The Indian Chefs Process

Patrick Dallaire[12] **Luca Ambrogioni**[3] **Ludovic Trottier**[1] **Umut Güçlü**[3] **Max Hinne**[3]
Philippe Giguère[2] **Brahim Chaib-Draa**[2] **Marcel van Gerven**[3] **Francois Laviolette**[2]
SmartyfAI[1], Université Laval[2], Radboud University[3]
{patrick.dallaire, ludovic.trottier}@smartyfai.com
{philippe.giguere, chaib, francois.laviolette}@ift.ulaval.ca
{l.ambrogioni, u.guclu, m.hinne, marcel.vangerven}@donders.ru.nl

Abstract

This paper introduces the Indian chefs process (ICP) as a Bayesian nonparametric prior on the joint space of infinite directed acyclic graphs (DAGs) and orders that generalizes the Indian buffet process. As our construction shows, the proposed distribution relies on a latent Beta process controlling both the orders and outgoing connection probabilities of the nodes, and yields a probability distribution on sparse infinite graphs. The main advantage of the ICP over previously proposed Bayesian nonparametric priors for DAG structures is its greater flexibility. To the best of our knowledge, the ICP is the first Bayesian nonparametric model supporting every possible DAG involving latent nodes. We demonstrate the usefulness of the ICP on learning the structure of deep generative sigmoid networks as well as convolutional neural networks.

1 INTRODUCTION

In machine learning and statistics, the directed acyclic graph (DAG) is a common modelling choice for expressing relationships between objects. Prime examples of DAG-based graphical models include Bayesian networks, feed-forward neural networks, causal networks, deep belief networks, dynamic Bayesian networks and hidden Markov models, to name a few. Learning the unknown structure of these models presents a significant learning challenge, a task that is often avoided by fixing the structure to a large and hopefully sufficiently expressive model. *Structure learning* is a model selection problem in which one estimates the underlying graphical structure of the model. Over the years, researchers have explored a great variety of approaches to this problem [1, 2, 3, 4, 5], from frequentist to Bayesian, and some using pure heuristic-based search, but the vast majority is limited to finite parametric models.

Bayesian nonparametric learning methods are appealing alternatives to their parametric counterparts, because they offer more flexibility when dealing with generative models of unknown dimensionality [6]. Instead of looking for specific finite-dimensional models, the idea is rather to define probability measures on infinite-dimensional spaces and then infer the finite subset of active dimensions explaining the data. Over the past years, there has been extensive work on constructing flexible Bayesian nonparametric models for various types of graphical models, allowing complex hidden structures to be learned from data. For instance, [7] developed a model for infinite latent conditional random fields while [8] proposed an infinite mixture of fully observable finite-dimensional Bayesian networks. In the case of time series, [9] developed the infinite hidden Markov random field model and [10] proposed an infinite dynamic Bayesian network with factored hidden states. Another interesting model is the infinite factorial dynamical model of [11] representing the hidden dynamics of a system with infinitely many independent hidden Markov models.

The problem of learning networks containing hidden structures with Bayesian nonparametric methods has also received attention. The cascading Indian buffet process (CIBP) of [12] is a Bayesian nonparametric prior over infinitely deep and infinitely broad layered network structures. However, the CIBP does not allow connections from non-adjacent layers, yielding a restricted prior over infinite DAGs. The extended CIBP (ECIBP) is an extension of the previous model which seeks to correct this limitation and support a larger set of DAG structures [13]. However, the ECIBP has some drawbacks: the observable nodes are confined to a unique layer placed at the bottom of the network, which prevents learning the order of the nodes or have observable inputs. An immediate consequence of this is the impossibility for an observable unit

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

to be the parent of any hidden unit or any other observable unit, which restricts the support of the prior over DAGs and makes their application to supervised deep learning problematic.

In the context of deep learning, structure learning is often part of the optimization. Recently, [14] proposed a method that enforces the model to dynamically learn more compact structures by imposing sparsity through regularization. While sparsity is an interesting property for large DAG-based models, their method ignores the epistemic uncertainty about the structure. Structure learning for probabilistic graphical models can also be applied in deep learning. For instance, [15] have demonstrated that deep network structures can be learned through the use of Bayesian network structure learning strategies. To our knowledge, no Bayesian nonparametric structure learning methods have been applied to deep learning models.

This paper introduces the Indian chefs process (ICP), a new Bayesian nonparametric prior for general DAG-based structure learning, which can equally be applied to perform Bayesian inference in probabilistic graphical models and deep learning. The proposed distribution has a support containing all possible DAGs, admits hidden and observable units, is layerless and enforces sparsity. We present its construction in Section 2 and describe a learning method based on Markov chain Monte Carlo in Section 3. In Section 4, we use the ICP as a prior in two Bayesian structure learning experiments: in the first, we compute the posterior distribution on the structure and parameters of a deep generative sigmoid network and in the second we perform structure learning in convolutional neural networks.

2 BAYESIAN NONPARAMETRIC DIRECTED ACYCLIC GRAPHS

We construct a probability distribution over DAGs and orders by adopting the methodology followed by [16]. We first define a distribution over finite-dimensional structures, then obtain the final distribution by evaluating it as the structure size grows to infinity.

Let $G = (V, Z)$ be a DAG where $V = \{1, \dots, K\}$ is the set of nodes and $Z \in \{0,1\}^{K\times K}$ is the adjacency matrix. We introduce an ordering $\boldsymbol{\theta}$ on the nodes so that the direction of an edge is determined by comparing the order value of each node. A connection $Z_{ki} = 1$ is only allowed when $\theta_k > \theta_i$, meaning that higher order nodes are parents and lower order nodes are children. Notice that this constraint is stronger than acyclicity since all $(Z, \boldsymbol{\theta})$ combinations respecting the order value constraint are guaranteed to be acyclic, but an acyclic graph can violate the ordering constraint.

We assume that both the adjacency matrix Z and the ordering $\boldsymbol{\theta}$ are random variables and develop a Bayesian framework reflecting our uncertainty. Accordingly, we assign a *popularity* parameter π_k and an order value θ_k, called *reputation*, to every node k in G based on the following model:

$$\theta_k \sim \mathcal{U}(0,1) \tag{1}$$

$$\pi_k \mid \alpha, \gamma, \phi, K \sim \text{Beta}\left(\frac{\alpha\gamma}{K} + \phi\mathbb{I}(k \in O), \alpha - \frac{\alpha\gamma}{K}\right) \tag{2}$$

$$Z_{ki} \mid \pi_k, \theta_k, \theta_i \sim \text{Bernoulli}\left(\pi_k \mathbb{I}(\theta_k > \theta_i)\right) . \tag{3}$$

Here, $\mathbb{I}$ denotes the indicator function, $\mathcal{U}(a,b)$ denotes the uniform distribution on interval $[a,b]$ and $O \subseteq V$ is the set of *observed* nodes. In this model, the popularities reflected by π control the outgoing connection probability of the nodes, while respecting the *total order* imposed by $\boldsymbol{\theta}$. Moreover, the Beta prior parametrization in Eq. (2) is motivated by the Beta process construction of [17], where Eq. (1) becomes the *base distribution*, and is convenient when evaluating the limit in Section 2.1. Also, α and γ correspond to the usual parameters defining a Beta process and the purpose of the new parameter ϕ is to control the popularity of the observable nodes and ensure a non-zero connection probability when required.

Under this model, the conditional probability of the adjacency matrix Z given the popularities $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^K$ and order values $\boldsymbol{\theta} = \{\theta_k\}_{k=1}^K$ is:

$$p(Z \mid \boldsymbol{\pi}, \boldsymbol{\theta}) = \prod_{k=1}^{K}\prod_{i=1}^{K} p(Z_{ki} \mid \pi_k, \theta_k, \theta_i) . \tag{4}$$

The adjacency matrix Z may contain connections for nodes that are not of interest, i.e. nodes that are not ancestors of any observable nodes. Formally, we define $A \subseteq V$ as the set of *active* nodes, which contains all observable nodes O and the ones having a directed path ending at an observable node.

When solely considering connections from A to A, i.e. the adjacency submatrix Z_{AA} of the A-induced subgraph of G, Eq. (4) simplifies to:

$$p(Z_{AA} \mid \boldsymbol{\pi}, \boldsymbol{a}, \boldsymbol{\theta}) = \prod_{k\in A} \pi_k^{m_k} \left(1 - \pi_k\right)^{a_k - m_k} , \tag{5}$$

where $m_k = \sum_{i\in A} Z_{ki}$ denotes the number of outgoing connections from node k to any active nodes, $a_k = \sum_{j\in A} \mathbb{I}(\theta_j < \theta_k)$ denotes the number of active nodes having an order value strictly lower than θ_k and $\boldsymbol{a} = \{a_k\}_{k=1}^K$. At this point, we marginalize out the popularity vector $\boldsymbol{\pi}$ in Eq. (5) with respect to the prior, by using the conjugacy of the Beta and Binomial distribu-

tions, and we get:

$$p(Z_{AA} \mid \alpha, \gamma, \phi, \boldsymbol{a}, \boldsymbol{\theta}) = \prod_{k \in H} \frac{[\frac{\alpha\gamma}{K}]^{\overline{m_k}} [\alpha - \frac{\alpha\gamma}{K}]^{\overline{a_k - m_k}}}{\alpha^{\overline{a_k}}} \prod_{k \in O} \frac{[\frac{\alpha\gamma}{K} + \phi]^{\overline{m_k}} [\alpha - \frac{\alpha\gamma}{K}]^{\overline{a_k - m_k}}}{[\alpha + \phi]^{\overline{a_k}}}, \quad (6)$$

where $x^{\overline{n}} = x(x+1)\cdots(x+n-1)$ is the Pochhammer symbol denoting the rising factorial and $H = A \setminus O$ is the set of active hidden nodes. This equation is analogous to Eq.(10) in [16] showing Beta-Binomial distributions.

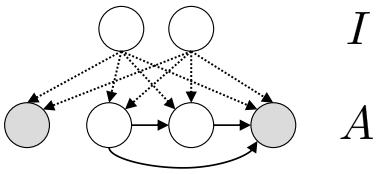


Figure 1: Graph with active nodes A and inactive nodes I. Solid arrows are connections among the active subgraph with hidden white nodes and gray observed nodes. Dashed arrows indicates the connections that must be zero to have exactly set I as the inactive set.

The set of active nodes A contains all observable nodes as well as their ancestors, which means there exists a part of the graph G that is disconnected from A. Let us denote by $I = V \setminus A$ the set of *inactive* nodes. Considering that the A-induced subgraph is effectively maximal, then this subgraph must be properly isolated by some envelope of no-connections Z_{IA} containing only zeros as in Fig.1. The joint probability of submatrices Z_{AA} and Z_{IA} is:

$$p(Z_{AA}, Z_{IA} \mid \alpha, \gamma, \phi, \boldsymbol{a}, \boldsymbol{\theta}) = p(Z_{AA} \mid \alpha, \gamma, \phi, \boldsymbol{a}, \boldsymbol{\theta}) \cdot \prod_{k \in I} \frac{[\alpha - \frac{\alpha\gamma}{K}]^{\overline{a_k}}}{\alpha^{\overline{a_k}}} \quad (7)$$

where the number of negative Bernoulli trials a_k depends on θ_k itself and $\boldsymbol{\theta}_A$. Notice that since the submatrices Z_{AI} and Z_{II} contain uninteresting and unobserved binary events, they are trivially marginalized out of $p(Z)$.

One way to simplify Eq. (7) is to marginalize out the order values $\boldsymbol{\theta}_I$ of the inactive nodes with respect to (1). To do so, we first sort the active node orders ascendingly in vector $\tilde{\boldsymbol{\theta}}_A$ and augment it with the extrema $\tilde{\theta}_0 = 0$ and $\tilde{\theta}_{K^+ + 1} = 1$, where we introduce $K^+ = |A|$ to denote the number of active nodes. We slightly abuse notation here since these extrema do not refer to any nodes and are only used to compute interval lengths. This provides us with all relevant interval boundaries, including the absolute boundaries implied by Eq. (1). We refer to the j^{th} smallest value of this vector as $\tilde{\theta}_j$. Based on the previous notation, the probability for an inactive node to lie between two active nodes is simply $\tilde{\theta}_{j+1} - \tilde{\theta}_j$. Using this notation, we have the following marginal probability:

$$p(\tilde{Z}_{AA}, Z_{IA}, \tilde{\boldsymbol{\theta}}_A \mid \alpha, \gamma, \phi) = \frac{(K-D)^{\underline{K^+ - D}}}{K^+!} \left(\sum_{j=0}^{K^+} (\tilde{\theta}_{j+1} - \tilde{\theta}_j) \frac{[\alpha(1 - \frac{\gamma}{K})]^{\overline{j}}}{\alpha^{\overline{j}}} \right)^{K^-} \prod_{k \in H} \frac{[\frac{\alpha\gamma}{K}]^{\overline{m_k}} [\alpha - \frac{\alpha\gamma}{K}]^{\overline{a_k - m_k}}}{\alpha^{\overline{a_k}}} \prod_{k \in O} \frac{[\frac{\alpha\gamma}{K} + \phi]^{\overline{m_k}} [\alpha - \frac{\alpha\gamma}{K}]^{\overline{a_k - m_k}}}{[\alpha + \phi]^{\overline{a_k}}}, \quad (8)$$

where $K^- = |I|$ denotes the number of inactive nodes, $x^{\underline{n}} = x(x-1)\ldots(x-n+1)$ symbolizes the falling factorial and $\tilde{Z}_{AA}$ is a reordering of the adjacency matrix according to $\tilde{\boldsymbol{\theta}}_A$. The latter is used because, due to the exchangeability of our model, the joint probability on both the adjacency matrix and active order values can cause problems regarding the index k of the nodes. By using this many-to-one transformation, we obtain a probability distribution on an equivalence class of DAGs that is analog to the *lof* function used by [16]. The number of permutations mapping to this sorted representation is accounted for by the normalization constant $(K-D)^{\underline{K^+ - D}}(K^+!)^{-1}$.

2.1 From Finite to Infinite DAGs

An elegant way to construct Bayesian nonparametric models is to consider the infinite limit of finite parametric Bayesian models [18]. Following this idea, we revisit the model of Section 2 so that G now contains infinitely many nodes. To this end, we evaluate the limit as $K \to \infty$ of Eq. (8), yielding the following probability distribution:

$$p(\tilde{Z}_{AA}, Z_{IA}, \tilde{\boldsymbol{\theta}}_A | \alpha, \gamma, \phi, O) = \frac{1}{K^+!} \exp\left(-\alpha\gamma \sum_{j=1}^{K^+} (\tilde{\theta}_{j+1} - \tilde{\theta}_j) \big[\psi(\alpha + j) - \psi(\alpha)\big] \right) \prod_{k \in H} \alpha\gamma \frac{(m_k - 1)!}{(\alpha + a_k - m_k)^{\overline{m_k}}} \prod_{k \in O} \frac{\phi^{\overline{m_k}} \alpha^{\overline{a_k - m_k}}}{[\alpha + \phi]^{\overline{a_k}}}, \quad (9)$$

where ψ is the digamma function. Eq. (9) is the proposed marginal probability distribution on the joint space of infinite DAGs and continuous orders.

2.2 The Indian Chefs Process

Now that we have the probability distribution (9), we want to draw random active subgraphs from it. This section introduces the Indian chefs process (ICP), a stochastic

process serving this purpose. In the ICP metaphor, chefs draw inspiration from other chefs, based on their *popularity* and *reputation*, to create the menu of their respective restaurant. This creates inspiration maps representable with directed acyclic graphs. ICP defines two types of chefs: 1) the star chefs (corresponding to observable variables) which are introduced iteratively and 2) the regular chefs (corresponding to hidden variables) which appear only when another chef selects them as a source of inspiration.

The ICP starts with an empty inspiration map as its initial state. The infinitely many chefs can be thought of as lying on a unit interval of reputations. Every chef has a fraction of the infinitely many chefs above him and this fraction is determined by the chef's own reputation.

The general procedure at iteration t is to introduce a new star chef, denoted i, within a fully specified map of inspiration representing the connections of the previously processed chefs. The very first step is to draw a reputation value from $\theta_i \sim \mathcal{U}(0,1)$ to determine the position of the star chef in the reputation interval. Once chef i is added, sampling the new inspiration connections is done in three steps.

Backward proposal Step one consists in proposing *star* chef i as an inspiration to *all* the a_i chefs having a lower reputation than chef i. To this end, we can first sample the total number of inspiration connections with:

$$q_i \sim \text{Binomial}\left(a_i, \frac{\phi}{\alpha+\phi}\right), \tag{10}$$

and then uniformly pick one of the $\binom{a_i}{q_i}$ possible configurations of inspiration connections.

Selecting existing chefs In step two, chef i considers *any* already introduced chefs of higher reputation. The probability for candidate chef k to become an inspiration for i is:

$$Z_{ki} \sim \text{Bernoulli}\left(\frac{m_k + \phi\mathbb{I}(k \in \text{star chefs})}{\alpha + a_k - 1 + \phi\mathbb{I}(k \in \text{star chefs})}\right), \tag{11}$$

where a_k includes the currently processed chef i.

Selecting new chefs The third step allows chef i to consider completely new *regular* chefs as inspirations in every single interval above i. The number of new regular chefs K_j^{new} to add in the j^{th} reputation interval above i follows probability distribution:

$$K_j^{new} \sim \text{Poisson}\left(\frac{(\tilde{\theta}_{j+1} - \tilde{\theta}_j)\alpha\gamma}{\alpha + a_j - 1}\right), \tag{12}$$

where the new regular chefs are independently assigned a random reputation drawn from $\mathcal{U}(\tilde{\theta}_j, \tilde{\theta}_{j+1})$. The *regular* chefs introduced during this step will be processed one by one using step two and three. Once all newly introduced regular chefs have been processed, the next iteration $t+1$ can begin with step one, a step reserved to star chefs only.

2.3 Some properties of the distribution

To better understand the effect of the hyperparameters on the graph properties, we performed an empirical study of some relations between the hyperparameters, the expected number of active nodes $\mathbb{E}[K^+|\alpha,\gamma]$ and the expected number of active edges $\mathbb{E}[E^+|\alpha,\gamma]$, where E^+ is the number of elements in Z_{AA}. Figure 3(a) depicts level curves of $\mathbb{E}[K^+|\alpha,\gamma]$ for the case of only 1 observable placed at $\theta_k = 0$. The figure shows that several combinations of α and γ leads to the same expected number of active nodes. Notice that fixing one hyperparameter, either α or γ, and selecting the expected number of nodes, one can retrieve the second hyperparameter that matches the relationship. We used this fact in the construction of Figure 3(b) where the unshown parameter γ could be calculated. In Figure 3(b), we illustrate the effect of α on $\mathbb{E}[E^+|\alpha,\gamma]$ which essentially shows that smaller values of α increase the graph density. For additional intuition on the effect of α and γ, we refer the reader to the two-parameter version of the Indian buffet process and its underlying Beta process [19, 20, 16].

When using Bayesian nonparametric models, we are actually assuming that the generative model of the data is infinite-dimensional and that only a finite subset of the parameters are involved in producing a finite set of data. The effective number of parameters explaining the data corresponds to the model complexity and usually scales logarithmically with respect to the sample size. Unlike most Bayesian nonparametric models, the ICP prior scales according to the number of observed nodes added to the network. In Figure 3, we show how the expected number of active hidden nodes increases as function of the number of observable nodes.

2.4 Connection to the Indian Buffet Process

There exists a close connection between the Indian Chefs Process (ICP) and the Indian Buffet Process (IBP). In fact, our model can be seen as a generalization of the IBP. Firstly, all realizations of the IBP receive a positive probability under the ICP. Secondly, the two-parameter IBP is recovered, at least conceptually, when altering the prior on order values (see Eq. (1)) so that all observed nodes are set to reputation $\theta = 0$ and all hidden nodes are set to reputation $\theta = 1$. This way, connections are prohibited between hidden nodes and between observable

nodes, while hidden-to-observable connections are still permitted as depicted in Fig.2.

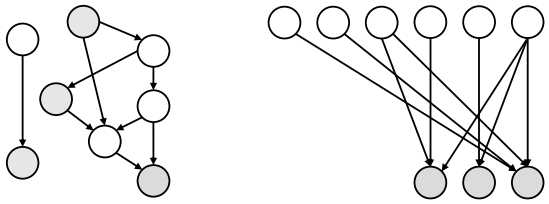


Figure 2: Left represents a random graph from an ICP (a directed acyclic graph) and right represents a sample from an IBP (a directed bipartite graph). Gray nodes are observable and white nodes are hidden. The layerless ICP can act as an IBP when all white nodes are set to $\theta = 0.0$ (top) and gray nodes are set to $\theta = 1.0$ (bottom).

3 STRUCTURE LEARNING

In this section, we present some Markov Chain Monte Carlo (MCMC) operators to perform Bayesian inference over structures following an ICP prior. We propose a reversible jump MCMC algorithm producing random walks on Eq. (9) [21]. This algorithm works in three phases: the first resamples graph connections without adding or removing any nodes, the second phase is a birth-death process on nodes and the third one only involves the order.

The algorithm itself uses the notion of *singleton* and *orphan* nodes. A node is a singleton when it only has a unique active child. Thus, removing its unique connection would disconnect the node from the active subgraph. Moreover, a node is said to be an orphan if it does not have any parents.

Within model moves on adjacency matrix: We begin by uniformly selecting a node i from the active subgraph. Here, the set of parents to consider for i comprises all non-singleton active nodes having an order value greater than θ_i. This set includes both current parents and candidate parents. Then, for each parent k, we Gibbs sample the connections using the following conditional probability:

$$p(\tilde{Z}_{ki} = 1 | \tilde{Z}_{AA}^{\neg ki}, \boldsymbol{\theta}_A) = \frac{m_k^{\neg i} + \phi \mathbb{I}(k \in O)}{\alpha + a_k - 1 + \phi \mathbb{I}(k \in O)}, \tag{13}$$

where $m_k^{\neg i}$ is the number of outgoing connections of node k excluding the connection to node i and $\tilde{Z}_{AA}^{\neg ki}$ has element ki removed. Also, all connections not respecting the order are prohibited and therefore have an occurrence probability of 0, and the same applies to singleton parent moves which are trans-dimensional.

Trans-dimensional moves on adjacency matrix: We begin with a random uniform selection of node i in the active subgraph and, with equal probability, propose either a *birth* or a *death* move.

In the birth case, we activate node k by connecting it to node i. The order θ_k is determined by uniformly selecting an insertion interval above θ_i. Assuming node i is also the i^{th} element in $\tilde{\boldsymbol{\theta}}_A$, we have $n_i = K^+ - i + 1$ possible intervals, including zero-length intervals. Let us assume that j and $j+1$ are the two nodes between which k is to be inserted. Then, we obtain the candidate order value of the new node by sampling $\theta_k \sim \mathcal{U}(\tilde{\theta}_j, \tilde{\theta}_{j+1})$. The Metropolis-Hastings acceptance ratio here is:

$$a_{birth} = \min \left\{ 1, \frac{p(\tilde{Z}'_{A'A'}, Z'_{I'A'}, \tilde{\boldsymbol{\theta}}'_{A'} \mid \alpha, \gamma, \phi, O)}{p(\tilde{Z}_{AA}, Z_{IA}, \tilde{\boldsymbol{\theta}}_A \mid \alpha, \gamma, \phi, O)} \cdot \frac{(\tilde{\theta}_{j+1} - \tilde{\theta}_j)(n_i + 1)K^+}{K_i^* + 1} \right\}, \tag{14}$$

where K_i^* is the number of singleton-orphan parents of i and $n_i = \sum_{j \in A} \mathbb{I}(\theta_j > \theta_i)$ is the number of active nodes above i.

In the death case, we uniformly select one of the K_i^* singleton-orphan parents of i if $K_i^* > 0$ and simply do nothing in case there exists no such node. Let k be the parent to disconnect and consequently deactivate. The Metropolis-Hastings acceptance ratio for this move is:

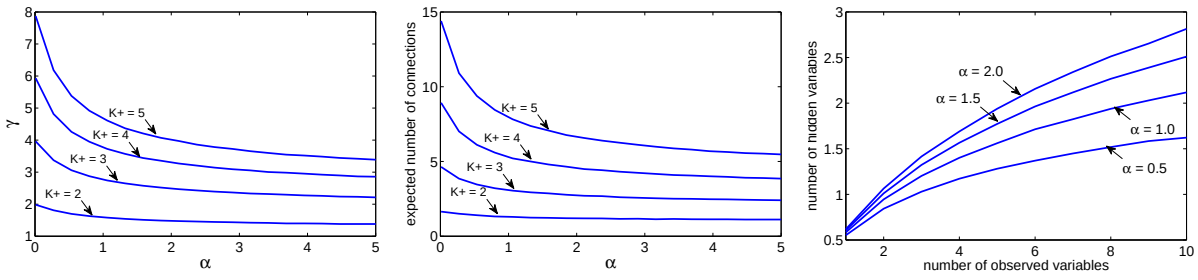
$$a_{death} = \min \left\{ 1, \frac{p(\tilde{Z}'_{A'A'}, Z'_{I'A'}, \tilde{\boldsymbol{\theta}}'_{A'} \mid \alpha, \gamma, \phi, O)}{p(\tilde{Z}_{AA}, Z_{IA}, \tilde{\boldsymbol{\theta}}_A \mid \alpha, \gamma, \phi, O)} \cdot \frac{K_i^*}{(\tilde{\theta}_{j+1} - \tilde{\theta}_j)(K^+ - 1)n_i} \right\}. \tag{15}$$

If accepted, node k is removed from the active subgraph.

Moves on order values: We re-sample the order value of randomly picked node i. This operation is done by finding the lowest order valued parent of i along with its highest order valued children, which we respectively denote l and h. Next, the candidate order value is sampled according to $\theta_i \sim \mathcal{U}(\theta_l, \theta_h)$ and accepted with Metropolis-Hasting acceptance ratio:

$$a_{order} = \min \left\{ 1, \frac{p(\tilde{Z}_{AA}, Z_{IA}, \tilde{\boldsymbol{\theta}}'_A \mid \alpha, \gamma, \phi, O)}{p(\tilde{Z}_{AA}, Z_{IA}, \tilde{\boldsymbol{\theta}}_A \mid \alpha, \gamma, \phi, O)} \right\}, \tag{16}$$

which proposes a new total order $\boldsymbol{\theta}$ respecting the partial order imposed by the rest of the DAG.



(a) (b) (c)

Figure 3: Empirical study of hyperpameters. Figure (a) shows the expected number of active nodes as a function of α and γ. Figure (b) shows that once we know the expected K^+ from α and γ, we can find the expected number of connections. Figure (c) shows the influence of α (with $\gamma = 1$) on the complexity (number of hidden nodes) as function of the number of observable nodes.

4 EXPERIMENTS

The ICP distribution (9) can be used as a prior to learn the structure of any DAG-based model involving hidden units. In particular, one can introduce *a priori* knowledge about the structure by fixing the order values of some observed units. Feedforward neural networks, for instance, can be modelled by imposing $\theta_k = 1$ for all input units and $\theta_k = 0$ for the output units. On the other hand, generative models can be designed by placing all observed units at $\theta_k = 0$, preventing interconnections between them and forcing the above generative units to explain the data. In Section 4.1, we use the ICP as a prior to learn the structures of a generative neural network by approximating the full posterior for 9 datasets. In Section 4.2, we use the ICP to learn the structure of a convolutional neural network (CNN) in a Bayesian learning framework.

4.1 Bayesian nonparametric generative sigmoid network

The network used in this section is the Nonlinear Gaussian Belief Network (NLGBN) [22], which is basically a generative sigmoid network. In this model, the output of a unit u_i depends on a weighted sum of its parents, where W_{ki} represents the weight of parent unit u_k, Z_{ki} indicates whether u_k is a parent of u_i and b_i is a bias. The weighted sum is then corrupted by a zero mean Gaussian noise of precision ρ_i, so that $a_i \sim \mathcal{N}(b_i + \sum_k Z_{ki} W_{ki} u_k, 1/\rho_i)$. The noisy preactivation a_i is then passed through a sigmoid nonlinearity, producing the output value u_i. It turns out that the density function of this random output u_i can be represented in closed-form, a property used to form the likelihood function given the data. An ICP prior is placed on the structure represented by Z along with priors $\gamma \sim \text{Gamma}(0.5, 0.5)$, $1/\alpha \sim \text{Gamma}(0.5, 0.5)$ and $\phi \sim \text{Gamma}(0.5, 0.5)$. To complete the prior on parameters, we specify $\rho_k \sim \text{Gamma}(0.5, 0.5)$, $b_k \sim \mathcal{N}(0, 1)$ and $W_{ki} \sim \mathcal{N}(0, 1)$.

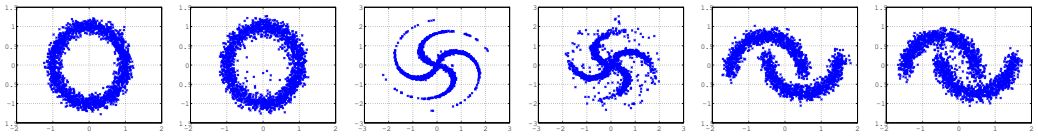
The inference is done with MCMC where structure operators are given in Section 3 and we refer to [12] for the parameter and activation operators. The Markov chain explores the space of structures by creating and destroying edge and nodes, which means that posterior samples are of varying size and shape, while remaining infinitely layered due to $\theta_k \in [0, 1]$. We also simulate the random activations u_k and add them into the chain state.

This experiment aims at reproducing the generative process of synthetic data sources. In the learning phase, we simulate the posterior distribution conditioned on 2000 training points. Fantasy data from the posterior are generated by first sampling a model from set of posterior network samples and then one point is generated from the selected model. Figure 4 shows 2000 test samples from the true distribution along with the samples generated from the posterior accounting for the model uncertainty.

Next, we compare the ICP (with observables at $\theta_k = 0$) against other Bayesian nonparametric approaches: The Cascading Indian Buffet Process [12] and the Extended CIBP [13]. The inference for these models was done with an MCMC algorithm similar to the one used for the ICP and we used similar priors for the parameters to ensure a fair comparison. The comparison metric used in this experiments is the Hellinger distance (HD), a function quantifying the similarity between two probability densities. Table 1 shows the HDs between the generated fantasy datasets and the ground truth datasets.

4.2 Bayesian nonparametric convolutional neural networks

So far, we introduced the ICP as a prior on the space of directed acyclic graphs. In this section we will use



(a) groundtruth (b) fantasy (c) groundtruth (d) fantasy (e) groundtruth (f) fantasy

Figure 4: Resulting fantasy data generated from the posterior on 3 toy datasets.

Table 1: Hellinger distance between the fantasy data from posterior models and the test set. Dimensionality of the data is given in parentheses. The baseline shows the distance between the training and test sets, representing the best achievable distance since the two come from the true source.

DATA SET	RING (2)	TWO MOONS (2)	PINWHEEL (2)	GEYSER (2)	IRIS (4)	YEAST (8)	ABALONE (9)	CLOUD (10)	WINE (12)
ICP	**0.0402**	**0.0342**	**0.0547**	**0.0734**	0.2666	**0.3817**	**0.1379**	**0.1495**	**0.3629**
CIBP	0.0493	0.0469	0.0692	0.1246	0.2667	0.4056	0.1502	0.1713	0.4079
ECIBP	0.0419	0.0450	0.0685	0.1171	**0.2632**	0.3840	0.1470	0.1501	0.3855
BASELINE	0.0312	0.0138	0.0436	0.0234	0.1930	0.3059	0.1079	0.1299	0.3387

this formalism in order to construct a prior on the space of convolutional neural architectures. The fundamental building blocks of (2D) convolutional networks are tensors T whose entries encode the presence of local features in the input image. A convolutional neural network can be described as a sequence of convolution operators acting on these tensors followed by entry-wise nonlinearity f.

In our nonparametric model, a convolutional network is constructed from a directed acyclic graph. Each node of the graph represents a tensor $T^{(i)}$. The entries of this tensor are given by

$$T^{(i)} = \text{ReLU}\left(\sum_{k \in \text{Parents}(i)} W^{(ki)} \star T^{(k)}\right) , \qquad (17)$$

where $W^{(ki)}$ is a tensor of convolutional weights and $\star$ is the discrete convolution operator. In most hand-crafted architectures, the spatial dimensions of the tensor are course-grained as the depth increases while the number of channels (each representing a local feature of the input) increases. In the ICP, the depth of a node i is represented by its reputation θ_i. In order to encode the change of shape in the nonparametric prior, we set the number of channels to be a function of θ:

$$N_c(\theta) = 2^{\lfloor N_{\text{bins}}(1-\theta))\rfloor} + N_0 , \qquad (18)$$

where N_{bins} is the number of different possible tensor shapes and N_0 is the number of channels of the lowest layers. Similarly, the number of pixels is given by:

$$N_p(\theta) = 2^{-\lfloor N_{\text{bins}}(1-\theta))\rfloor} M , \qquad (19)$$

where M is the number of pixels in the original image. The shape of the weight tensors $W^{(ki)}$ is determined by the shape of parent and child tensor.

In a classification problem, the nonparametric convolutional network is connected to the data through two observed nodes. The input node X stores the input images and we set $\theta_X = 1$. On the other hand, for the output node we set $\theta_Y = 0$, and have it receive input through fully connected layers:

$$Y = \text{Softmax}\left(\sum_{k \in \text{Parents}(Y)} \left(\sum_{a,b,c} V_a^{(k)} T_{abc}^{(k)}\right)\right) , \qquad (20)$$

where $V^{(k)}$ is a tensor of weights.

Note that, when computing the acceptance ratios in Eqs. (14-16), we now need to add the model evidence $\log p(y \mid G, X)$ for the proposal graph and current graph to the numerators and denominators, respectively. In this paper, we use a point estimate of the log model evidence:

$$p(y \mid G, x) \approx p(y \mid G, \{\hat{W}^{(ki)}\}, x) , \qquad (21)$$

where $\hat{W}^{(ki)}$ are the parameters of the network optimized using Adam (α=0.1, β_1=0.9, β_2=0.999, *eps*=1e-08, η=1.0) [23].

We performed architecture sampling on the MNIST dataset. For computational reasons, we restricted the dataset to the first three classes (1, 2 and 3). We sampled the DAGs using the MCMC sampler introduced in Section 3 with prior parameters $\alpha = 1$, $\gamma = 20$ and $\phi = 5$. For each sampled DAG, we trained the induced convolutional architecture until convergence (540 iterations, batch size equal to 100). The number of bins in the partition of the range of the reputations was five and the number of channels of the first convolutional layer was four. We ran 15 independent Markov chains in order to sample from multiple modes. Each chain consisted of 300 ac-

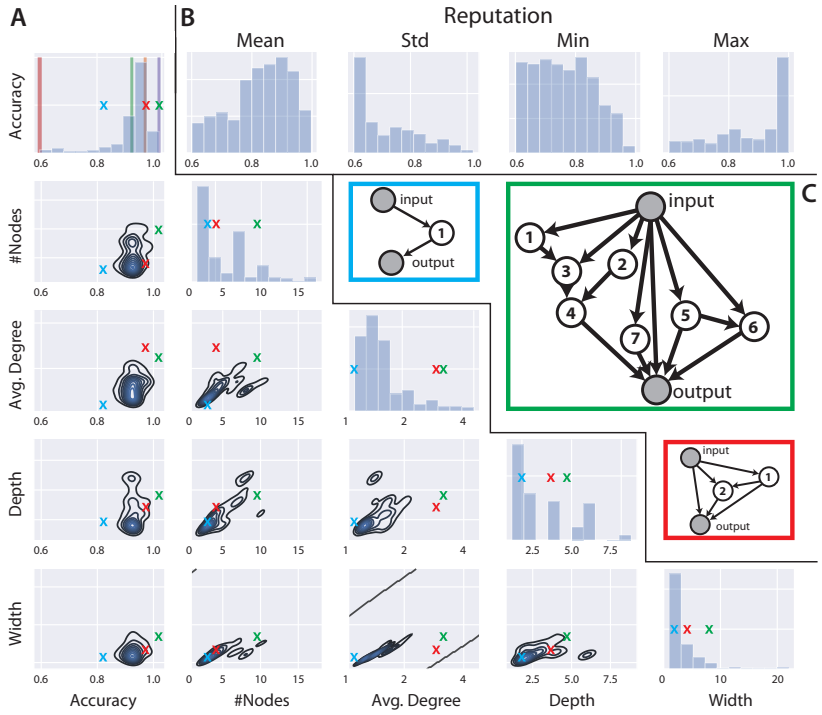


Figure 5: Statistics of the sampled convolutional architectures. A) Histograms and bivariate density plots of test set accuracy, number of nodes, average degree width and depth. The three colored crosses denote the statistics of the three visualized networks. B) Histogram of the mean, standard deviation, min and max of the popularity values. C) Examples of visited architectures visited during the inference.

cepted samples. After sampling, all chains were merged, resulting in a total of 4500 sampled architectures[1].

Figure 5A shows accuracy and descriptive statistics of the sampled convolutional architectures. In all these statistics, we only considered nodes that receive input from the input node (directly or indirectly) as the remaining nodes do not contribute to the forward pass of the network. The network *width* is quantified as the total number of directed paths between input and output nodes, while *depth* is quantified as the maximal directed path length. The sampler reaches a wide range of different architectures, whose number of layers range from three to fifteen, and whose average degree range from one to four. Some examples of architectures are shown in Figure 5C. Interestingly, the correlation between the number of nodes, degree, width and depth and accuracy is very low. Most likely, this is due to the simple nature of the MNIST task. The ensemble accuracy (0.95), obtained by averaging the label probabilities over all samples, is higher than the average accuracy (0.91), but lower than the maximum accuracy (0.99). Figure 5B shows the histograms of mean, standard deviation, minimum and maximum of the reputation values in the networks.

[1]The code is available at https://github.com/mhinne/NPDAG

5 CONCLUSION AND FUTURE WORK

This paper introduced the Indian chefs process (ICP) as a Bayesian nonparametric distribution on the joint space of infinite directed acyclic graphs and orders. The model allows for a novel way of learning the structure of deep learning models. As a proof of concept, we have demonstrated how the ICP can be used to learn the architecture of convolutional deep networks trained on the MNIST data set. However, for more realistic applications, several efficiency improvements are required. First, the inference procedure over the model parameters could be performed using Hamiltonian Monte Carlo. This would remove the need to fully train the network for every sampled DAG. Second, add deep learning-specific sampling moves. For example, add an ”increase depth” move that replaces a connection with a path comprised by two connections and a latent node. And third, extend ICP beyond deep learning architectures. For example, the ICP may serve as a basis for nonparametric causal inference, where a DAG structure is learned when the exact number of relevant variables is not known a priori, or when certain relevant input variables are not observed [24].

References

[1] M. Schmidt, A. Niculescu-Mizil, K. Murphy, *et al.*, "Learning graphical model structure using L1-regularization paths," in *AAAI*, vol. 7, pp. 1278–1283, 2007.

[2] S. Banerjee and S. Ghosal, "Bayesian structure learning in graphical models," *Journal of Multivariate Analysis*, vol. 136, pp. 147–162, 2015.

[3] V. Mansinghka, C. Kemp, T. Griffiths, and J. Tenenbaum, "Structured priors for structure learning," *arXiv preprint arXiv:1206.6852*, 2012.

[4] A. Mohammadi, E. C. Wit, *et al.*, "Bayesian structure learning in sparse Gaussian graphical models," *Bayesian Analysis*, vol. 10, no. 1, pp. 109–138, 2015.

[5] D. G. R. Tervo, J. B. Tenenbaum, and S. J. Gershman, "Toward the neural implementation of structure learning," *Current Opinion in Neurobiology*, vol. 37, pp. 99–105, 2016.

[6] N. Hjort, C. Holmes, P. Muller, and S. Walker, "An invitation to Bayesian nonparametrics," *Bayesian Nonparametrics*, vol. 28, p. 1, 2009.

[7] Y. Jiang and A. Saxena, "Infinite latent conditional random fields," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 262–266, 2013.

[8] K. Ickstadt, B. Bornkamp, M. Grzegorczyk, J. Wieczorek, M. R. Sheriff, H. E. Grecco, and E. Zamir, *Nonparametric Bayesian Networks*. Oxford University Press, 2010.

[9] S. P. Chatzis and G. Tsechpenakis, "The infinite hidden Markov random field model," *IEEE Transactions on Neural Networks*, vol. 21, no. 6, pp. 1004–1014, 2010.

[10] F. Doshi, D. Wingate, J. Tenenbaum, and N. Roy, "Infinite dynamic Bayesian networks," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 913–920, 2011.

[11] I. Valera, F. Ruiz, L. Svensson, and F. Perez-Cruz, "Infinite factorial dynamical model," in *Advances in Neural Information Processing Systems*, pp. 1657–1665, 2015.

[12] R. P. Adams, H. M. Wallach, and Z. Ghahramani, "Learning the structure of deep sparse graphical models," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010.

[13] P. Dallaire, P. Giguere, and B. Chaib-draa, "Learning the structure of probabilistic graphical models with an extended cascading Indian buffet process," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[14] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Advances in Neural Information Processing Systems*, pp. 2074–2082, 2016.

[15] R. Y. Rohekar, S. Nisimov, Y. Gurwicz, G. Koren, and G. Novik, "Constructing deep neural networks by Bayesian network structure learning," in *Advances in Neural Information Processing Systems*, pp. 3047–3058, 2018.

[16] T. L. Griffiths and Z. Ghahramani, "The Indian buffet process: An introduction and review," *The Journal of Machine Learning Research*, vol. 12, pp. 1185–1224, 2011.

[17] J. Paisley, *Machine Learning with Dirichlet and Beta Process Priors: Theory and Applications*. PhD thesis, Duke University, Durham, North Carolina, 2010.

[18] P. Orbanz and Y. W. Teh, "Bayesian nonparametric models," in *Encyclopedia of Machine Learning*, pp. 81–89, Springer, 2010.

[19] R. Thibaux and M. I. Jordan, "Hierarchical beta processes and the Indian buffet process," in *International conference on artificial intelligence and statistics*, 2007.

[20] J. Paisley, A. Zaas, C. Woods, G. Ginsburg, and L. Carin, "A stick-breaking construction of the beta process," in *Proceedings of the International Conference on Machine learning*, 2010.

[21] P. J. Green and D. I. Hastie, "Reversible jump MCMC," *Genetics*, vol. 155, no. 3, pp. 1391–1403, 2009.

[22] B. J. Frey, "Continuous sigmoidal belief networks trained using slice sampling," in *Advances in Neural Information Processing Systems 9*, pp. 452–458, MIT Press, 1997.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] K. Mohan, J. Pearl, and J. Tian, "Graphical models for inference with missing data," in *Advances in Neural Information Processing Systems 26* (Burges, Bottou, Welling, Ghahramani, and Weinberger, eds.), pp. 1277–1285, Curran Associates, Inc., 2013.