Bayesian Nonparametric Deep Learning

Patrick Dallaire
Department of Computer Science
Laval University, Québec, Canada
patrick.dallaire@ift.ulaval.ca

Ludovic Trottier
Department of Computer Science
Laval University, Québec, Canada
ludovic.trottier.1@ulaval.ca

Philippe Giguère
Department of Computer Science
Laval University, Québec, Canada
philippe.giguere@ift.ulaval.ca

Brahim Chaib-draa
Department of Computer Science
Laval University, Québec, Canada
chaib@ift.ulaval.ca

François Laviolette
Department of Computer Science
Laval University, Québec, Canada
francois.laviolette@ift.ulaval.ca

Abstract

This paper presents a Bayesian nonparametric formulation of deep learning that is based on the Indian Chefs Process (ICP), a Bayesian nonparametric prior on the joint space of infinite directed acyclic graphs (DAGs) and orders. The distribution relies on a latent Beta process controlling both the orders and outgoing connection probabilities of the units, making the graph sparse and infinite. In the experiments, we demonstrate the usefulness of the ICP on learning deep generative models.

1 Formalizing the Bayesian Nonparametric Deep Generative Model

We consider a layerless formulation of neural networks where connections are not constrained by layers and units can connect to any units below them with some probability. In particular, we use the Nonlinear Gaussian Belief Network (NLGBN) as our generative model [4]. 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. Afterwards, this sum is corrupted by zero mean Gaussian noise with 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 function, producing the output value of unit u_i.

1.1 Probability Model on Finite DAGs

We define $G = (V, Z)$ as the DAG structure of a neural net where $V = \{1, \ldots, K\}$ is the set of units and Z is the $K \times K$ adjacency matrix of connections. We define an ordering $\boldsymbol{\theta}$ on the units so that the direction of a connection is determined by comparing the order value of each unit. One can see θ as a continuous layer index, making the structure infinitely layered. Thus, we impose the constraint that a connection Z_{ki} is only allowed between units when $\theta_k > \theta_i$.

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

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

and an order value θ_k to every node k in G based on the following probability 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 [10], where Eq. (1) becomes the *base distribution*, and is convenient when evaluating the limit of section 1.2. 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:

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

The adjacency matrix Z may contain connections for nodes that are not of interest. As an example, when learning a neural network with hidden neurons, we are only interested in the observable neurons and their ancestors. Formally, we define $A \subseteq V$ as the set of *active* nodes, containing 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:

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

where $m_k = \sum_{i \in A} Z_{ki}$ denotes the number of outgoing connections from node k to any active nodes, $\downarrow_k = \sum_{j \in A} \mathbb{I}(\theta_j < \theta_k)$ denotes the number of active nodes having an order value strictly lower than θ_k and $\downarrow = \{\downarrow_k\}_{k=1}^K$. 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 distributions, leading to the following equation:

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

where $x^{\overline{n}} = x(x+1)\ldots(x+n-1)$ is the Pochhammer symbol denoting the rising factorial and $H = A \setminus O$ is the set of active hidden nodes.

The set of active node A contains all observable nodes as well as their ancestors, which means it is disconnected from the other parts of the graph G. 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. The joint probability of these submatrices is:

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

where the number of negative Bernoulli trials $\downarrow_k$ 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 $\boldsymbol{\theta}_A^{\nearrow}$ and augment it with extrema $\theta_0^{\nearrow} = 0$ and $\theta_{K^+ + 1}^{\nearrow} = 1$. 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 $\theta^{\nearrow}_j$. Based on the previous notation, the probability for an inactive nodes to lie between two active nodes is simply $\theta^{\nearrow}_{j+1} - \theta^{\nearrow}_j$. Using this notation, we have the following marginal probability:

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

where we introduce $K^+ = |A|$ to denote the number of active nodes, $K^- = |I|$ to denote the number of inactive nodes and $x^{\underline{n}} = x(x-1)\dots(x-n+1)$ symbolizes the falling factorial. 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. One way to simplify this is to reorder the adjacency matrix according to $\boldsymbol{\theta}^{\nearrow}_A$, which we denote $Z^{\nearrow}_{AA}$. 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 [7]. The number of permutation mapping to this sorted representation is accounted for by the normalization constant $\frac{(K-D)^{\underline{K^+-D}}}{K^+!}$.

1.2 From Finite to Infinite DAGs

An elegant way to construct Bayesian nonparametric models is to consider the infinite limit of finite parametric Bayesian models [9]. Following this idea, we revisit the model of section 1.1 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:

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

where ψ is the digamma function. Eq. (9) is the proposed marginal probability distribution on the joint space of infinite DAGs and continuous orders, which allows to define infinite dimensional and arbitrarily deep networks.

1.3 The Indian Chefs Process

Sampling random DAGs from probability distribution (9) can be done with the Indian chefs process (ICP). 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) star chefs (corresponding to observable nodes) which are introduced iteratively and 2) regular chefs (corresponding to hidden nodes) 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 on 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 $\downarrow_i$ chefs having a lower reputation than chef i. To this end, we can first sample the total number of inspiration connections with:

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

and then uniformly pick one of the $\binom{\downarrow_i}{q_i}$ 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:

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

where $\downarrow_k$ 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:

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

where the new regular chefs are independently assigned a random reputation drawn from $\mathcal{U}(\theta^{\nearrow}_j, \theta^{\nearrow}_{j+1})$. 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.

1.4 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 $\theta = 0$ and all hidden nodes are set to $\theta = 1$. This way, connections are prohibited between hidden nodes and between observable nodes, while hidden-to-observable connections are still permitted.

2 Markov Chain Monte Carlo Inference for the Indian Chefs Process

We propose a reversible jump MCMC algorithm producing random walks on Eq. (9) [6]. 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. The set of potential parents 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 potential parent k, we Gibbs sample the connections using the following conditional probability:

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

where $m_k^{\neg i}$ is the number of outgoing connections of node k excluding connections going to node i and $Z^{\nearrow^{\neg ki}}_{AA}$ has element ki removed. Also, all connections not respecting the order are prohibited and therefore have an occurrence probability of 0.

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

In the birth case, we activate node k by connecting it to node i. Its order θ_k is determined by uniformly selecting an insertion interval above θ_i. Assuming node i is also the i^{th} element in $\boldsymbol{\theta}_A^{\nearrow}$, we have $\uparrow_i = K^+ - i + 1$ 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 $\theta_k \sim \mathcal{U}(\theta_j^{\nearrow}, \theta_{j+1}^{\nearrow})$. The Metropolis-Hastings acceptance ratio for this move is:

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

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

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

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

Moves on order values: We resample 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 ratio:

$$a_{order} = \min\left\{1, \frac{p(Z^{\nearrow}_{AA}, Z_{IA}, \boldsymbol{\theta}'^{\nearrow}_{A}|\alpha,\gamma,\phi,O)}{p(Z^{\nearrow}_{AA}, Z_{IA}, \boldsymbol{\theta}^{\nearrow}_{A}|\alpha,\gamma,\phi,O)}\right\}. \tag{16}$$

This operation proposes a new total order $\boldsymbol{\theta}$ respecting the partial order imposed by the rest of the current directed acyclic graph structure.

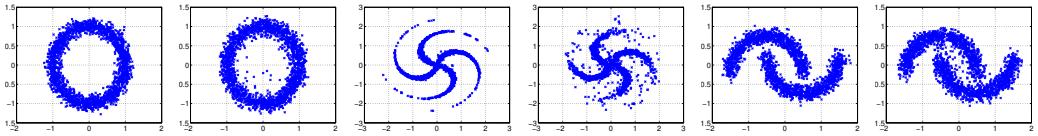
3 Experiments on Density Estimation

The ICP prior in equation (9) can be used to learn the structure of deep DAG-based models. One can force specific structures by fixing the order values of some observed units. Feedforwards neural nets for instance can be modelled by fixing $\theta_k = 1$ for all input units and $\theta_k = 0$ for the output units. Fully generative models can be designed by fixing all observed units to $\theta_k = 0$, preventing interconnections between them and forcing the above generative units to explain the data. The present experiments are based on this last specification of the ICP prior over structures.

To complete the prior, we specify $\rho_k \sim \text{Gamma}(0.5, 0.5)$, $b_k \sim \mathcal{N}(0,1)$, $W_{ki} \sim \mathcal{N}(0,1)$, $\gamma \sim \text{Gamma}(0.5, 0.5)$, $1/\alpha \sim \text{Gamma}(0.5, 0.5)$ and $\phi \sim \text{Gamma}(0.5, 0.5)$. It turns out that the density function of the NLGBN random output can be represented in closed-form, a property used to form the likelihood function given the data. The inference is done through Gibbs sampling, Metropolis-Hastings and reversible jump Markov Chain Monte Carlo. The Markov chain explores the space of structures by creating and the killing units, which means that posterior samples are of varying size and shape, while remaining infinitely layered due to $\theta_k \in [0, 1]$. We also add the random activations u_k into the chain.

The following density estimation experiments aims at reproducing the generative process of a data source with a marginalized network using Bayesian model averaging. In practice, this is done by generating a fantasy data set from the marginalized network and comparing the result with a test set. More precisely, generating the fantasy dataset is done by first sampling several random posterior networks and then sampling a unique data point from each network.

The comparison metric used in the experiments is the Hellinger distance (HD), a function quantifying the similarity between two probability densities. It is symmetric, returns 0 for identical measures and



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

Figure 1: Fantasy datasets produced from the ICP posterior.

Table 1: Estimated Hellinger distance between fantasy datasets generated by the learned models and the test set. The baseline row shows the Hellinger distance between the training and test sets, representing the best achievable Hellinger distance.

	Baseline	ICP	CIBP	ECIBP
Ring (2)	0.0312	**0.0402**	0.0493	0.0419
Two Moons (2)	0.0138	**0.0342**	0.0469	0.0450
Pinwheel (2)	0.0436	**0.0547**	0.0692	0.0685
Geyser (2)	0.0234	**0.0734**	0.1246	0.1171
Iris (4)	0.1930	0.2666	0.2667	**0.2662**
Yeast (8)	0.3059	**0.3817**	0.4056	0.3840
Abalone (9)	0.1079	**0.1379**	0.1502	0.1470
Cloud (10)	0.1299	**0.1495**	0.1713	0.1501
Wine (12)	0.3387	**0.3629**	0.4079	0.3855

1 for complete dissimilarity. When dealing with data samples instead of probability densities, we can only approximate the HD between the two sets of data points. This can be done by using kernel density estimation to create a density function and then use Monte carlo sampling to compute the Bhattacharyya coefficient [2].

To compare the ICP with other Bayesian nonparametric models, we also evaluated how the cascading Indian buffet process (CIBP) [1] and the extended CIBP [3] perform when learning deep NLGBNs. 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. In Table 1, we can see that models learned with the ICP prior outperformed the ones learned with CIBP and ECIBP most of the time. Moreover, the table includes the baseline distance between the training set and the test set. Since both of them are generated by the true source of data, this measure provides an intuition about the difficulty of capturing the generative model of a particular source. This also gives, for each data set, an idea of what to expect as the best achievable performance in terms of Hellinger distance.

4 Conclusion and Future Work

We have presented a closed-form probability distribution on the joint space of infinite directed acyclic graphs and orders allowing a novel Bayesian nonparametric formulation of deep learning models. We are currently investigating inference methods for the Indian chefs process and exploring potential connections between the dropout approach to Bayesian deep learning [5] and the birth-death process involved in our reversible jump MCMC inference. Improvements on the inference procedure could be made by using stochastic gradient MCMC [8] for the parameters of the model.

References

[1] 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.

[2] E. L. Boone, J. R. Merrick, and M. J. Krachey. A hellinger distance approach to mcmc diagnostics. *Journal of Statistical Computation and Simulation*, 84(4):833–849, 2014.

[3] 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.

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

[5] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[6] P. J. Green and D. I. Hastie. Reversible jump mcmc. *Genetics*, 155(3):1391–1403, 2009.

[7] T. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. *Advances in neural information processing systems*, 18:475, 2006.

[8] Y.-A. Ma, T. Chen, and E. Fox. A complete recipe for stochastic gradient mcmc. In *Advances in Neural Information Processing Systems*, pages 2917–2925, 2015.

[9] P. Orbanz and Y. W. Teh. Bayesian nonparametric models. In *Encyclopedia of Machine Learning*, pages 81–89. Springer, 2010.

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