\boldsymbol{l}}|\boldsymbol{l}) = \left\{1, \frac{p(\tilde{\boldsymbol{l}}, \mathcal{G}, N|\boldsymbol{X})}{p(\boldsymbol{l}, \mathcal{G}, N|\boldsymbol{X})} \frac{q(\tilde{\boldsymbol{l}}|\boldsymbol{l})}{q(\boldsymbol{l}|\tilde{\boldsymbol{l}})}, \right\}, \tag{14}$$

where the proposal probabilities $q(.|.)$ depend on the move type (M1, M2, split, merge). Finally, the Gibbs move re-allocates only one single observation by sampling its new allocation from the corresponding full conditional distribution (see Nobile and Fearnside (2007)) while leaving N and $\boldsymbol{l}$ unchanged. In the following we give an idea how the allocation moves work, for a detailed description including the corresponding Metropolis-Hastings acceptance probabilities, see Nobile and Fearnside (2007).

Gibbs move on the allocation vector $\boldsymbol{l}$
If there is one component only, symbolically $N = 1$, select another move type. Otherwise randomly select an observation i among the n available and determine to which component h $(1 \leq h \leq N)$ this observation currently belongs. For each mixture component $\tilde{h} = 1, \ldots, N$ replace the i-th entry of the allocation vector $\boldsymbol{l}$ by component $\tilde{h}$ to obtain $\boldsymbol{l}(i \longleftarrow \tilde{h})$. We note that $\boldsymbol{l}(i \longleftarrow h)$ is equal to the current allocation vector $\boldsymbol{l}$. Subsequently, sample the $i-$th entry of the new allocation vector $\tilde{\boldsymbol{l}}$ from the corresponding multinomial full conditional distribution.

The M1 move on the allocation vector $\boldsymbol{l}$
If there is one component only, symbolically $N = 1$, select a different type of move. Otherwise randomly select two mixture components h and $\tilde{h}$ among the N available. Draw a random number p from a Beta distribution with parameters equal to the corresponding hyperparameters of the Dirichlet prior on the mixture weights. Re-allocating each observation currently belonging to the h-th or $\tilde{h}$-th component to component h with probability p or to component $\tilde{h}$ with probability $1 - p$ gives the proposed allocation vector $\tilde{\boldsymbol{l}}$.

The M2 move on the allocation vector $\boldsymbol{l}$
If there is one component only, symbolically $N = 1$, select a different move type. Otherwise randomly select two mixture components h and $\tilde{h}$ among the N available and then randomly select a group of observations allocated to component h and attempt to re-allocate them to component $\tilde{h}$. If the h-th component is empty the move

fails outright. Otherwise draw a random number u from a uniform distribution on $1, \ldots, n_h$ where n_h is the number of observations allocated to the h-th component. Subsequently, randomly select u observations from the n_h in component h and allocate the selected observations to component $\tilde{h}$ to obtain the proposed allocation vector $\tilde{\boldsymbol{l}}$.

The split move
Randomly select a mixture component h $(1 \leq h < N)$ as the ejecting component. Draw p_E from a $Beta(a, a)$ distribution with $a > 0$ and re-allocate each observation currently allocated to component h in the vector $\boldsymbol{l}$ with probability p_E to a new component with label $N + 1$. Subsequently swap the labels of the new mixture component $N + 1$ with a randomly chosen mixture component label $\tilde{h}$ including the label $N + 1$ of the ejected component itself $(1 \leq \tilde{h} \leq N + 1)$ to obtain the proposed allocation vector $\tilde{\boldsymbol{l}}$.

The merge move
Randomly select a mixture component h $(1 \leq h \leq N)$ as the absorbing component and another component $\tilde{h}$ $(1 \leq \tilde{h} \leq N)$ with $\tilde{h} \neq h$ as the disappearing component. Re-allocate all observations currently allocated to the disappearing component $\tilde{h}$ by $\boldsymbol{l}$ to component h to obtain the new allocation vector $\tilde{\boldsymbol{l}}$. Then delete the (empty) component $\tilde{h}$ to obtain the new number of components $N = N - 1$.

A disadvantage of the split move is the fact that allocations are chosen randomly to form the new mixture component. A way to partially overcome this problem is to use informative starting values of the algorithm. One approach with which we have made good experience is to start the sampler based on the result of a k-means clustering with a large number of components. The merge move then rather quickly finds a good allocation of the mixture components.