Learning the Structure of Probabilistic Graphical Models with an Extended Cascading Indian Buffet Process

Patrick Dallaire and **Philippe Giguère** and **Brahim Chaib-draa**
Department of Computer Science and Software Engineering
Université , Québec, Canada G1V 0A6
{patrick.dallaire,philippe.giguere,brahim.chaib-draa}@ift.ulaval.ca

Abstract

In this paper, we present an extension of the cascading Indian buffet process (CIBP) intended to learning arbitrary directed acyclic graph structures as opposed to the CIBP, which is limited to purely layered structures. The extended cascading Indian buffet process (eCIBP) essentially consists in adding an extra sampling step to the CIBP to generate connections between non-consecutive layers. In the context of graphical model structure learning, the proposed approach allows learning structures having an unbounded number of hidden random variables and automatically selecting the model complexity. We evaluated the extended process on multivariate density estimation and structure identification tasks by measuring the structure complexity and predictive performance. The results suggest the extension leads to extracting simpler graphs without scarifying predictive precision.

1 INTRODUCTION

Probabilistic graphical models are elegant representations for multivariate joint probability distribution which use graph structures to describe relations among variables. However, learning the structure of probabilistic graphical models is a difficult task, especially when it involves discovering hidden variables. Over the last two decades, researchers have explored a variety of approaches to this problem, from frequentist (Lauritzen, 1996, Whittaker, 1990) to Bayesian (Friedman and Koller, 2003), but most of them assume finite sets of hidden variables.

Nonparametric Bayesian methods are well known for their great flexibility regarding the unknown dimensionality of generative models. Instead of looking for specific models having finite dimensions, the idea is rather to define probability measures on infinite dimensional spaces and infer a finite subset of active dimensions explaining the data. The Dirichlet process (Ferguson, 1973), surely the most famous of these methods, is a prior stochastic process on random probability measure allowing for infinitely many outcome possibilities. It has notably been applied to clustering tasks where the number of components is not known in advance (MacEachern and Müller, 1998). Another important prior is the Beta process (Hjort, 1990), a distribution on completely random measures that has initially been used in survival analysis and later applied to construct infinite factorial models (Paisley and Carin, 2009).

The Dirichlet and Beta processes are among the most important stochastic processes for the development of other nonparametric Bayesian approaches. In machine learning, these processes have largely been used to produce priors on infinite binary matrices, namely the Chinese restaurant process (CRP) (Pitman, 1995) and the Indian buffet process (IBP) (Griffiths and Ghahramani, 2011). These processes are now often used to construct more complex processes such as hierarchical models (Jordan, 2010, Teh et al., 2006, Thibaux and Jordan, 2007).

Recently, there have been attempts to combine structure learning in graphical models and nonparametric Bayesian methods. Wood et al. (2006) used the IBP to define graphical models with a single infinite layer of hidden variables that are viewed as latent features for the observable data. A more expressive version of this model was proposed by Chen et al. (2011) who used a hierarchy of Beta processes to construct graphical models featuring three infinite hidden layers. Adding even more flexibility to the model, Adams et al. (2010) introduced the cascading Indian buffet process (CIBP) as a prior on infinitely deep networks, significantly enlarging the set of possible structures. However, the probability distribution induced by the CIBP still does not support the set of all possible directed acyclic graph structures due to its specific layered structure.

In this paper, we first describe the cascading Indian buffet process in terms of Beta processes. Secondly, we use the Beta process representation to propose an extension that allows inferring arbitrary directed acyclic graph structures among hidden variables. Thirdly, we demonstrate the learning capability of the vanilla and extended cascading Indian buffet process on density estimation and structure identification tasks, and by the same occasion, report the first quantitative results for the vanilla CIBP. In particular, Section 2 introduces the specific graphical model used for this work. The proposed extension is presented in Section 3 and the inference procedure in Section 4. Experiments comparing both methods are detailed in Section 5 and discussed in Section 6.

2 MODEL DESCRIPTION

Let us first consider a dataset $\{\mathbf{x}_n\}_{n=1}^N$ containing D-dimensional samples drawn from a statistical model of interest. We represent the generative model of the data as a belief network with hidden units and assume conditional independency among observable units given the hidden ones. In our definition, each unit belongs to a specific layer $m \in \{0, \ldots, M\}$ and we use notation $u_k^{(m)}$ to refer to the k^{th} unit in layer m. In this structure, units are restricted to having parents only in strictly deeper layers, thus all observable units can be associated to layer $m = 0$. We denote by $K^{(m)}$ the number of connected (or *active*) units in layer m. Moreover, to indicate whether unit $u_k^{(m)}$ is a parent of unit $u_i^{(s)}$, we define the binary variable $Z_{i,k}^{(s,m)}$ which indicates the existence of a directed edge between the two units.

Although many conditional probability distributions could be used to describe unit behaviours, in the following we adopt the *nonlinear Gaussian belief network* model specification of Adams et al. (2010) for the purpose of comparison. In that model, the conditional probability of $u_i^{(s)}$ depends on a weighted sum of its parents, where $W_{i,k}^{(s,m)}$ represents the weight of parent unit $u_k^{(m)}$. This sum is also biased and is expressed as:

$$y_i^{(s)} = \gamma_i^{(s)} + \sum_{m=s+1}^{M} \sum_{k=1}^{K^{(m)}} Z_{i,k}^{(s,m)} W_{i,k}^{(s,m)} u_k^{(m)} \tag{1}$$

where the parameter $\gamma_i^{(s)}$ is the actual bias. Afterward, the sum $y_i^{(s)}$ is corrupted by zero mean Gaussian noise of precision $\nu_i^{(s)}$ and then passed through sigmoid function $\sigma(x) = 2/(1 + e^{-x}) - 1$, yielding the output value of unit $u_i^{(s)}$. Ultimately, the conditional probability distribution of $u_i^{(s)}$ is:

$$p(u_i^{(s)} | y_i^{(s)}, \nu_i^{(s)}) = \frac{\exp\left\{-\frac{\nu_i^{(s)}}{2}\left[\sigma^{-1}(u_i^{(s)}) - y_i^{(s)}\right]^2\right\}}{\sigma'(\sigma^{-1}(u_i^{(s)}))\sqrt{2\pi/\nu_i^{(s)}}} \tag{2}$$

where $\sigma'(x) = \frac{d}{dx}\sigma(x)$ and $\sigma^{-1}(x)$ is the inverse sigmoid function. This distribution is particularly interesting since by only varying $\nu_i^{(s)}$, it can model various modes of operation such as *linear* or *nonlinear* as continuous behaviours, and *deterministic* or *binary* as discrete behaviours (Frey, 1997).

3 PRIOR ON NETWORK STRUCTURES

According to the Bayesian learning framework, identifying the parameters and structure of a graphical model requires specifying appropriate prior probability distributions on these unknowns. In what follows, we present a Bayesian prior supporting every possible directed acyclic graph (DAG) structures among hidden units.

3.1 Infinite Dimensional Layer

In Section 2, we introduced a model where units were associated to layers, but did not specify the effective size of the layers. Let us first consider a case involving only two successive layers, a layer $m - 1$ and its parent layer m comprising K hidden (potentially inactive) units. To learn the structure of such graphical models, Wood et al. (2006) adopted a nonparametric Bayesian approach allowing for an unbounded number of hidden units. In their construction, the authors assumed that every hidden unit $u_k^{(m)}$ has an associated parameter $\theta_k^{(m)}$ determining the unit's behaviour and a *popularity* parameter $\pi_k^{(m)}$ reflecting its connection probability. The prior probabilities on these parameters are:

$$\pi_k^{(m)} \sim \text{Beta}(\alpha\beta/K, \beta - \alpha\beta/K) \tag{3}$$

$$\theta_k^{(m)} \sim \alpha^{-1} B_0^{(m)} \tag{4}$$

where $B_0^{(m)}$ is a *base measure* of mass α acting as a prior distribution on hidden units behaviours. According to this formulation, we can represent layer m with the following discrete measure:

$$B^{(m)} = \sum_{k=1}^{K} \pi_k^{(m)} \delta_{\theta_k^{(m)}} \tag{5}$$

where δ_θ denotes a unit point mass at θ. Basically, the random measure $B^{(m)}$ is a function indicating which unit is present in layer m by assigning positive popularity to the specific representing parameters. When considering infinitely many hidden units by letting $K \to \infty$, the Beta distribution (3) on popularities degenerates and it results in a *Beta process* prior $B^{(m)} \sim \text{BP}(\beta, B_0^{(m)})$ on infinite layers of units, where β controls the expected sum of popularities (Thibaux and Jordan, 2007).

When determining the parents of a unit, we have to proceed according to the popularity associated with each potential parent. This means for unit $u_i^{(m-1)}$, that a binary variable is sampled for all units in layer m according to their respective Bernoulli distribution $Z_{i,k}^{(m-1,m)} \sim \text{Ber}(\pi_k^{(m)})$. The resulting parental connections for this unit can then be expressed in terms of discrete measure:

$$C_i^{(m-1,m)} = \sum_{k=1}^{\infty} Z_{i,k}^{(m-1,m)} \delta_{\theta_k^{(m)}} \tag{6}$$

where effective unit masses on parameters θ designate the actual parents. Since there are infinitely many parents to consider, the probability distribution of this measure can be represented as a *Bernoulli process* that we denote by $C_i^{(m-1,m)} \sim \text{BeP}(B^{(m)})$. In what follows, we refer to this type of measure as *connection measure* as they only indicate the existence of connections.

The conjugacy between the Beta and Bernoulli distributions extends to their stochastic process counterparts. When provided with a set of connection measures, the posterior distribution on hidden units' popularity π and parameters θ remains a Beta process. At this point, there are two possibilities: explicitly represent the popularity vector $\pi^{(m)}$, leading to a *stick-breaking* representation of the Beta process (Paisley et al., 2010), or marginalize the Beta process measure, producing an Indian buffet process prior on connections (Thibaux and Jordan, 2007). We adopt the latter representation, leading to the following marginalized posterior Bernoulli process on a newly introduced active unit:

$$C^{(m-1,m)}_{K^{(m-1)}+1} | C^{(m-1,m)}_{1\dots K^{(m-1)}} \sim \text{BeP}\left(\frac{\beta B_0^{(m)}}{\beta + K^{(m-1)}} + \frac{C^{(m-1,m)}}{\beta + K^{(m-1)}} \right) \quad (7)$$

where we defined $C^{(m-1,m)} = \sum_{i=1}^{K^{(m-1)}} C_i^{(m-1,m)}$ to simplify notation, which counts the number of connections of each parent. When considering units in layer $m-1$ as customers and units in layer m as dishes, the sequential process described by equation (7) coincides with the two-parameter Indian buffet process (Griffiths and Ghahramani, 2011) which we denote as $Z^{(m-1,m)} \sim \text{IBP}(\alpha, \beta)$.

3.2 Infinitely Deep Network

In certain cases, having a prior on graphical models able to generating every possible valid structure is highly desirable. Although the model presented in Section 3.1 assumes infinitely many units in a single hidden layer, its expressivity is limited due to the particular nature of DAG structures supported by the resulting prior. Nevertheless, each time an extra layer is added to the network, the number of supported structures also increases. By letting the number of layers $M \to \infty$, Adams et al. (2010) obtained a prior stochastic process, called the cascading Indian buffet process (CIBP), thus allowing greater flexibility.

The CIBP produces infinite sequences of binary matrices by recursively sampling IBPs where the number of active columns in a random matrix determines the number of rows in the next. When applied to belief network structures, this correspondence concerns the units' incoming and outgoing connections. For unit $u_k^{(m)}$, it means that row vector $Z_{k,\cdot}^{(m,m+1)}$ only appears when column vector $Z_{\cdot,k}^{(m-1,m)}$ is non-zero. The CIBP begins by sampling $Z^{(0,1)}$ according to an IBP(α, β) with $K^{(0)}$ rows. Even though there are infinitely many units per layer, the IBP only selects a finite subset of them with probability one. The sampling procedure continues by recursively sampling the connections $Z^{(m,m+1)} \sim \text{IBP}(\alpha, \beta)$ of the successive layers, each of them having an (almost surely) finite number of rows and columns.

3.3 Jumping Connections

In order to specify a prior assigning positive probability to every possible directed acyclic graph structures, we propose an extension to the CIBP allowing connections between non-consecutive layers. The downside of this limitation is that representing a fully connected graph with 3 variables would require adding a 4th unit in the intermediate layer to pass activities deterministically from the root variable to the leaf variable. Likewise, the more layers activities have to go-through, the less likely it is to see the appropriate chain of units added to the structure. The proposed solution to this problem consists in by-passing intermediate layers via what we call *jumping connections*.

The proposed process operates in two steps. First, a sequence of binary matrices $Z^{(0,1)}, Z^{(1,2)}, Z^{(2,3)}, \dots$ is drawn according to the CIBP, which results in $K^{(1)}, K^{(2)}, K^{(3)}, \dots$ active hidden units in the respective layers. We recall from Section 3.1 that what underlies the connection process for adjacent layer $m-1$ and m is the unknown popularity measure $B^{(m)}$ that has been marginalized. For the sake of consistency, the connection probability for units in layers $s \in \{0, \dots, m-2\}$ to connect to a unit in layer m should also be governed by the same measure $B^{(m)}$ which characterizes the layer. Therefore, we propose the following hierarchical prior:

$$B^{(s,m)} | C^{(m-1,m)}_{1\dots K^{(m-1)}} \sim \text{BP}\left(\beta', \frac{C^{(m-1,m)}}{\beta + K^{(m-1)}} \right) \quad (8)$$

where the base measure is the connection measures used in posterior update (7) with the novelty part $B_0^{(m)}$ removed. We also introduce β' to point out that layer-specific parameters could be defined.

As we did for equation (7), we can marginalize the posterior Beta process random measure to obtain a Bernoulli process representing the sequential process on jumping connections:

$$C^{(s,m)}_{K^{(s)}+1} | C^{(s,m)}_{1\dots K^{(s)}}, C^{(m-1,m)}_{1\dots K^{(m-1)}} \sim \text{BeP}\left(\frac{\beta'}{\beta' + K^{(s)}} \frac{C^{(m-1,m)}}{\beta + K^{(m-1)}} + \frac{C^{(s,m)}}{\beta' + K^{(s)}} \right) \quad (9)$$

where the parameter β' controls the impact of the observed connection frequencies between layer $m-1$ and m on which we condition.

The previous probability model on jumping connections is motivated in two ways. Firstly, the probabilities respect the underlying popularity vector associated to each layer. In fact, using a hierarchical BP has the advantage of allowing a flexible number of grandparent units by conditioning on these popularity vectors and it also remains consistent with the CIBP construction. Secondly, the model prevents jumping connections to activate new units and restricts them to already-active grandparent units. In other words, a unit can only connect to a new unconnected unit if this unit belongs to its immediate parent layer, thus preventing the total number of active units to diverge. Since new units can only connect to the network based on the underlying CIBP, using the extension will not modify the convergence properties.

3.4 Prior on Parameters

Defining a prior on belief networks also requires considering parameters uncertainty. To this end, we specified typical priors on the parameters, namely Gaussian prior distributions on the weights and biases, and gamma prior distributions on precisions. These priors have been mainly chosen for their conjugacy property with the likelihood function defined in equation (2).

4 INFERENCE

This section presents the Bayesian nonparametric inference procedure used to learn belief networks when provided with sets of observations. Since the posterior distribution on belief networks is generally highly complex, we rely on Markov chain Monte Carlo (MCMC) methods to draw samples from it. The state of the Markov chain contains the connections and weights of connected units, as well as their biases and precisions. Moreover, the particular values taken by hidden units to generate the observations are also unknown. Since we cannot marginalize these values analytically, they are also included as part of the Markov chain. In what follows, we present the transition operators related to unit activations and structure. For brevity, the conditional distributions for weights, precisions and biases are not included.

Sampling the Unit Activations. Unlike other parameters of the model, the posterior distribution on unit activation $u_k^{(m)}$ has no analytical form. Consequently, we followed the sampling method suggested by Adams et al. (2010) and used a specific variant of the multiple-try Metropolis-Hastings[1] of Liu et al. (2000). The approach consists first in sampling $2q-1$ proposals $u_1, \ldots, u_{2q-1}$ from the prior determined by the parent units, then computing the following acceptance ratio:

$$a_u = \min\left\{1, \frac{f(u_1) + \cdots + f(u_q)}{f(u_{q+1}) + \cdots + f(u_{2q-1}) + f(u_{k,n}^{(m)})}\right\} \tag{10}$$

where f is used to denote the likelihood function on unit u. When the proposal is accepted, the new value for $u_{k,n}^{(m)}$ is selected among $u_1, \ldots, u_q$ with probability proportional to their likelihood.

Sampling the Edges. To perform structure inference, various network proposals are generated by adding or removing edges from the current network and are accepted in a way that it remains consistent with the posterior distribution. When removing an edge, the action might render the parent unit *inactive*, implying we no longer need the information concerning that unit. On the other hand, adding an edge might activate a new unit in the network. In that case, we have to sample its deeper connections according to the extended CIBP prior, which might activate even more units and create new connections to already active units. These two cases actually refer to *singleton* units since they point to exactly one children unit and we deal with them in the second phase of structure inference. In the first phase, we deal with *non* singleton units, a case where the random outcome of adding or removing an edge does not influence any unit activation. These phases are applied by iterating on every active unit of the network.

Phase 1: We consider connection between candidate parent unit $u_k^{(m)}$ and unit $u_i^{(s)}$. To sample the existance of such connection according to the posterior distribution, we need to compute the prior probability of the edge $Z_{i,k}^{(s,m)}$. As mentioned in Section 3, there are two possible priors. When considering adjacent layers, $m = s+1$, the prior is the same as in the CIBP:

$$Z_{i,k}^{(s,m)} \sim \text{Ber}\left(\frac{n_{-i,k}^{(s,m)}}{\beta + K^{(s)} - 1}\right) \tag{11}$$

where $n_{-i,k}^{(s,m)}$ is the number of outgoing connections from unit $u_k^{(m)}$ to any unit in layer s, excluding unit i. When $m > s+1$, we have the following prior:

$$Z_{i,k}^{(s,m)} \sim \text{Ber}\left(\frac{\beta'}{\beta' + K^{(s)}} \frac{n_k^{(m-1,m)}}{\beta + K^{(m)}} + \frac{n_{-i,k}^{(s,m)}}{\beta' + K^{(s)}}\right) \tag{12}$$

which corresponds to the jumping connections that we propose as an extension to the CIBP. Notice that equation (12) becomes part of the likelihood when considering adjacent layers due to its hierarchical definition. To obtain the posterior probability of an edge, we also have to evaluate the units' likelihood function from $Z_{i,k}^{(s,m)} W_{i,k}^{(s,m)} \sim \mathcal{N}(\mu_e, 1/\rho_e)$ having the following mean and precision:

$$\mu_e = \frac{\sum_n u_{k,n}^{(m)}\left(\sigma^{-1}(u_{i,n}^{(s)}) - \xi_{i,k,n}^{(s,m)}\right)}{\sum_n (u_{k,n}^{(m)})^2} \tag{13}$$

$$\rho_e = \nu_i^{(s)} \sum_n (u_{k,n}^{(m)})^2 \tag{14}$$

where the exact acceptance probabilities are obtained after normalization.

Phase 2: We next consider adding or removing connection between unit $u_i^{(m-1)}$ and singleton parents with a Metropolis-Hastings operator using a birth/death process. With probability 1/2, we propose adding a new singleton parent by drawing it from the prior and accepting it with probability:

$$a_z^{(+)} = \frac{(K_\circ + 1)^{-2}\alpha\beta}{(\beta + K^{(m)} - 1)} \prod_{n=1}^{N} \frac{f(u_{k,n}^{(m)} | Z_{i,k}^{(m-1,m)} = 1)}{f(u_{k,n}^{(m)} | Z_{i,k}^{(m-1,m)} = 0)} \tag{15}$$

where $K_\circ$ is the current number of singleton parents connecting to unit $u_i^{(m-1)}$. On the other hand, if we do not propose to add a new parent unit, we consider removing one of the existing singleton parents by uniformly selecting among them. The acceptance probability in that case is:

$$a_z^{(-)} = \frac{(\beta + K^{(m)} - 1)}{K_\circ^{-2}\alpha\beta} \prod_{n=1}^{N} \frac{f(u_{k,n}^{(m)} | Z_{i,k}^{(m-1,m)} = 0)}{f(u_{k,n}^{(m)} | Z_{i,k}^{(m-1,m)} = 1)} \tag{16}$$

with $u_k^{(m)}$ representing the randomly selected singleton parent to deactivate. Notice that removing a singleton parent

[1] We used the MTM (II) algorithm with independent proposal function $T(\mathbf{x}, \cdot)$ defined as the prior distribution induced by parents and symmetric function $\lambda(\mathbf{x}, \mathbf{y}) = [T(\mathbf{x}, \mathbf{y})T(\mathbf{y}, \mathbf{x})]^{-1}$.

having grandchildren units is impossible as it would lead to jumping connections having null probabilities under the prior, making the transition probability equals to 0.

5 EXPERIMENTAL RESULTS

5.1 Density Estimation

We now compare the benefits of using the extended CIBP (eCIBP) on learning graphical models as opposed to the vanilla version of the CIBP. Theoretically, the extended version should allow infering smaller, more densely-connected networks to explain a given dataset. We evaluated both approaches on five density estimation tasks, two synthetic and three real-world datasets, where posterior belief network samples were used to generate fantasy data. The predictive performances have been measured by estimating the Kullback-Leibler (KL) divergence between test sets and fantasy data obtained from the learned models (Pérez-Cruz, 2008). For comparison purposes, we include results from other learning methods: the Dirichlet process mixture of Gaussians (DPMoG), a nonparametric Bayesian approach used to learn density functions composed of infinitely many Gaussians (Görür and Edward Rasmussen, 2010); kernel density estimation (KDE) using Gaussian kernels with automatic bandwidth selection based on local leave-one-out likelihood criterion (Barnard, 2010)

The learning procedure of eCIBP consists in applying the MCMC operators described in Section 4 to draw samples from the posterior distribution on belief networks. For the vanilla CIBP, the procedure is exactly the same, except for jumping connections transitions. Both priors were specified with identical fixed hyperparameters $\alpha = \beta = 1$. The additional hyperparameter for eCIBP has been set to $\beta' = 1$. The priors on weights, precisions and biases were $\mathcal{N}(0, 1)$ and $\text{Gamma}(1, 1)$. These values were selected to obtain a nonparametric Bayesian prior that does not provide precise information about the model and thus force the data to bring the relevant information in the posterior distribution. Finally, all Markov chains were initialized with networks containing only D observable units in the first layer and having no hidden units.

Experiments on Synthetically Generated Data. We first evaluated the learning approaches on synthetic data exhibiting obvious structures, the *ring* and *pinwheel* data sets. The ring data are generated by uniformly sampling points on the unit circle and apply Gaussian noise to the radius. On the other hand, the pinwheel data are generated by stetching and rotating four Gaussian distributions, resulting in a spiral shape. Both training sets contained 2000 points and their respective generative process had to be encapsulated in the learned belief network structures and parameters.

For the inference, we ran Markov chains for 500,000 iterations. The burn-in period was set to 250,000 iterations and a thinning of 100 has been applied to compile results. To compare the posterior structure complexity of the learned belief networks, we used the total number of hidden units and the total number of edges composing the structure. Since simpler structures might be obtained at the cost of predictive performance, we evaluated the difference of the learned models from the true distribution by estimating the KL divergence with fantasy data and fixed test sets of 2000 data points. In Table 1, we report the average posterior KL divergence estimations computed for all chains and include 2 standard deviations to reflect the variance of the posterior. As a point of reference, the estimated KL divergence of the training set from the test set is 0.003 on ring and 0.025 on pinwheel. Additionally, the posterior structure complexity can be compared based on the total number of hidden units and the number of connections reported in Table 2.

Experiments on Real-World Data To assess the performances of our approach on more realistic learning tasks, we conducted experiments on three real-world datasets, *Geyser*, *Iris* and *Abalone*. The first dataset, *Geyser*, consists of 2-dimensional data collected from the Old Faithful geyser (Scott, 1992). We used the eruption durations along with the waiting time to next eruptions and learned the joint distribution using all approaches. On this problem, 272 observations were available and halved to make the training and test sets, both containing 136 samples. The second dataset, *Iris*, contains 4 physical measurements from different types of iris plants (Fisher, 1936). From the 150 observations available, 75 were used for training and 75 for testing. Lastly, the *Abalone* dataset consists of various physical measurements of abalone shells along with the age and sex of each individual (Blake and Merz, 1998). In particular, it has 7 positive continuous dimensions and 2 discrete dimensions. For this one, the training set had 2000 examples as well as the test set.

To learn the belief network generative model for these datasets, we ran Markov chains for 350,000 iterations. A burn-in period of 200,000 was used in this case, together with a thinning of 100 to compile results. As for the synthetic data, we present the predictive precision in Table 1 and structure complexity in Tables 2. For real-world datasets, the estimated KL divergence of the training set from the test set is less than 0.001 on Geyser, 0.113 on Iris and 0.109 on Abalone.

5.2 Structure Identification

We evaluated the structure learning performance of eCIBP in identifying the exact structure of a known belief network. Depending on the data, however, the original belief network might not be identifiable since multiple structures could be equally good to explain the data. To reduce the effect of learning the posterior distribution on an equivalence class of belief networks, we carried out experiments on a simpler network, facilitating the evaluation of the learned structures.

The network used for this experiment produces bidimensional data with 2 hidden units in layer 1 and 3 hidden units in layer 2. Using the notation of Section 2, the parameters of

Table 1: Kullback-Leibler Divergence Estimations

Dataset	CIBP	eCIBP	DPMoG	KDE
Ring (2)	0.049 ± 0.055	$\mathbf{0.030 \pm 0.034}$	0.051 ± 0.029	0.085 ± 0.034
Pinwheel (2)	0.162 ± 0.049	0.161 ± 0.041	$\mathbf{0.154 \pm 0.080}$	0.216 ± 0.043
Geyser (2)	0.078 ± 0.150	$\mathbf{0.075 \pm 0.145}$	0.077 ± 0.120	0.143 ± 0.099
Iris (4)	0.207 ± 0.333	$\mathbf{0.177 \pm 0.280}$	0.231 ± 0.269	0.305 ± 0.227
Abalone (9)	0.074 ± 0.080	$\mathbf{0.070 \pm 0.108}$	4.760 ± 0.220	2.934 ± 0.134

Table 2: Posterior Means and Deviations on Structure Complexity

	Number of units		Number of edges	
Dataset	CIBP	eCIBP	CIBP	eCIBP
Ring (2)	11.6 ± 4.1	$\mathbf{9.3 \pm 3.2}$	$\mathbf{20.3 \pm 8.4}$	20.6 ± 7.3
Pinwheel (2)	22.4 ± 3.0	$\mathbf{19.7 \pm 1.6}$	105.4 ± 11.1	$\mathbf{104.1 \pm 9.3}$
Geyser (2)	9.1 ± 4.3	$\mathbf{8.7 \pm 3.9}$	$\mathbf{13.1 \pm 9.9}$	$\mathbf{13.1 \pm 9.5}$
Iris (4)	14.2 ± 4.4	$\mathbf{10.3 \pm 3.4}$	28.4 ± 11.4	$\mathbf{20.5 \pm 7.2}$
Abalone (9)	54.2 ± 12.3	$\mathbf{45.2 \pm 11.2}$	233.8 ± 23.0	$\mathbf{210.4 \pm 23.8}$

the structure are:

$$W^{(0,1)} = \mathbf{I}$$

$$W^{(1,2)} = \begin{bmatrix} 1 & 0.3 & 0 \\ 0 & 0.3 & 1 \end{bmatrix}$$

$$W^{(0,2)} = \begin{bmatrix} -0.15 & 0 & -0.15 \\ 0.3 & 0.3 & 0 \end{bmatrix}$$

The precisions are 10^4 in the first 2 layers, 10^{-2} in the top layer and all biases are set to 0. When sampling this network, the generated data form a pattern that resemble a rotated cube.

The learning procedure used for this experiment was identical to the one used for synthetical datasets. We used training and test sets of 2000 data points and generated fantasy data from posterior models. The average KL divergence for this data set was 0.203 ± 0.054, meaning that predictions were accurate. Concerning the posterior distribution on network features, the total number of units and edges was $7.2{\pm}0.8$ and $11.2{\pm}1.1$ respectively, while the network depth was 3.1 ± 0.6. These results are close to the original model having 7 units, 10 edges and a depth of 3. When looking at posterior structures, we observe that all existing edges are often identified with appropriate weight. However, an extra edge connecting a top layer node to an observable node is frequently incorrectly introduced with a small weight. The precisions in the intermediate layer are underestimated while the ones in the observable layer are overestimated and is the main source of KL divergence.

As a measure of comparison, we also performed network inference with CIBP and obtained an average KL divergence of $0.207{\pm}0.043$. The posterior distributions on network features are: 10.2 ± 2.4 on units; 20.2 ± 5.0 on edges; and 4.2 ± 0.9 on depth. These results are in line with our hypothesis that more complex networks are produced with this prior compared to the extended version.

6 DISCUSSION

In this paper, we extended the cascading Indian buffet process (CIBP) to learn arbitrary directed acyclic graphs (DAG) and applied it to structure learning of belief networks. By introducing connections between non-consecutive layers, the proposed approach was able to learn structures with fewer hidden units than the cascading Indian buffet process while retaining predictive performances. This is mainly attributable to the increased flexibility provided by the extension, combined with the fact that the method implicitly incorporates a model complexity tradeoff based on the data.

On density estimation tasks, we observed that the extended cascading Indian buffet process (eCIBP) and the Dirichlet process mixture of Gaussian (DPMOG) obtained similar results. This suggest that eCIBP could be used to develop efficient density estimation algorithms that also extract compact graphical model representations. However, the computation time associated with the proposed Markov Chain Monte Carlo inference procedure limits its application to relatively small problems.

In practice, the computational cost attributed to the extended version was not significantly higher comparatively to the original version. During the experiments, we observed that on average it roughly takes twice the amount of time for the eCIBP to complete a Markov chain. In fact, since the CIBP tends to produce larger networks to compensate the unallowed connections, it also requires more computation time, which therefore balanced the computational cost of the two methods. However, for harder tasks, the difference should increase as more and more units are required to reproduce complex joint distributions.

This work is a step towards learning directed acyclic graphs with nonparametric Bayesian methods. As part of our future work, we now plan to develop a nonparametric Bayesian prior on infinite DAGs producing a simpler posterior on the underlying Bernoulli probabilities.

References

Adams, R. P., Wallach, H. M., and Ghahramani, Z. (2010). Learning the structure of deep sparse graphical models. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*.

Barnard, E. (2010). Maximum leave-one-out likelihood for kernel density estimation.

Blake, C. and Merz, C. (1998). {UCI} repository of machine learning databases.

Chen, B., Polatkan, G., Sapiro, G., Dunson, D., and Carin, L. (2011). The hierarchical beta process for convolutional factor analysis and deep learning.

Ferguson, T. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7(2):179–188.

Frey, B. (1997). Continuous sigmoidal belief networks trained using slice sampling. *Advances in Neural Information Processing Systems*, pages 452–458.

Friedman, N. and Koller, D. (2003). Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine learning*, 50(1):95–125.

Görür, D. and Edward Rasmussen, C. (2010). Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4):653–664.

Griffiths, T. and Ghahramani, Z. (2011). The indian buffet process: an introduction and review. *Journal of Machine Learning Research*, 12(April):1185–1224.

Hjort, N. (1990). Nonparametric bayes estimators based on beta processes in models for life history data. *The Annals of Statistics*, pages 1259–1294.

Jordan, M. (2010). Hierarchical models, nested models and completely random measures.

Lauritzen, S. (1996). *Graphical models*, volume 17. Oxford University Press, USA.

Liu, J., Liang, F., and Wong, W. (2000). The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association*, pages 121–134.

MacEachern, S. and Müller, P. (1998). Estimating mixture of dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):223–238.

Paisley, J. and Carin, L. (2009). Nonparametric factor analysis with beta process priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 777–784. ACM.

Paisley, J., Zaas, A., Woods, C., Ginsburg, G., and Carin, L. (2010). A stick-breaking construction of the beta process. In *Proceedings of the International Conference on Machine learning (ICML)*. Citeseer.

Pérez-Cruz, F. (2008). Kullback-leibler divergence estimation of continuous distributions. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 1666–1670. Ieee.

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102(2):145–158.

Scott, D. (1992). Multivariate density estimation. *Multivariate Density Estimation, Wiley, New York, 1992*, 1.

Teh, Y., Jordan, M., Beal, M., and Blei, D. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Thibaux, R. and Jordan, M. (2007). Hierarchical beta processes and the indian buffet process. In *International Conference on Artificial Intelligence and Statistics*. Citeseer.

Whittaker, J. (1990). *Graphical models in applied multivariate statistics*, volume 16. Wiley New York.

Wood, F., Griffiths, T., and Ghahramani, Z. (2006). A nonparametric bayesian method for inferring hidden causes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, volume 22. Citeseer.