A Survey on Bayesian Nonparametric Learning

JUNYU XUAN, University of Technology Sydney, Australia
JIE LU, University of Technology Sydney, Australia
GUANGQUAN ZHANG, University of Technology Sydney, Australia

Bayesian (machine) learning has been playing a significant role in machine learning for a long time due to its particular ability to embrace uncertainty, encode prior knowledge, and endow interpretability. On the back of Bayesian learning's great success, Bayesian nonparametric learning (BNL) has emerged as a force for further advances in this field due to its greater modelling flexibility and representation power. Instead of playing with the fixed-dimensional probabilistic distributions of Bayesian learning, BNL creates a new "game" with infinite-dimensional stochastic processes. BNL has long been recognised as a research subject in statistics and, to date, several state-of-the-art pilot studies have demonstrated that BNL has a great deal of potential to solve real-world machine learning tasks. However, despite these promising results, BNL has not created a huge wave in the machine learning community. Esotericism may account for this. The books and surveys on BNL written by statisticians are overcomplicated and filled with tedious theories and proofs. Each is certainly meaningful but may scare away new researchers, especially those with computer science backgrounds. Hence, the aim of this paper is to provide a plain-spoken, yet comprehensive, theoretical survey of BNL in terms that researchers in the machine learning community can understand. It is hoped this survey will serve as a starting point for understanding and exploiting the benefits of BNL in our current scholarly endeavours. To achieve this goal, we have collated the extant studies in this field and aligned them with the steps of a standard BNL procedure - from selecting the appropriate stochastic processes, through manipulation, to executing the model inference algorithms. At each step, past efforts have been thoroughly summarised and discussed. In addition, we have reviewed the common methods for implementing BNL in various machine learning tasks along with its diverse applications in the real-world as examples to motivate future studies.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; *Machine learning*; • **Mathematics of computing** → *Probability and statistics*;

Additional Key Words and Phrases: Data science, machine learning, Bayesian learning

ACM Reference Format:
Junyu Xuan, Jie Lu, and Guangquan Zhang. 2018. A Survey on Bayesian Nonparametric Learning. *ACM Comput. Surv.* 1, 1, Article 1 (January 2018), 37 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The Bayesian paradigm for machine learning, also known as Bayesian (machine) learning, is to apply probability theories and techniques to represent and learn knowledge from data. Bayesian learning has played a significant and irreplaceable role in machine learning area ever since the

Authors' addresses: Junyu Xuan, University of Technology Sydney, Centre for Artificial Intelligence, P.O. Box 123 Broadway, Sydney, Australia, 2007, Junyu.Xuan@uts.edu.au; Jie Lu, University of Technology Sydney, Centre for Artificial Intelligence, P.O. Box 123 Broadway, Sydney, Australia, 2007, Jie.Lu@uts.edu.au; Guangquan Zhang, University of Technology Sydney, Centre for Artificial Intelligence, P.O. Box 123 Broadway, Sydney, Australia, 2007, Guangquan.Zhang@uts.edu.au.

0360-0300/2018/1-ART1 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

pioneering works of Professor Judea Pearl. Some renowned examples are Bayesian network, Gaussian mixture models, hidden Markov model [134], Markov random field, conditional random field, and latent Dirichlet allocation [11]. Compared to other learning paradigms, Bayesian learning has distinctive advantages. First, it embraces the uncertainty by explicitly representing, manipulating, and mitigating it based on a solid theoretical foundation - probability, which makes Bayesian learning more robust facing real-world intelligent systems [96]. Second, Bayesian learning can smoothly encode the prior knowledge about a problem under study in parallel with the knowledge learned from observed data - an asset that is also useful for overcoming the overfitting issues that tend to arise with limited amounts of data because priors can be treated as regularisers of the data. Lastly, Bayesian learning has inherent interpretability thanks to its clear and meaningful probabilistic structure. Beyond its abilities to make predictions, these interpretability becomes an additional desired learning outcome. In a standard Bayesian learning procedure, the goal is to build a model composed of a set of parameters (or random variables) to suit the target problem and then infer the posterior of these parameters given observed data. The model's parameters, which are normally defined to satisfy known fixed-dimensional probabilistic distributions (e.g., Gaussian, Dirichlet, gamma, and multinomial distributions), control the data's goodness-of-fit and the model's complexity. Thus, inappropriate model parameters may lead to underfitting (i.e., bad goodness-of-fit but good model complexity) or overfitting (i.e., good goodness-of-fit but bad model complexity) issues. Since the building blocks of Bayesian learning are fixed-dimensional probabilistic distributions, the number of parameters must be finite; hence the name, Bayesian parametric learning. Determining or learning the number of these parameters are based on human labour or restarting the algorithm several times to find the optimal settings - a process that is time-consuming and not scalable to large-scale unfamiliar data.

Bayesian nonparametric learning (BNL) advances Bayesian learning in terms of the representation power and modelling flexibility. First, it is necessary to clarify that *nonparametric* does not mean "there are no parameters". In fact, quite the opposite is true. In theory, there are an infinite number of parameters in Bayesian nonparametric models. Therefore, *nonparametric* means "there is no need to predefine the dimensionality for the parameters". Studies on Bayesian nonparametrics began with two papers written by Professor Thomas S. Ferguson in 1973 [57] and Professor Kjell Doksum in 1974 [43]. However, the nonparametric paradigm did not attract much attention from computer scientists until 2005 when a conference paper titled "Sharing clusters among related groups: Hierarchical Dirichlet processes" applied Bayesian nonparametrics to machine learning [166]. Due to the great success of this work, computer scientists began to pay attention to Bayesian nonparametrics, giving rise to BNL. From then on, BNL became an interdisciplinary subject for statisticians and computer scientists. Rather than playing with fixed-dimensional probabilistic distributions, BNL is a "game" to play with infinite-dimensional stochastic processes (e.g., Dirichlet, Gaussian, Poisson, gamma, and negative binomial processes). The benefits are two-fold. Stochastic processes are not restricted by predetermined underlying assumptions about the dimensionality of the data, which provides a far greater ability to represent the data accurately. Further, BNL builds probabilistic models with flexible structures that can autonomically adapt to new incoming observations (also known as "letting the data speak"). A typical scenario is document modelling. Here, BNL can not only learn the summarised topics in a set of documents but can also adapt the number of learned topics according to the documents in the set. Further, when new documents arrive, BNL can change the number and content of the topics effortlessly. However, there is no such thing as a free lunch. While BNL does bring powerful representations and highly flexible models to the learning table, model inference still faces great challenges. Fortunately, statistical inference techniques have simultaneously experienced a similar pace of advancement as BNL has progressed. These advancements include sampling-based inference algorithms (e.g., slice sampling and Hamiltonian Monte Carlo), optimisation-based inference algorithms (e.g., variational inference), and hybrid inference algorithms (e.g., stochastic gradient Markov chain Monte Carlo). Together with the growing computational powers of "hardware" (e.g., multi-core CPUs, GPUs, and distributed computing platforms), these efficient inference algorithms make applying BNL on large-scale real-world tasks possible.

Only a small number of surveys and books in the literature focus on BNL and most were written by statisticians, Among these, the pioneer review [58] and three books [74, 88, 123] stand out. While thorough, these works each express a perspective on BNL from a statistician's point of view. Applications for BNL are based on statistical scenarios and supported by deep and detailed theoretical analyses and property proofs in those contexts. Although profound and highly necessary for developing new models or investigating consistency and asymptotic behaviours, all are difficult to understand for those with a computer science background. Further, they lack a roadmap to applying these techniques to machine learning tasks or real-world data-driven scenarios. Outside of statistics, there are also two general introductions to BNL [70, 96] and one survey specifically related to non-exchangeable priors [60]. However, all either present very early research progress or focus on one aspect of BNL. A substantial amount of progress has been made in this field over the past ten years, which justifies a comprehensive and updated review.

This survey aims to provide a good starting point for researchers who are interested in BNL - primarily those in the computer science community. To achieve this goal, we have organised the work in this field to align with the standard BNL procedure. That is, a) select appropriate stochastic processes, b) manipulate those processes, and c) execute the model inference. In presenting the stochastic processes, we have mainly focused on the ability of each to model different kinds of data, rather than on detailed theoretical definitions and proofs. These definitions and proofs have been purposely omitted with corresponding references for further reading. Our purpose is to highlight the potential pathways for studying BNL so researchers can choose their own points of interest as a launch pad for further explorations. For example, developing a new stochastic process for a specific data structure or designing a more efficient inference algorithm. The Bayesian nonparametric extensions of current machine learning algorithms or models have been reviewed as motivating examples for researchers who already have knowledge in machine learning. The goal here is to explain BNL's merits for consideration to those who intend to extend other algorithms or models. Additionally, we have presented a selection of real-world applications across a diverse range of domains to show the practical value in BNL as encouragement to applied researchers to consider BNL as an analytical tool in their future studies.

The remainder of this work is organised as follows. The definitions of BNL are introduced in Section 2. Section 3 presents the basic ingredients (i.e., various stochastic processes) of BNL followed by their manipulations summarised in Section 4. Section 5 reviews the model inference techniques used in this community. Sections 6 and 7 discuss the use of BNL in machine learning tasks and real-world applications. Section 8 concludes this survey and provides our visions for the future.

2 DEFINITIONS

One closely related definition of Bayesian nonparametrics is given by statisticians as

Definition 1 (**Bayesian nonparametrics** [88]). *"Bayesian nonparametrics are models and methods characterised by (a) big parameter spaces (unknown density and regression functions, link and response functions, etc.) and (b) construction of probability measures over these spaces."*

This definition is used to distinguish four important concepts in statistics: frequentist parametrics, Bayesian parametrics, frequentist nonparametrics, and Bayesian nonparametrics. Another simpler definition of Bayesian nonparametric models comes from the *Encyclopedia of Machine Learning* as

Definition 2 (**Bayesian nonparametric model** [128]). *"A Bayesian nonparametric model (BNP) is a Bayesian model on an infinite-dimensional parameter space. The parameter space is typically chosen as the set of all possible solutions for a given learning problem."*

Definition 2 was formulated for researchers in the machine learning field, so it concentrates more on the infinite-dimensional characteristic and the learning ability of BNP. Additionally, the ability to build models on infinite-dimensional parameter spaces is an important distinction in this definition, as it implicates stochastic processes as an alternative to probabilistic distributions. Compared to traditional and naive applications of stochastic processes in time series events modelling, one major contribution of Bayesian nonparametric learning to the machine learning community is to introduce stochastic processes for two (or higher)-dimensional space partition or more complex data structures. BNL's strengths from a computer science perspective are its flexible data structures and manipulations. Therefore, we argue it would be better to define BNL according to those strengths to promote a better understanding of BNL in computer scientists and enhance their willingness to adopt BNL in their studies. An alternative definition for BNL follows.

Definition 3 (**Bayesian nonparametric learning**). *Bayesian nonparametric learning (BNL) is to build and inference the probabilistic models for specific learning tasks based on stochastic processes and their manipulations.*

In the above definition, the core of BNL is to define and manipulate stochastic processes according to the target task. As an analogy, building a Bayesian nonparametric model for a specific learning task is just like using Lego bricks to build a robot, where the stochastic processes are bricks of different shapes, and the manipulation is like assembling the bricks to form more complex objects, e.g., a robot. This definition vividly demonstrates the core of BNL and the standard procedure for building a model. In the next section, we review the different bricks, i.e., stochastic processes, currently used in this field, followed by the different manipulation techniques.

3 BASIC INGREDIENTS: STOCHASTIC PROCESSES

3.1 Definition and property

In probability theory, a stochastic process is a set of (indexed) random variables, where indexes are derived from an (countably infinite) index/parameter space and the variables are derived from a state space. Statistics defines a number of different stochastic processes according to the different properties of these random variables, which we have organised some popular ones in BNL into a relational diagram in Figure 1. Not all of these processes have been used in machine learning field (yet), but the most popular and representative ones are reviewed below.

— *Gaussian Process (GP).* A GP [151] is a specific set of random variables with a property that any finite subset of these variables satisfies a Gaussian distribution. A GP prior is represented as

$$g(x) \sim \mathrm{GP}(f_m(x), k_c(x, x')) \tag{1}$$

where $g(x)$ is a realisation of GP, $f_m(x) = \mathbb{E}[g(x)]$ is the mean function, and $k_c(x, x') = \mathbb{E}[(g(x) - f_m(x))(g(x') - f_m(x'))]$ is the covariance function.

When used in machine learning, $g(x)$ models an underlying function, e.g., the mapping relationships between the features and labels of data. Hence, the GP is a good prior for latent functions and is especially useful for situations when there no knowledge of the functional form exists. It has been proven that GP has a deep relationship with traditional algorithms in machine learning, e.g., support vector machines [178].

One large group of popular stochastic processes in BNL falls within the scope of Lévy processes [148], which are (informally) defined as the stochastic processes with stationary and independent

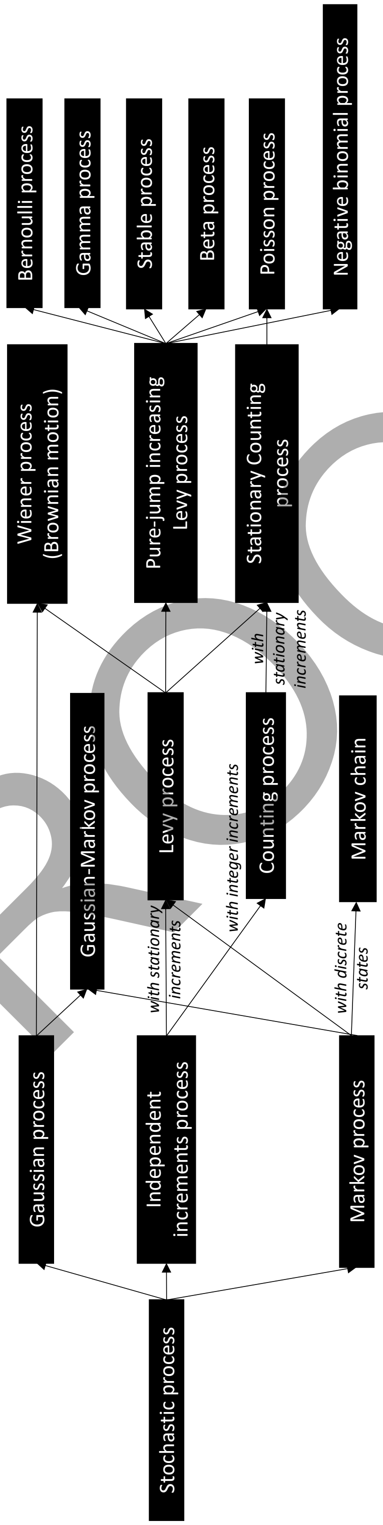

Fig. 1. Various stochastic processes in literature.

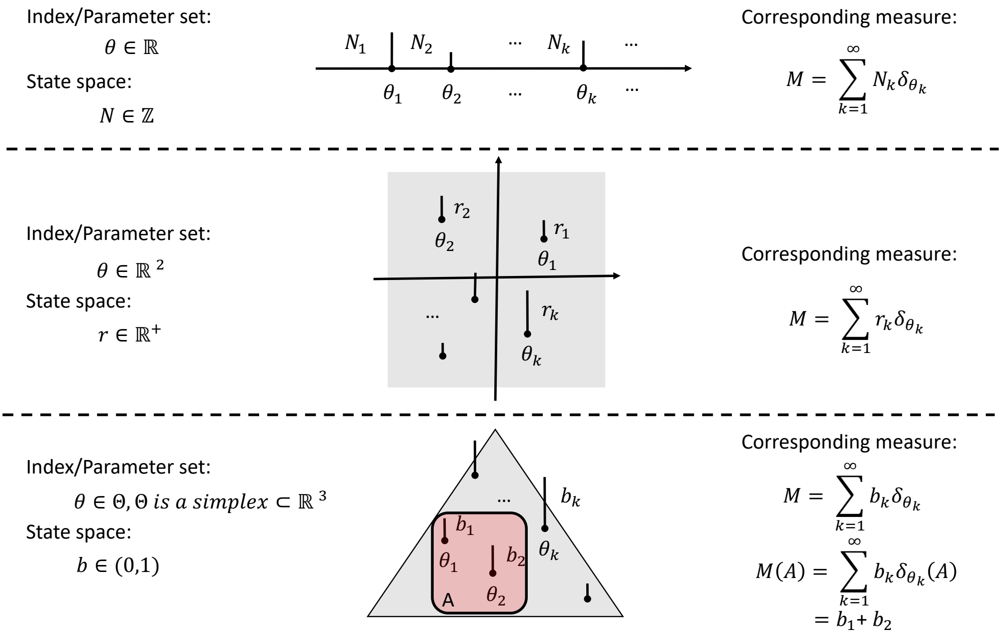
Index/Parameter set:
$\theta \in \mathbb{R}$
State space:
$N \in \mathbb{Z}$
N_1 N_2 ... N_k ...
θ_1 θ_2 ... θ_k ...
Corresponding measure:
$M = \sum_{k=1}^{\infty} N_k \delta_{\theta_k}$
Index/Parameter set:
$\theta \in \mathbb{R}^2$
State space:
$r \in \mathbb{R}^+$
r_2 θ_2 r_1 θ_1 ... r_k θ_k
Corresponding measure:
$M = \sum_{k=1}^{\infty} r_k \delta_{\theta_k}$
Index/Parameter set:
$\theta \in \Theta, \Theta\ is\ a\ simplex \subset \mathbb{R}^3$
State space:
$b \in (0,1)$
... b_k b_1 θ_1 b_2 θ_k A θ_2
Corresponding measure:
$M = \sum_{k=1}^{\infty} b_k \delta_{\theta_k}$
$M(A) = \sum_{k=1}^{\infty} b_k \delta_{\theta_k}(A)$
$= b_1 + b_2$

Fig. 2. Illustrations of a parameter/index set, a state space, a realisation, and corresponding measure. The three illustrated examples are: Poisson, gamma, and beta processes.

increments. According to the Lévy-Itô decomposition, a Lévy process could be roughly decomposed into two components: one continuous component and one discrete component. When there is no continuous component and discrete jumps are positive, the Lévy process is called a pure-jump increasing Lévy process and is characterised by Lévy measure [177]. One merit of this pure-jump increasing Lévy process is its close relationship with Poisson process that each pure-jump increasing Lévy process is corresponding to a Poisson process using Lévy measure as mean measure. Because of this relationship, the realisation of each pure-jump increasing Lévy process could be viewed as a (countably infinite) set of points in a product space defined by its index/parameter space and state space, where each point is a pair <a parameter, a variable>. These points/pairs are all that is obtained from a stochastic process. Based on these points/pairs, a measure on parameter space can be constructed as $M = \sum_{k=1}^{\infty} \text{variable}_k\ \delta_{\text{parameter}_k}$, where M is a measure, k is the point index, and δ is Dirac measure. The constructed measure is called a random measure since the realisation of process is random. Furthermore, an important property of the Poisson process is its equivalence with completely random measures (CRM), noting that any measure M from a Poisson process is a completely random measure [100]. A number of stochastic processes in this area have been proven to be as special cases of CRMs, and these special stochastic processes can be manipulated thank to such property as discussed in more detail in the following section. Figure 2 illustrates these concepts using three examples. Next, we review some representative examples in this group in more details.

— *Poisson Process (PP).* A PP [102] over the product space $\mathbb{Z}^+ \times \Theta$ is represented as $\Pi \sim \text{PP}(\Pi_0)$, where Π_0 is a base measure over Θ. Note that, in theory, the base measure Π_0 could be any measure, but PP is normally used as the likelihood, so Π_0 is often a discrete measure from a prior, such as $\Pi_0 = \sum_{k=1}^{\infty} \lambda_k \delta_{\theta_k}$. A realisation Π is set of points $(\theta_k, N_k) \in \Theta \times \mathbb{Z}^+$, and is represented as

$$\Pi = \sum_k N_k \delta_{\theta_k} \tag{2}$$

where N_k satisfies a Poisson distribution parametrised by λ_k, i.e., $N_k \sim \text{Poisson}(\lambda_k)$.

- *Gamma Process (GaP).* A GaP [145] $\Gamma \sim \text{GaP}(c, \Gamma_0)$ is a Lévy process whose Lévy measure depends on two parameters: c and Γ_0. c denotes concentration parameter, and Γ_0 represents the base measure.

$$\nu(d\theta, dr) = cr^{-1}e^{-cr}\mathrm{d}r\Gamma_0(\mathrm{d}\theta) \tag{3}$$

  A realisation Γ is a set of points $(\theta_k, r_k) \in \Theta \times \mathbb{R}^+$ from a corresponding PP with mean measure ν and is represented as

$$\Gamma = \sum_k r_k \delta_{\theta_k} \tag{4}$$

  where r_k satisfies an improper gamma distribution. Since $\{r\}$ are all positive numbers, GaP is used as a prior for positive variables. For the computational convenience, PP is usually selected as the likelihood when GaP is used as prior just like the conjugate relationship between gamma and Poisson distributions.

- *Beta Process (BP).* A BP [87, 170] $B \sim BP(c, B_0)$ is a Lévy process whose Lévy measure depends on two parameters: c and B_0. c is a positive concentration parameter, and B_0 is a base measure on Θ. If B_0 is continuous, the Lévy measure of $\Theta \times [0, 1]$ of the BP is

$$\nu(\mathrm{d}\theta, \mathrm{d}b) = cb^{-1}(1-b)^{c-1}\mathrm{d}bB_0(\mathrm{d}\theta) \tag{5}$$

  A realisation B is a set of points $(\theta_k, b_k) \in \Theta \times [0, 1]$ from a PP with mean measure ν and is represented as

$$B = \sum_k b_k \delta_{\theta_k} \tag{6}$$

  where b_k satisfies an improper beta distribution. A realisation from BP can be viewed as a list of (countably infinite) probabilities (the summation does not need to be one), which is commonly used as, but not limited to, the prior for the rows of a matrix in factor analysis [13]. Another three-parameterised version is proposed in [14].

- *Negative-binomial Process (NBP).* A NBP [196, 200] $X \sim \text{NBP}(\Gamma_0, B_0)$ is another process over the product space $\mathbb{Z}^+ \times \Theta$, parameterised by two parts: $\Gamma_0 = \sum_k r_k \delta_{\theta_k}$ is a shape measure and $B_0 = \sum_k b_k \delta_{\theta_k}$ is the probability measure. A realisation X from NBP is a set of points $(\theta_k, N_k) \in \Theta \times \mathbb{Z}^+$ and the obtained measure is

$$X = \sum_k \kappa_k \delta_{\theta_k} \tag{7}$$

  where κ_k satisfies a negative-binomial distribution $\kappa_k \sim \text{NB}(r_k, b_k)$. NBP is an alternative as the likelihood for a GaP prior. Compared to PP where the mean and variance must be same, NBP is with a free variance-to-mean ratio which means the mean and variance are detached so it is suitable for the over-dispersed count data [197]. The BP for b is another prior for NBP, and it has been proven that NBP is conjugate with a three-parameterised BP [14].

- *Bernoulli Process (BeP).* A BeP [102] I is Lévy process $I \sim \text{BeP}(I_0)$ where $I_0 = \sum_k b_k \delta_{\theta_k}$ is a hazard measure and a realisation of BeP is

$$I = \sum_k \zeta_k \delta_{\theta_k} \tag{8}$$

where ζ_k satisfies a Bernoulli distribution $\zeta_k \sim \text{Bernoulli}(b_k)$. BeP is similar to PP, except that it gives weight 1 to points. Hence, BeP is normally used as a likelihood for the binary data and BP is often set as the corresponding prior.

There are also some processes that are not Lévy processes but can be derived from them. For example, Dirichlet process is a normalisation of GaP and Mondrian process is a multidimentional extension of PP.

- *Dirichlet Process (DP).* DP [57, 162] is the pioneer and foundation of the BNL. It is defined as: A DP, which is specified by a base measure H on a measurable space Θ and a concentration parameter α, is a set of countably infinite random variables that can be viewed as a (probability) measure on partitions from a random infinite partition $\{\Omega\}_{k=1}^{\infty}$ of Θ. An explicit representation of a DP using the stick-breaking process [162] is
$$G = \sum_k \pi_k \delta_{\theta_k} \tag{9}$$
where $\sum_k \pi_k = 1$. One property of DP is that: for any finite partition $\{\Omega\}_{k=1}^{K}$, the variables (measures on these partitions) from DP satisfy a Dirichlet distribution parameterised by the measures from a based measure H on relative areas
$$(G(\Omega_1), G(\Omega_2), \ldots, G(\Omega_K)) \sim \text{Dir}(\alpha H(\Omega_1), \alpha H(\Omega_2), \ldots, \alpha H(\Omega_K)) \tag{10}$$
where G is a realisation of $\text{DP}(\alpha, H)$ and Dir() denotes the Dirichlet distribution. The property $\sum_k \pi_k = 1$ in DP is normally used as the mixing distribution in mixture models, which makes DP the most active and successful stochastic process in BNL. Analogous to the relationship between gamma and Dirichlet distributions, DP could be seen as a normalised GaP, i.e., $\pi_k = r_k / \sum_j r_j$. An interesting extension of DP is the Pitman-Yor process (PYP) [161], which has an additional power-law property that is useful for language and network models.
- *Mondrian Process (MoP).* Let Θ be a box in $\mathbb{R}^D$ with a positive linear dimension. The Mondrian process [143] on Θ, denoted as $\text{MP}(\Theta)$, is a temporal stochastic process $(M_t)_{t \geq 0}$ takes values in guillotine partitions of Θ and its distribution is specified by the generative process Mondrian(Θ): the random variable M_t is the guillotine partition of Θ formed by cuts/nodes with birth time $t_b \leq t$.

Although the measure is not a widely used concept in computer science, we have still included it in this paper because the measure view of these stochastic processes is essential for developing aspects of BNL, such as new manipulations and posterior inference algorithms. The following example illustrates such concept: The bottom row of Figure 2 shows a sample (i.e., an infinite number of points): $(b_1, \theta_1), (b_2, \theta_2), \ldots, (b_k, \theta_k)$ from a beta process with a 2-simplex as the index space and (0, 1) as the state space. The measure composed by these points is: $M = \sum_k b_k \delta_{\theta_k}$, which is defined as a measure of the index/parameter space, i.e., it maps a subset of this space to a nonnegative value. For a subset of the 2-simplex (i.e., A in the figure), the measure value on A by M is $M(A) = \sum_k b_k \delta_{\theta_k}(A) = b_1 + b_2$. From this example, we can see that the summation in the measure definitions is to sum the weights of the points within the subset A. To make it further simple, particularly for the computer scientists, a measure can be simply considered as a set of points, e.g., $(b_1, \theta_1), (b_2, \theta_2), \ldots, (b_k, \theta_k)$ shown in Figure 2. Such points are, in fact, what we obtain and operate in the BNL. A concrete example to show how the points from measure definition are used follows: Suppose we have a number of data, i.e., $\mathbf{x}$, in 2-dimensional real space as shown in Figure 3. We can use the points from the Dirichlet Process, i.e., $(\pi_1, \theta_1 =< \mu_1, \Sigma_1 >), (\pi_2, \theta_2 =< \mu_2, \Sigma_2 >), \ldots, (\pi_k, \theta_k =< \mu_k, \Sigma_k >)$, to model these data as $p(\mathbf{x}) = \sum_k \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)$ where $\mathcal{N}(x; \mu_k, \Sigma_k)$ denotes a Gaussian probability density function parameterised by a mean vector μ_k and covariance matrix Σ_k. We can see that one component of a point θ is used as the parameters for a mixture, and the other component π is

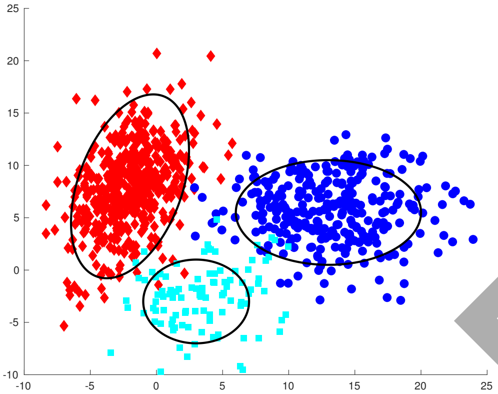

Fig. 3. Illustrations of the usage of a measure. A DP measure $G = \sum_k \pi_k \delta_{\theta_k}$ is used to model a set of 2-dimensional data. After the posterior inference, three mixtures/points (i.e., three ecllipes in the figure) are chosen to represent/model such data.

Table 1. Properties of stochastic processes. 'Stick-breaking' denotes if the process has a stick-breaking constructive representation; 'CRM' denotes if the process is a completely random measure; 'NRM' denotes if the process is a normalised random measure; 'Power-law' denotes if there is a version with power-law (cluster number to data number) phenomenon.

Stochastic Process	Stick-breaking	CRM	NRM	Power-law
Poisson Process		√		
Dirichlet Process	√ [162]		√	√ [167]
Beta Process	√ [13]	√		
Bernoulli Process		√		
Gamma Process	√ [145]	√		
Negative-binomial Process		√	√	√ [14]
Hierarchical Dirichlet Process	√ [176]		√	√ [12]
Indian Buffet Process	√ [164]			

used as a weight for a mixture. After the posterior inference, an appropriate number of points are selected to optimally model the data. As shown in Figure 3, three points/mixtures (i.e., eclipses) are chosen for modelling/representing the data. The above model is well-known as an infinite Gaussian mixture model [137].

Some important properties of the above stochastic processes are summarised in Table 1, including the stick-breaking construction methods and the power-law extensions. Note that these stochastic processes are only a very small portion of all stochastic processes. In addition to the above processes, there are a large number of stochastic processes with special properties existing in the current statistic literature, many of which may be valuable for modelling different data structures or resolving different learning tasks. We believe these processes are a huge undiscovered treasure that deserves more attention from the machine learning community.

3.2 Construction

Recall that the definition of a process only states its respective properties but does not provide a way to construct its realisations. Discovering a way to construct the realisation of a stochastic process (also known as a representation) is crucial in posterior analysis. The above subsection only gives the definitions of different processes from which we can judge whether a measure is from a specific stochastic process. In order to obtain an instance/sample from a process and perform the model inference, an explicit construction method is needed. Based on the properties of Beta distribution, stick-breaking is commonly accepted as an efficient construction methodology. Several sticking-breaking methods have been proposed for different processes. These are summarised below.

— *Sticking-breaking for Dirichlet Process [152].* Sticking-breaking continuously breaks a stick with a unit length, and constructs the weights using these breaks. For DP, the procedure is as follows:

$$G = \sum_{k}^{\infty} \pi_k \delta_{\theta_k} \qquad \pi_k = v_k \prod_{j=1}^{k-1} (1 - v_j) \qquad v_k \sim \text{Beta}(1, \alpha) \qquad \theta_k \sim H$$

where Beta() denotes a Beta distribution and v_k is the k-th random break in a unit stick with a Beta distribution parameterised by 1 and α. We can see that the weights π_k of G can be explicitly constructed/represented and that this G has already been proven to be a sample of DP.

— *Sticking-breaking for Hierarchal Dirichlet Process [167, 176].* There are two versions of stick-breaking for HDP: Teh's version [167] and Sethuraman's version [176]. The Sethuraman's verion is

$$\begin{aligned} G_0 &= \sum_{k}^{\infty} \pi_{0,k} \delta_{\theta_k} & \pi_{0,k} &= v_{0,k} \prod_{j=1}^{k-1} (1 - v_{0,j}) & v_{0,k} &\sim \text{Beta}(1, \alpha_0) \\ G_d &= \sum_{t}^{\infty} \pi_{d,t} \delta_{\theta_{d,t}} & \pi_{d,t} &= v_{d,t} \prod_{j=1}^{t-1} (1 - v_{d,j}) & v_{d,t} &\sim \text{Beta}(1, \alpha_d) \\ \theta_k &\sim H & \theta_{d,t} &= \theta_{z_{d,t}} & z_{d,t} &\sim \pi_0 \end{aligned}$$

where $z_{d,t}$ denotes an index to one of $\{\theta_k\}_{k=1}^{\infty}$. Compared to Teh's version, Sethuraman's version has the advantage in that the stick weights in different layers are decoupled, which makes the posterior inference easier. From this constructive representation, we can see the factor sharing property of HDP. G_d in the lower layer shares the factors $\{\theta_k\}_{k=1}^{\infty}$ of G_0 in higher layers. Another interesting point is that the constructions of π_0 and $\{\pi_d\}$ are independent, and the only connections between G_0 and $\{G_d\}$ are the relationships between θ_k and $\{\theta_d\}$.

— *Sticking-breaking for Indian Buffet Process [164].* To obtain $Z \sim \text{IBP}(\alpha)$, the stick-breaking procedure is

$$z_{i,k} \sim \text{Bernoulli}(b_k) \qquad b_k = v_k b_{k-1} = \prod_{l=1}^{k} v_l \qquad v_k \sim \text{Beta}(\alpha, 1)$$

where $b_1 > b_2 > \cdots > b_K$. It has been proven that $\{z_{i,k}\}$ from such a procedure satisfies IBP. As illustrated in Figure 4, there is an interesting connection between the above procedure and stick-breaking for DP: Suppose we fix $v \sim \text{Beta}(1, \alpha)$. Initially, the stick weight for DP is v_1 and $1 - v_1$ for IBP. The stick weight for DP in the k step is $v_k \prod_{i=0}^{k-1}(1 - v_i)$ and $(1 - v_k) \prod_{i=0}^{k-1}(1 - v_i)$ for IBP. Hence, we can see that at each breaking step, the weight for DP is the length of discarded stick while the weight for IBP is the length of the remainder of the stick.

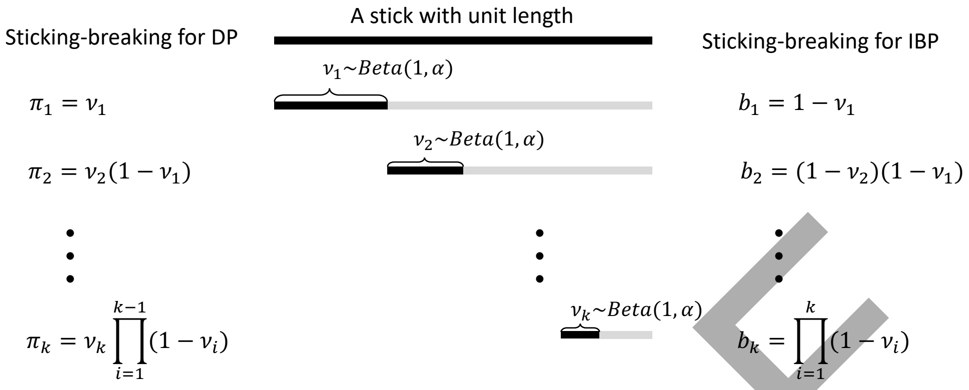
Sticking-breaking for DP
A stick with unit length
Sticking-breaking for IBP
$v_1 \sim Beta(1,\alpha)$
$\pi_1 = v_1$
$b_1 = 1 - v_1$
$v_2 \sim Beta(1,\alpha)$
$\pi_2 = v_2(1 - v_1)$
$b_2 = (1 - v_2)(1 - v_1)$
$v_k \sim Beta(1,\alpha)$
$\pi_k = v_k \prod_{i=1}^{k-1}(1 - v_i)$
$b_k = \prod_{i=1}^{k}(1 - v_i)$

Fig. 4. Illustrations of the connection between the stick-breaking procedures for DP and IBP.

- *Sticking-breaking for Beta Process* [13]. The stick-breaking construction procedure for $B \sim \mathrm{BP}(c, B_0)$ is different from IBP, in that there are more than one unit sticks. The procedure is

  $$B = \sum_{i=1}^{\infty}\sum_{j=1}^{C_i} v_{i,j}^{(i)} \prod_{l=1}^{i-1}(1 - v_{i,j}^{(l)})\delta_{\theta_{i,j}} \qquad v_{i,j}^{(l)} \sim \mathrm{Beta}(1, c) \qquad C_i \sim \mathrm{Poisson}(B_0(\Omega)) \qquad \theta_{i,j} \sim B_0$$

  where each atom has two subscripts (i, j). The stick weight is equal to the i-th break in the atom-specific stick, and the atom number follows a Poisson distribution.

4 MANIPULATIONS OF STOCHASTIC PROCESSES

The aim of the BNL is to use the above stochastic processes to model the data and then resolve the specific tasks. However, the modelling ability of any single stochastic process is limited. Therefore, in more complicated situations, two or more stochastic processes may need to be manipulated in different ways. A summary of the state-of-the-art designs of the manipulation follows, and their relationships to the most popular stochastic processes used in BNL are illustrated in Figure 5.

4.1 Layering

The base measure of a stochastic process can either be a continuous or discrete, and the realisation of a stochastic process is a random measure. Hence, it is possible to use a random measure from one stochastic process as the base measure for other stochastic processes to share the statistical strength between these processes. Take document modelling for example. The random measure in the upper layer could be the topic pool and the lower measure at the document level could be the topics in a document. The existing new Bayesian nonparametric priors based on this manipulation are summarised as follows.

- *Hierarchical Dirichlet Process (HDP).* HDP [167] is built by piling one DP on top of another DP(s) to transfer some statistical strengths from top layer to the bottom layer. For example, suppose a top measure G_0 over Θ is with a DP (α, H) prior and G_d is the measure in the bottom layer also with a DP (α_d, G_0) prior, then we have

  $$G_d \sim \mathrm{DP}(\alpha_d, G_0),\ G_0 \sim \mathrm{DP}(\alpha, H) \tag{11}$$

  Meanwhile, each random measure G_d is conditionally independent with others given G_0. HDP is the first and most successful model to use this manipulation; hence, many extensions to this

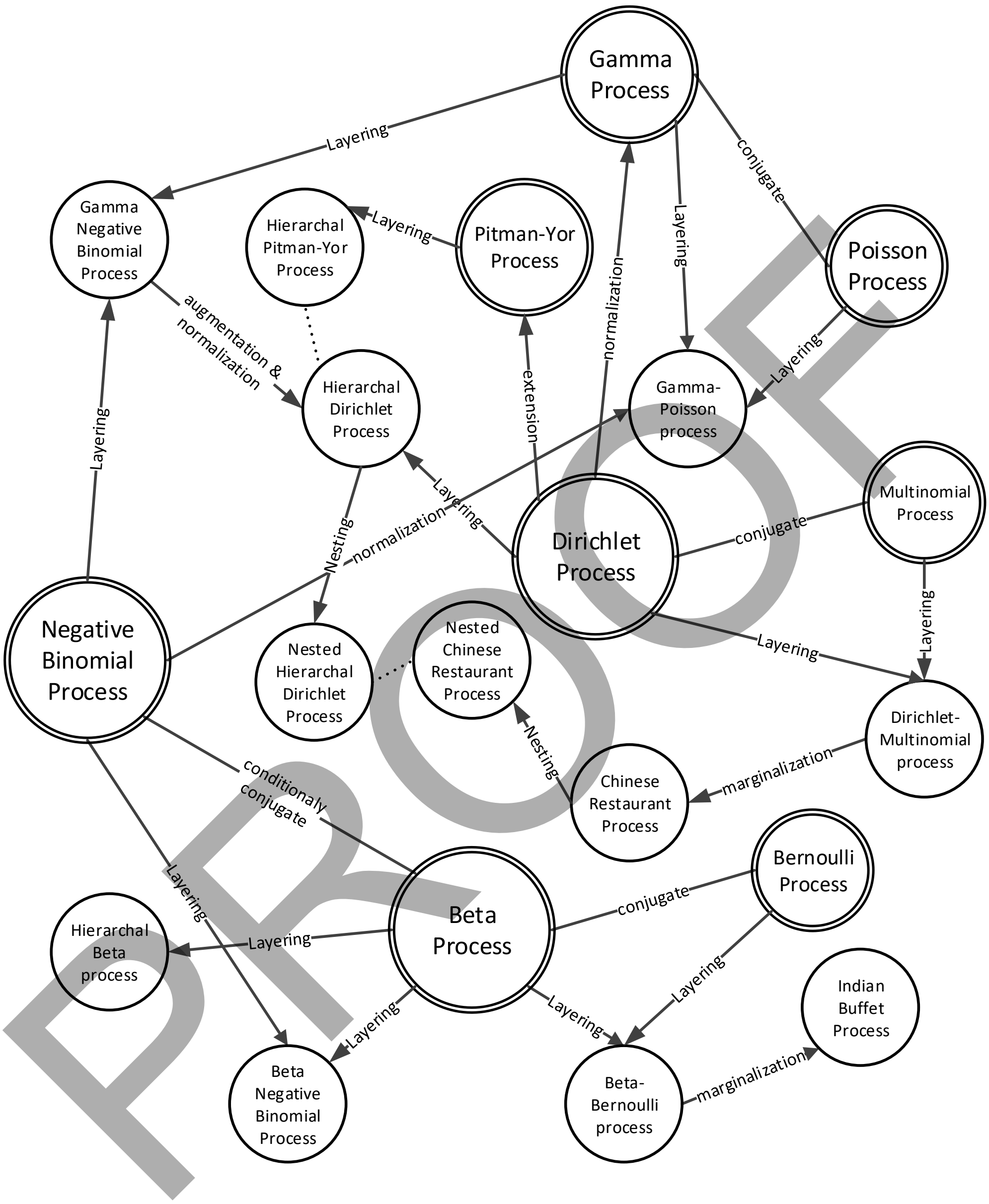
Gamma Process
Gamma Negative Binomial Process
Hierarchal Pitman-Yor Process
Pitman-Yor Process
Poisson Process
Hierarchal Dirichlet Process
Gamma-Poisson process
Multinomial Process
Dirichlet Process
Negative Binomial Process
Nested Hierarchal Dirichlet Process
Nested Chinese Restaurant Process
Dirichlet-Multinomial process
Chinese Restaurant Process
Bernoulli Process
Beta Process
Hierarchal Beta process
Beta Negative Binomial Process
Beta-Bernoulli process
Indian Buffet Process
Layering
conjugate
normalization
extension
augmentation & normalization
Nesting
marginalization
conditionaly conjugate

Fig. 5. Visualisation of relations between famous stochastic processes used in BNL. (A double circle denotes basic processes, and a single circle denotes outcomes from manipulations.)

model have been developed, including: supervised [36], incremental [67], nested [129], adapted [195], tree [17], evolutionary [185], and dynamic versions [139].

- *Hierarchical Pitman-Yor processes (HPYP).* Similar to HDP, HPYP [110] piles one PYP on top of other PYP(s). Unlike HDP, HPYP has two parameters: a concentration parameter α and a discount parameter β.

$$Y_d \sim \text{PYP}(\alpha_d, \beta_d, Y_0), \; T_0 \sim \text{PYP}(\alpha, \beta, H) \tag{12}$$

The discount parameter β controls the power-law property so that the PYP degenerates to a DP when $\beta = 0$. This power-law property means that HPYP is well-suited to document and network modelling.

- *Hierarchical Beta Process (HBP).* Akin to HDP, HBP [170] was proposed to make the different BPs share the same discrete measure (an infinite number of probabilities) from a global BP through

$$B_d \sim \mathrm{BP}(c_d, B_0), \;\; B_0 \sim \mathrm{BP}(c, H) \tag{13}$$

BP is an effective tool for the factor analyses and HBP were designed as a way to share factors during this process. Hence, they have been successfully used for document classification, convolutional factor analysis [26], shared subspace learning [80], image interpolation, and denoising [201].

- *Gamma-Negative Binomial Process (GNBP).* In contrast to HDP and HBP, GNBP [197] is not composed of a single process (e.g., a DP and a BP), but rather of two processes: a GaP and a NBP as follows:

$$I_d \sim \mathrm{NBP}(p, \Gamma_0), \;\; \Gamma_0 \sim \mathrm{GaP}(c, H) \tag{14}$$

More layers could be added by assigning further random measures to H from another GaP.

- *Beta-Negative Binomial Process (BNBP).* Analogous to a negative-binomial distribution, a BNP has two parameters. While GNBP places a GaP on one parameter, another BNBP layering manipulation [14] is to place a BP on another parameter as follows

$$I_d \sim \mathrm{NBP}(B_0, r), \;\; B_0 \sim \mathrm{BP}(c, H) \tag{15}$$

- *Enriched Dirichlet Process (enDP).* EnDP [173] could be viewed as a special case of HDP, which is defined as

$$G_\theta \sim \mathrm{DP}(\alpha, H_\theta), \;\; G_{\phi|\theta} \sim \mathrm{DP}(\alpha_{\phi|\theta} G_\theta) \tag{16}$$

and aims to model the joint distribution of (ϕ, θ) but with more flexibility and convenience (conjugacy) than a process that uses a direct product base measure.

4.2 Superposition

Superposition is used to combine two or more random measures together, like a “plus” operation. If layering is considered to be a form of *vertical* manipulation, the superposition is a *horizontal* manipulation.

- *Superposition of Poisson Processes.* According to the Superposition Theorem in [102], it is known that combining a set of independent Poisson processes yields a new PP with a mean measure that is the sum of the mean measures of the individual processes:

$$\Pi_1 \oplus \cdots \Pi_n \sim \mathrm{PP}(\mu_1 + \cdots + \mu_n) \tag{17}$$

where $\Pi_n \sim \mathrm{PP}(\mu_n)$. Given the relationship between PP and completely random measures, this theorem has become the foundation for the superposition of other processes.

- *Superposition of Dirichlet Processes.* Inspired by the superposition of PP, [112] proposed the superposition of DP as follows:

$$\begin{gathered} (c_1, c_2, \cdots, c_n) \sim \mathrm{Dir}(\mu_1(\Omega), \mu_2(\Omega), \cdots, \mu_n(\Omega)) \\ c_1 G_1 \oplus c_2 G_2 \oplus \cdots \oplus c_n G_n \sim \mathrm{DP}(\mu_1 + \mu_2 + \cdots + \mu_n) \end{gathered} \tag{18}$$

where $\{G\}$ is a set of independent DPs.

- *Superposition of Normalised Random Measures (SNRM).* Since DP is a special case of NRM, the superposition manipulation of DP has been extended to the more general NRMs [28] as follow; the superposition of n independent normalised random measures $\{\mu_j\}_{j=1}^n$ on Θ is

$$\mu_1 \oplus \cdots \oplus \mu_n = c_1 \mu_1 + \cdots + c_n \mu_n \tag{19}$$

where $c_n = \frac{\mu_n(\Theta)}{\sum_j \mu_j(\Theta)}$. This more general definition extends this manipulation to more situations.

4.3 Subsampling

Subsampling (randomly) selects parts of infinite components in a random measure from a stochastic process. When certain conditions are satisfied, the selected components form a new random measure from an underlying process.

- *Thinned Poisson Process [112].* Based on the Marking Theorem [102], let $\Pi \sim \text{PP}(\mu)$ be a PP on the space Θ, and a measurable function $q : \Theta \rightarrow [0,1]$. If independently drawing $z_\theta \in \{0,1\}$ for each $\theta \in \Pi$ with $P(z_\theta = 1) = q(\theta)$, a new PP is

$$\Pi_t = \{\theta \in \Pi : z_\theta = 1\} \sim \text{PP}(q\mu) \tag{20}$$

  where Π_t still satisfies a PP on Θ with mean measure $q\mu$.
- *Thinned Completely Random Measures (TCRM) [59].* Let $\Pi = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$ be a CRM on Θ and a measurable function $q : \Theta \rightarrow [0,1]$. For each point (θ_k, π_k), we define a Bernoulli variable r_k with $P(r_k = 1) = q(\theta_k)$ (independent with other $\{r\}$).

$$TCRM = \sum_{k=1}^{\infty} r_k \pi_k \delta_{\theta_k} \tag{21}$$

  TCRM is still a CRM, but with a mean measure $q\mu$. Note that PP is a special case of CRM. TCRM generalises this manipulation to all the CRMs, such as BP and NBP.
- *Thinned Dirichlet Process [112].* Let $G \sim \text{DP}(\mu)$ be a DP on Θ that can be represented as $G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$ and a measurable function $q : \Theta \rightarrow [0,1]$. For each k, independently draw r_k through $P(r_k = 1) = q(\theta_k)$,

$$G_t = \sum_{k: r_k = 1} \pi'_k \delta_{\theta_k} \sim \text{DP}(q\mu) \tag{22}$$

  where $\pi'_k = \frac{\pi_k}{\sum_j \pi_j}$ are the re-normalised coefficients for the selected components.
- *Thinned Normalised Random Measures (TNRM) [28, 29, 112].* Given an NRM $\mu = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$ on Θ and a Bernoulli variable $r_k \in [0,1]$. The TNRM is,

$$TNRM = \sum_{k: r_k = 1} \pi'_k \delta_{\theta_k} \tag{23}$$

  where $\pi'_k = \frac{\pi_k}{\sum_j \pi_j}$ are the re-normalised coefficients for the selected components. Note that DP is a case of TNRM.

4.4 Point-transition

Point-transition moves the points of a random measure according to an underlying probabilistic transition.

- *Point-transition of Poisson Process.* Based on the Transition Theorem [102], let $\Pi \sim \text{PP}(\mu)$ be a PP on space Θ and $T : \Theta \times \mathcal{F}_\Theta \rightarrow [0,1]$ be a probabilistic transition. From [112], the transformed measure is

$$\Pi_p = \{T(\theta) : \theta \in \Pi\} = \text{PP}(T\mu) \tag{24}$$

- *Point-transition of Dirichlet Process [112].* Let $G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} \sim \text{DP}(\mu)$, and its point-transition is

$$G_p = \sum_{k=1}^{\infty} \pi_k \delta_{T(\theta_k)} \sim \text{DP}(T\mu) \tag{25}$$

- *Point-transition of Normalised Random Measures (PNRM) [28].* Given an NRM $\mu = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$ on Θ, the point-transition of μ is to draw atoms $\theta_k^{'}$ from a transformed base measure to yield a new NRM as
$$PNRM(\mu) = \sum_{k} \pi_k \delta_{\theta_k^{'}=T(\theta_k)} \tag{26}$$
where T is a transition kernel.

4.5 Nesting

With a partition of a space by a process, e.g., DP, nesting is to further partition each area through another process. In other words, each component of a random measure is attached to an additional random measure.

- *Nested Dirichlet Process (nDP) [8].* The root partition is created by a DP, and each component in that partition is further attached to another DP. Repeating this procedure forms a tree of infinite width and depth. Each datum (e.g., a document) is associated with a path of this tree.
- *Nested Hierarchical Dirichlet Process (nHDP) [129].* This process generates a nested structure similar to a nDP but associates a datum with a new revised copy of the generated tree structure rather than a simple path.

4.6 Marginalisation

According to de Finetti's Theorem [2], an exchangeable sequences could be obtained through marginalising Finetti mixing measure out as follows,

$$P(X_1, X_2, \cdots, X_n) = \int \prod_{i=1}^{n} P(X_i|\theta)P(\mathrm{d}\theta) \tag{27}$$

where $P(\cdot|\theta)$ is a named mixture measure and $P(\mathrm{d}\theta)$ is named mixing measure. In BNL, the mixing measure comes from a stochastic process, so marginalising out a stochastic process results in a new exchangeable measure. Some examples of this type of marginalisation manipulation are shown below. Such marginalisation not only contributes new processes but also provides alternative constructive representations from the stick-breaking construction in Section 3.2 for the corresponding processes which is crucial for BNL inference.

- *Chinese Restaurant Process (CRP).* Marginalising a DP through,
$$CRP(X) = \int P(X|G)\mathrm{d}G,\ G \sim DP \tag{28}$$
where the mixing measure G comes from the DP. CRP [2] can be seen as a prior for random partitions of data points, and its name comes from the metaphor used to understand the process: In a Chinese restaurant, the i-th customer walks into this restaurant and chooses to sit at an occupied table with the probability $\frac{n_t}{\alpha+i-1}$ or a new table with the probability $\frac{\alpha}{\alpha+i-1}$, where n_t is the number of customers siting at table t. If the customer picks an occupied table, she eats the dish already on the table; if a new table is picked, she needs to order a new dish for the table from H. As a result, θ_i is the dish eaten by the i-th customer. $\{\theta_i\}$ from such process has been proven to be a sample from $\mathrm{DP}(\alpha, H)$.
- *Chinese Restaurant Franchise (CRF) [167].* CRF is a marginalisation of HDP. Based on CRP, the metaphor to understand this process is as follows: There are D Chinese restaurants with a shared menu. The i-th customer walks into the d-th restaurant and picks an occupied table at which to sit with the probability $\frac{n_{d,t}}{\alpha_d+i-1}$ or a new table with the probability $\frac{\alpha_d}{\alpha_d+i-1}$, where $n_{d,t}$ is the number of customers siting at table t in d-th restaurant. If this customer picks an occupied table, she just eats the dish already on that table; if a new table is picked, she needs to order a new dish.

The new dish is ordered from the menu according to its popularity. The probability that the new dish is the same as the one on other tables has a probability of $\frac{T_k}{\alpha+\sum_k T_k}$ and the probability that it is a new dish is $\frac{\alpha}{\alpha+\sum_k T_k}$, where T_k is the number of tables with the same dish θ_k. As a result, $\theta_{d,t}$ is the dish on table t of restaurant d, and $\theta_{d,i}$ is the dish eaten by customer i in restaurant d.

- *Nested Chinese Restaurant Process (nCRP) [141].* nCRP is a marginalisation of nDP. A metaphor based on CRP can be used to understand this process is as follows. There are infinite restaurants in a city and each restaurant has infinite tables. On each table, there is a card with the address of another restaurant. As such, all restaurants are organised into an infinite tree structure. A customer visiting this city dines first at the root restaurant and chooses a table using the CRP strategy. she notes the name of the restaurant on the card on her table and the next night she dines there, again choosing the table using CRP. This procedure repeats infinitely many times.
- *Nested Chinese Restaurant Franchise (nCRF) [1].* Analogous to nCRP and nDP, nCRF is a marginalisation of nHDP. The metaphor based on CRP and CRF to understand this process is slightly different: There are multiple cities, and each city has infinite restaurants with infinite table. The card on the table has the name of another restaurant in the same city as with nCRP. However, there are also an infinite number of menus organised using nCRP. The tree structures for all cities and menus are the same. A customer visiting a city dines first at the root restaurant and again chooese a table using nCRP. But this time, the meal is ordered from the menu (at the same position of the infinite tree) according to popularity as determined by CRF. Hence, nCRF combined with CRF helps to share the menus between restaurants, and nCRP helps to build the hierarchy.
- *Indian Buffet Process (IBP).* Marginalising a BP through,

$$IBP(X) = \int P(X|B)\mathrm{d}B, \ B \sim BP \tag{29}$$

where the mixing measure B comes from a BP. The basic IBP can be found in [77, 78]. The metaphor for IBP is: There is an Indian buffet restaurant with infinite number of dishes. The first customer walks into this restaurant and picks Poisson(α) number of dishes. The i-th customer follows the former customers and picks an formerly selected dish with a probability of $\frac{n_k}{i}$, where n_k is the number of customers picking dish k, and then she picks Poisson($\frac{\alpha}{i}$) number of new dishes that no one has chosen them before. The customer-dish selection from such procedure has been proven to be a sample from IBP(α). In machine learning, IBP is often used as the prior for a binary matrix with an unfixed number of columns, which is especially suitable for the factor analysis.

- *Nested Indian Buffet Process (nIBP) [31].* Similar to nCRP, a nested version of IBP is able to build a hierarchy where each layer is composed of a number of IBPs (i.e., the number of features in IBPs at the up-layer). Continuing the metaphor, there are infinite number of Indian buffet restaurants in a city, each has an infinite number of dishes, and a card with the address of another restaurant in the same city next to each dish. Hence, the restaurants in this city are organised as an infinite tree structure. A customer who visits the root restaurant in this city and chooses dishes using the IBP strategy but uses a layer-dependent probability Poisson($\frac{\alpha/\ell}{i}$) to choose any new dishes. The next day, the customer will visit all the restaurants on the cards next to her chosen dishes. Note that a customer can go to multiple restaurants with nIBP, rather than only a single one in nCRP.

5 POSTERIOR INFERENCE

After building an appropriate Bayesian nonparametric models for a specific task, the next step is to infer the latent variables defined in the model (more accurately, the posterior joint distribution of latent variables), given an amount of observed data. To obtain the posterior distribution, there are

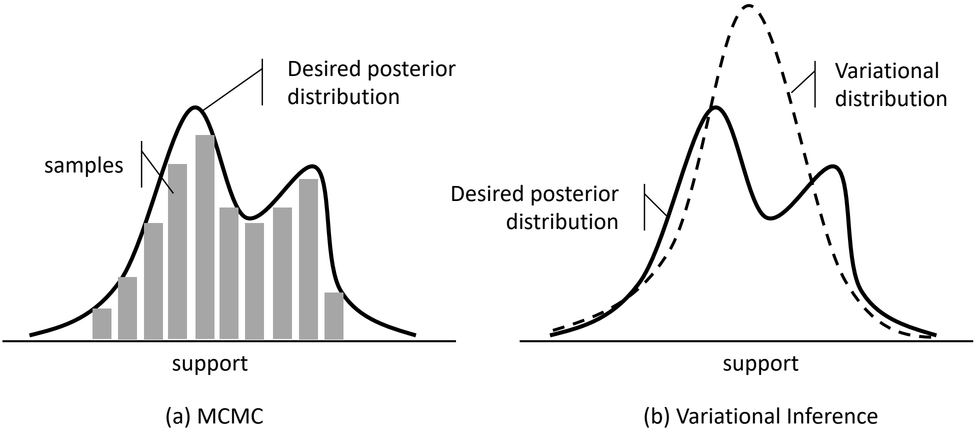
Desired posterior distribution
samples
support
(a) MCMC
Variational distribution
Desired posterior distribution
support
(b) Variational Inference

Fig. 6. The illustration and comparison of the ideas of two posterior inferences: (a) MCMC and (b) variational inference.

two main methodologies: sampling-based (i.e., Markov chain Monte Carlo) and optimisation-based (i.e., Variational inference). The most straightforward candidate for the Bayesian model posterior inference is Markov chain Monte Carlo (MCMC) (e.g., Gibbs sampling), which has also been widely adopted for Bayesian nonparametric models. The basic idea of MCMC [3] is to approximate a (posterior) distribution from its samples. To obtain the samples, a Markov chain is constructed with its stationary distribution as the desired (posterior) distribution. Figure 6(a) illustrates this idea, where the bins are the count of samples in the corresponding support areas. It is clear that more samples means a more accurate approximation. Although the Monte Carlo Markov chain (MCMC) methodology can, in theory, derive the exact posterior distribution of the latent variables, it is inefficient. As illustrated in Figure 6(b), an alternative for BNL is variational inference [10]. This methodology uses a set of (often simpler, independent and parameterised) variational distributions to approximate the real posterior distribution. This approach transforms a posterior distribution inference problem into a high-dimensional optimisation problem. With the help of gradients, this method can efficiently explore the parameter space of the variational distributions to approximate the desired posterior distribution as much as possible. In general, MCMC has better theoretical distribution approximation accuracy than variational inference when a sufficient number of samples are obtained, because introducing the variational distributions will introduce additional unnecessary approximation errors which do not exist in MCMC. In the case of big data, variational inference is more efficient than MCMC because obtaining sufficient samples in MCMC is very time-consuming and hard to evaluate. Yet gradient-based variational inference is able to explore the parameter space efficiently. The existing works for basic processes in the current literature on BNL literature are summarised below and in Table 2.

5.1 Markov chain Monte Carlo

While the (countably) infinite nature of BNL enables great modelling power, it also brings a challenge on model inference in that an infinite number of factors and weights make the posterior inference of the latent variables much harder. One commonly accepted solution in BNL is the truncation method [62, 181], which sets the component number so large that the given data would only adopt a subset of them, but it does introduce an approximation error. Another successful technique to resolve

Table 2. Existing inference algorithms for popular stochastic processes in BNL. Slice is short for slice sampling; VI for Variational inference; SMC for Sequential Monte Carlo; EP for Expectation Propagation.

Stochastic Process	Slice	VI	SMC	EP	Scalable
Dirichlet Process	√ [97]	√ [9]	√ [20]		√ [113]
Beta Process	√ [14]	√ [19]			
Gamma Process		√ [145]			
Hierarchical Dirichlet Process		√ [176]			√ [155]
Indian Buffet Process		√ [44]	√ [182]	√ [42]	

this problem is: *data/variable augmentation* [160, 172], also known as *Slice Sampling* [39, 126]. The existing Bayesian nonparametric models that use the slice sampling method for model inference are summarised below.

- *Slice Sampling for DP.* A realisation from DP can be represented as $G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$, where π_k are (a countably infinite number of) stick weights. The slice sampling for DP [97] introduces an auxiliary variable $u_i \sim \text{Unif}(0, \pi_{k_i})$ which functions as an adaptive truncation for the data i, and then only needs to sample $\pi > u_i$ for data i.
- *Slice Sampling for BP.* Instead of the fixed range used in DP, a positive decay function f : $\lim_{k\to\infty} f(\pi_k) = 0$ controls the support for the auxiliary variable $u_k \sim \text{Unif}(0, f(\pi_k))$ [14]. Compared to the function used in DP, the decay function is more flexible and more applicable to many other models.

5.2 Variational inference

Using variational inference involves two important components: setting variational distributions and choosing optimisation methods. Clearly setting the variational distribution can significantly reduce the additional approximation error and appropriate optimisation methods can boost the inference efficiency and adapt to the complex scenarios, e.g., large-scale and streaming data. Existing work on variational inference for BNL is summarised below.

- *Ordinary Variational Inference.* The form of variational inference for BNL is based on a stick-breaking representation where the latent variables in this representation include stick weights, atom parameters, and data assignment indexes. Since stochastic processes such as DP [9, 106] and GaP [145] have the potential to produce an infinite number of atoms, we have opted for a truncation method to ensure that only a finite number of atoms need to be approximated with a finite number of mean-field (latent variable) distributions. Other similar works have been applied for HDP [9], IBP [44], and nCRP [174].
- *Collapsed Variational Inference.* As stated, there are normally three groups of latent variables in BNL (i.e., stick weights, atom parameters, and data assignment indexes). Sometimes stick weights could be marginalised out to make the inferences of DP [105] and HDP [147, 168] more accurate and efficient.
- *Online/Stochastic Variational Inference.* Instead of coordinate-ascent optimisation which is not efficient for large datasets because a full pass of all the data at each iteration is required, stochastic optimisation is used to update the variational parameters [15, 176]. At each iteration, a number of data are sampled from the entire dataset before updating the variational parameters. This optimisation could improve the inference efficiency because of the estimated noisy gradients of the variational objective have been proven to be natural gradients of the Kullback-Leibler divergence objective [90].

- *Truncation-free Variational Inference.* One problem with the above variational inference methods is that the truncation is needed, which brings additional approximation errors that are resolved with a locally collapsed method. Here, the global latent variables are marginalised, and the distribution of the local variables could be sampled [175]. This method can be viewed as a combination of the sampling and variational inference methods. Another idea to avoid truncation is variational DP [106], where a tying assumption is given. That is, the variational distributions of components larger than T are set as priors, and T are adaptively increased with a decrease in the optimisation objective and KD-tree-based data organisation is introduced to accelerate the speed of inference.
- *Streaming Variational Inference.* Streaming variational inference algorithms were proposed to handle data that arrives sequentially in the form of a data stream. This method does not require data to be preserved in memory, which means it can be discarded after processing. The idea is to decompose the joint posterior to a recursive form where the distribution of the incoming datum is derived from the posterior distribution of the former observed data. Different techniques can then be used for approximation, such as a designed variational stochastic process with independent factors for DP [111] or more general extensions based on assumed density filtering [159] or component identification [16].

5.3 Scalability

Considering the exponentially increasing amount of data in many areas, the ability to handle Big Data is also a research direction for BNL. The idea is to extend current inference algorithms, particularly those designed for a single processor/machine into parallel versions for multiple processors/machines. Existing inferences algorithms for this problem are mainly categorised into the following two groups: MCMC and variational.

- *Parallel MCMC.* [155] proposed a parallel MCMC for HDP using an asynchronous method. The advantages of this approach are that it is easy to incorporate new data and processors, and it is extremely fault-tolerant. The disadvantages is with additional approximation. To overcome this disadvantage, further parallel MCMC for DP, HDP, and PYP were proposed [48, 113, 179] based on the inverse-superposition of DPs. Here, each processor or machine handles one supercluster (i.e., one of DPs). Generating data-based super-clusters generation further improves the efficiency [23]. Although the above methods could use marginalised representations of DP or HDP to avoid truncation, a slice sampling-based parallel MCMC [68] has been developed to explicitly sample the stick weights in DP and HDP in an elegant way.
- *Parallel Variational Inference.* In this form of inference, the posterior distribution is decomposed for the streaming and parallel inference, and a component identification algorithm resolves the mismatch problem when merging different variational posteriors from different processing nodes [16]. Instead of obtaining the exact variational approximation of the real posterior distribution by combining subposteriors (from different processing nodes) together, a Markov chain is designed to collect samples from the variational approximation [127], which again can be viewed as a combination of variational and sampling.

5.4 Others

Some other inference methods that have been applied to BNL are:

- *Sequential Monte Carlo.* Also known as particle filtering, this approach approximates the posterior distribution through a large collection of samples (i.e., particles) that are propagated over time and updated by sequential importance sampling. The sequential Monte Carlo method has been applied to DP mixture with time varying mixtures [20], beta-binomial DP mixtures [116], general conjugate DP mixtures [56], and nonparametric Bayesian matrix factorisation [182].

Table 3. Summery of the classical models in machine learning and their corresponding Bayesian nonparametric extensions.

Classical machine learning models	Nonparametric extensions
Gaussian Mixture Model (GMM)	[137]
Latent Dirichlet Allocation (LDA)	[167][4]
Hidden Markov Model (HMM)	[167] [64] [65]
Linear Dynamic Systems (LDS)	[21, 61]
Support Vector Machine (SVM)	[202]
Nonnegative Matrix/Tensor Factorisation (NMF/NTF)	[42, 182] [108, 189, 192]
Mixed Membership Stochastic Block models (MMSB)	[54, 122]
Partially-Observable Reinforcement Learning (PORL)	[46]
Conditional Random Field (CRF)	[95, 135]
Markov Random Field (MRF)	[25]
Author Topic Model (ATM)	[190]
Principal Component Analysis (PCA)	[51]
Bayesian Inverse Reinforcement Learning (BIRL)	[32, 33]

- *Power Expectation Propagation.* [121] This method generalises expectation propagation and variational inference using a flexible α-divergence. An example is nonparametric Bayesian matrix factorisation [42].

6 APPLICATIONS IN MACHINE LEARNING

BNL is an efficient mechanism that has been used to resolve many tasks in machine learning. Indeed, many famous models in machine learning already have a Bayesian nonparametric counterpart, as summarised in Table 3. This section presents a selection of studies on the application of BNL arranged according to task. Each category of task explores how BNL has been used a solution task and the additional benefits BNL has brought. Note that these state-of-the-art works not only show the wide applicability of BNL, but also showcase how the BNL procedure has been used in practice. Some popular stochastic processes in BNL which suit different machine learning tasks are summarised in Table 4.

6.1 Supervised learning

Sometimes data have labels (also known as responses), such as the emotion tags, the GDPs of countries, network relations, and so on. Supervised learning is a method for modelling the relationships between data and these responses/covariants to make predictions - a basic task in machine learning. Supervised BNL tends to fall into two categories.

- *Generalized linear models (GLM)-based*, where the data are assigned to a number of clusters, and the responses (for classification) of the data are modelled by multinomial logistic models (also called 'softmax') [82, 153]. A DP is used as a prior for weighting $\{\pi_k\}_{k=1}^{\infty}$ the clusters and the parameters $\{\theta_k\}_{k=1}^{\infty}$ are used in the multinomial logit models. Although the data and the responses in each cluster have a linear relationship, multiple clusters make it possible to capture non-linear relationships. This idea has been further extended for group data using HDP [36, 193].
- *Covariant space-based.* The relationship between the covariant space and the data is captured by dependent stochastic processes by: 1) setting the base measure as a special stochastic process, such as a single-variable stochastic process [114] and a multi-variable Gaussian process [69]; 2) varying the stick-breaking procedure, such as linking stick weights through a stochastic process

Table 4. Summery of popular stochastic processes in BNL which suit different machine learning tasks and real-world tasks.

Machine learning tasks	Processes	Real-world tasks	Processes
Supervised Learning	DP, HDP, GP, BP, CRP, IBP	Text Mining	DP, HDP, IBP, PYP, GaP
Factor analysis	IBP, BP, GaP	Natural language processing	CRP, DP, HDP
Transfer learning	HDP, HBP, PYP, DP, nCRP	Computer vision	BP, IBP, DP
Tree structure learning	nCRP, HDP, DP, PP, GaP	Biology	DP, HDP, PYP
Relational learning	DP, IBP, PP	Music analysis	HDP, GP
Reinforcement learning	HDP, PYP, GP, IBP, BP	Robots	DP, HDP, GP
Causal inference	GP, DP		
Metric learning	BP		

[115], a kernel function [50], or a permutation stick-breaking order [34, 115]; 3) auxiliary PP in a covariant space, such as kernel beta process [140] and correlated normalised random measure [76]; and 4) revising the seating mechanisms in CRP or IBP, such as ddCRP [7] and ddIBP [71]. More discussions on this aspect can be found in [60].

6.2 Factor analysis

Factor analysis describes or captures the variability of observed or correlated data with the help of a collection of unobserved variables called factors, which is useful for denoising or storage [171]. The mathematical definition is $Y = \Phi X + E$, where Y is data, Φ is factors, X is the factor loading matrix, and E is the error. Traditionally, the dimensionality of Φ needs to be given in advance, but BNL relaxes this requirement. Since IBP defines a distribution for infinite (on columns) binary matrices, it has become a significant cog in this field [77]. Below, we summarise the studies into two important branches of factor analysis: Principal component analysis and Non-negative matrix/tensor factorisation.

- *Principal component analysis (PCA).* PCA is a very popular tool for dimension reduction, but the selection of the number of significant components requires strong background knowledge that is normally unknown. Unlike general factor analysis, PCA aims to project the data to an space spanned by orthonormal vectors. A BNP-PCA [51] is defined as
$$Y = P(X \odot Z) + E \tag{30}$$
where P is an orthogonal matrix that satisfies $P^T P = I$, and Z is with an IBP prior.
- *Non-negative matrix/tensor factorisation (NMF/NTF).* Same with PCA, the nonparametric versions of NMF/NTF are also mainly based on IBP or BP. The basic idea is to factor the data matrix or tensor into two matrices: one is binary (mask) matrix and the other is a factor matrix, and an infinite prior is given to the binary matrix, such as IBP [42, 182] or BP [108].
$$Y = XZ \tag{31}$$
where Z has an IBP prior. Instead of providing a prior to one matrix, another idea is to simultaneously given two matrices a joint prior, e.g., a dependent IBP [192]. A similar idea was also applied to NTF based on a tensor-variate Gaussian process or a tensor-variate t process [189].

Due to the infinity property of those processes, the nonparametric extensions of NMF or NTF can remove the requirement of prefixing the matrix dimensions. Another category of research is to simultaneously assign an infinite prior (i.e., a GaP [6]) to the combination parameter of the likelihood function as

$$Y = X\Gamma M \tag{32}$$

where Γ is with a GaP prior.

6.3 Transfer learning

Transfer learning [131] is a learning scheme that extracts transferable knowledge from the a source domain(s) and reuses this knowledge in a target domain to resolve the problem of insufficient data in traditional machine learning schemes. It appears that the core of the transfer learning is to extract transferable knowledge between domains which can be generalised to the target domain. BNL is good at generalisation because of its ability to build a flexible prior and, hence, has been applied to transfer learning. The existing works in this area are summarised according to the different forms of the transferable knowledge below.

— *Transfer by sharing factors.* Factors are real-valued vectors and can also be viewed as high-level semantic descriptions of data; this allows them to be transferred from one domain to another. In Bayesian nonparametric joint factor analysis (NJFA) [79], domains are jointly factorised with shared factors and domain-specified factors. In contrast to simply sharing factors between two domains through a matrix product and summation, factor sharing in NJFA relies on an ingenious HBP prior. HBP is used to control the assignment of factors between domains, and its advantages are: 1) factors are shared by domains through hierarchically dependent beta processes; 2) the number of both shared and domain-specified factors does not need to be prefixed but rather automatically learned from the data.

— *Transfer by sharing topics.* Different from factors, topics are a set of latent variables characterised by the unit summation and are used as the transferable statistical strengths between domains. BNL makes these transferable statistical strengthens more flexible by giving them Bayesian nonparametric priors. In Clustered Naive Bayes [142], a number of Naive Bayes share topics (named parameters in [142]) from a DP. Further tasks within the same group share the same parameters so that well-trained tasks can help to train the insufficiently-trained task. As we discussed in Section 4.2, HDP defines a way to share statistical strengths, which has also been applied to transfer learning [18]. Here, categories share the topics/clusters. HPYP is also able to benefit the transfer learning analogy to HDP. For example, a set of basic topics is first generated for sharing, and each domain uses these topics as the base measure of a Transformed PYP, which transforms these topics to another set of topics using a domain-specific transformation matrix [27]. Unlike HDP, the activated topics are controlled by an additional IBP. The advantage over HDP is that sharing of topics and their weights could be decoupled, which makes it possible to share low-weight topics between domains [53].

— *Transfer by sharing tree.* Knowledge is inherently multi-granular, and the more general the knowledge in the source domain, the larger the probability it will be reusable in a (related) target domain compared to specific knowledge. Hence, sharing trees has emerged as an efficient method for transfer learning. In terms of the tree construction, BNL could contribute a rather flexible tree prior. One example is the transfer Hierarchical LDA (thLDA) [98], which transfers the knowledge in an existing tree to a target domain, where a path of the tree from the root to the bottom node is sampled for each document in the target domain through nCRP. In addition to thLDA, the second example is [146] where a tree structure is built with a fixed three-layer depth and unbounded width through nCRP. In this model, domains are treated as nodes in the bottom layer (named 'level 1'), the parent nodes of the domains are named *supercategory* in the middle layer (named 'level 2'), and the root node in the top layer (named 'level 3') is a set of two variables. Each domain is characterized by a Gaussian distribution, and domains are expected to share similar parameters with their parent nodes (i.e., supercategories or root nodes). The third example treats the parameters in the tree as transferable knowledge to assist the classification of classes with less labeled data [156] where a tree structure with leaf nodes as labels/domains is built with an nCRP prior. Another example is the polylingual tree-based topic model [91], which uses a similar method to nCRP. A tree is assigned a path through a probability made up of the product of the weights of branches in the path, where the tree is also separately built from the source domain. Without resorting to nCRP, the doubly Hierarchal Pitman-Yor Language Model [183] builds a latent HPYLM as the transferable knowledge and, simultaneously, two separate HPYLMs are built for the two domains with the latent HPYLM as part of the base measures in PYPs in two HPYLMs.

6.4 Tree structure learning

Tree structures play an important role in machine learning because they are pervasively applied and reflect the human habit of organising information. Thus, learning out a hierarchical structure from plain data has attracted a great deal of attention from researchers in the Bayesian nonparametric field. Compared to other approaches to this task, Bayesian nonparametric models have the advantage of a more flexible hierarchical structure. The lack of bounds on the structure's width and depth makes it much easier to incorporate the new data.

- *nCRP-based.* Here, the nCRP views a tree as a nested sequence of partitions. A space is first partitioned by a CRP and each area in this partition is further partitioned into several areas to generate a tree of potentially infinite depth and branches. A datum (i.e., a document) is associated with a path in the tree using DP or Flexible Martingale [157] priors with the nCRP [8]. The datum could also be associated with a subtree of the generated tree using HDP priors in nHDP [129] instead of with a path.
- *Stick-breaking-based.* With this method, an iterative stick-breaking process is used to construct a Pólya Tree (PT) [133] in a nested fashion. A datum is associated with a leaf node of the generated tree, and the traditional stick-breaking process is revised to generate breaks within the tree structure to result in a Tree Structured Stick Breaking (TSSB) [72]. A datum is attached to a node in the generated tree.
- *Diffusion-based.* Both Kingman's coalescent [101, 163, 165] and Dirichlet Diffusion Tree (DDT) [125] define a prior for an infinite (binary) tree. The idea is that the data are generated by a diffusion procedure with several divergences during this procedure. An additional time-varying continuous stochastic processes (i.e., Markov process) is needed for the divergence control. A datum is placed at the end of branches in the diffusions. DDT has been extended into a more general structure: multifurcating branches by a Pitman-Yor Diffusion Tree (PYDT) [103] and to a feature hierarchy by Beta Diffusion Tree (BDT) [84].
- *Others.* Motivated by the deep belief network (DBN) [86], the Poisson gamma belief network (PGBN) [199] learns a hierarchical structure where nodes have nonnegative real-valued weights rather than binary-valued weights in DBN and the width of each layer is flexible rather than fixed. Each layer's nodes can be viewed as an abstract feature expression of the input data.

6.5 Relational learning

The aim of relational learning (also known as stochastic relational learning) is to analyse, model, and predict the relationships between entities. Many data contain relationships by nature, e.g., social networks, traffic networks, protein interactions, and user-item relationships. Hence, a typical and practical application of relational learning is predicting potential future connections using a trained model. In traditional Bayesian learning, the key to better predictions is learning the hidden generative procedures in a network given a prior that encodes different understandings and explanations. BNL holds the potential provide a much more flexible prior [149]. The existing research in this direction is summarised below.

- *Latent class models.* The idea of these models is to group nodes into a number of classes based on the same behaviours. For example, all the nodes in one class may have the same probability of being related to all the nodes in another class [188] or to all the nodes that sit outside their own class [30]. DP is usually the main machinery behind latent class modelling due to its flexibility on latent classes. [188] was the first to model relationships using BNL. By Xu et al.'s definition, the probability of observing a link $l_{a,b}$ is modelled as $p(l_{a,b}|A_a, A_b) = \theta_{a,b}$ where A_a and A_b are the attributes of nodes linked by $l_{a,b}$, and $\theta_{a,b}$ is the link probability from a DP prior to make it more flexible. Further, the attributes of nodes are generated according to the latent variables Z_a, i.e., $p(A_a|Z_a)$, and the link $p(l_{a,b}|Z_a, Z_b, W)$ is modelled using these latent variables rather than directly using node attributes where W is a model (connectivity) parameter. These latent variables can be understood as the 'interests' or hidden features (i.e., latent classes) of nodes that determine the formation of the links. Using a DP prior for Z means the data itself determines the optimal number of states for the variables [99, 187], which has been proven to be effective for social network analysis [186]. This link patterns between groups are also set to be shared across multiple networks in [93] for multi-relational data. Rather than modelling the link probability between groups, [188] models the probability that a node belongs to a class, and nodes with similar patterns are grouped into classes.
- *Latent feature models.* [120, 122] model a link as $p(l_{a,b}|Z_a, Z_b, W)$, where Z_a is a binary vector in [187] and IBP is used as the prior for Z. This means that each node is either characterised by a series of hidden features or it belongs to a series of clusters. A DP can also be used to supplement the IBP by creating sub-clusters within each feature/cluster [130].
- *Random network models.* In contrast to the above models that focus on a given network with a fixed number of nodes and links, random network models try to model the generation of both links and nodes to allow the network to grow as more nodes join. Another advantage of random network models is their ability to model sparse networks (i.e., where the number of nodes is $o(n^2)$). Essentially, a network is represented as a link sequence by $Z = \sum_{n=1}^{N} z_{i,j}\delta_{\theta_i,\theta_j}$, where θ_i for each node is from a CRM or NCRM, e.g. PP [22] and DP [180]. Additionally, with a simple transformation, the same idea could also be applied to binary, integer, and multi-relational networks.

6.6 Reinforcement learning

The reinforcement learning (RL) is a category of sequential decision-making problems that seeks to find the optimal policy for maximising its long-term profit given an agent that interacts with an uncertain environment. RL is normally modelled through a Markov Decision Process (MDP) which is usually represented as tuple $< S, A, T, R >$ in which S denotes the environment state set, A denotes the action set of the agent, $T(s'|s, a)$ is the state transition function from the state s to s' after the action a, and $R(r|a, s')$ represents reward function that defines the profit from action a. The agent needs to find the optimal policy $\pi(a|s)$ for attaining a long-term reward through the interaction with the environment. While RL has many applications, it is typically used in robots and games, such as the famous AlphaGo program [154]. In RL, BNL could provide a principled way to tackle the core task of exploration-exploitation problem [73] and the ability to model the underlying environment dynamics in a flexible way. The existing studies on BNL in RL are summarised below according to whether the target is a policy or a reward.

— *Forward reinforcement learning*. Sometimes, the state of an environment cannot be observed or can only be partially observed, which provides incomplete information (termed observation o). Determining the optimal policy in such situations is known as partially-observable RL and is modelled as a Partially-Observable MDP. Here, the hidden states or environmental models need to be inferred through the historical (interaction) data. In other words, the transition, observation, and reward functions need to be learned from a history. GP [37, 40] is a natural and powerful prior for unknown functions and is particularly efficient in a continuous state space. However, in discrete state spaces, a Bayesian nonparametric approach based on HDP-HMM [45–47] models the hidden state transitions as stick weights from HDP. The base measure is composed of two parts: one is a function for reward evaluation, the other generates observations. Another form of forward reinforcement learning is based on PYP [46], which extends the classical Deterministic Markov Models (DMM) [117] using the same method.

— *Inverse reinforcement learning*. In contrast to forward RL, the target of inverse reinforcement learning is a reward function. The reward function is given to other elements of an MDP and the trajectories/demonstrations (i.e., state-action pairs) determined by experts with an assumption that experts made decisions with knowledge of the underlying environment model. Classical methods model this reward function as a linear combination of handcrafted features, which apparently limits the ability to capture the complex behaviours, but this constraint is relaxed using GP [107]. Instead of improving the learning capability of a single function, a different idea to resolve the complex function learning is to partition the observations [118, 119] or divide experts [32] into groups and learn relatively simple reward function for each group based on DP. A similar idea was used to resolve switched MDP using sticky HDP-HMM [158] as the prior with the additional ability to partition the data based on their temporal properties, which is beneficial for representing rich behaviours. Another IBP-based or BP-based idea [33, 136] is to learn the composite features as conjunctions of the original features and then learn the reward function for the composite features rather than original features.

6.7 Others

Causal inference and metric learning are other interesting applications of BNL.

— *Causal inference*, which estimates the effect of an action, such as a medical treatment or a sales policy, given some observational data. Consider the following basic scenario. Suppose there are a number of patients X, a treatment regime A, and observed medical history data $< X, Y, A >$. X takes A which leads to the result Y, noting that each X can only choose to take $A = 1$ or take $A = 0$ at any given time. The goal is to estimate the effect of the treatment regime, i.e., $\mathbb{E}[Y|X, A]$ or $\mathbb{E}[Y|X, A = 1] - \mathbb{E}[Y|X, A = 0]$. In [85], GP is used to approximate the conditional distributions $P(Y|X, A = 0)$ and $P(Y|X, A = 1)$ separately. Another strategy is to approximate the joint distribution $P(X, Y, A)$ and then obtain the conditional expectations through marginalisation [144], while a counterfactual GP [150] is designed for the continuous-time scenarios.

— *Metric learning*, which aims to learn out a similarity measure that forces similar data to be close and dissimilar data to be further apart. A Bayesian nonparametric model for this task is based on BP [5], where the data likelihood is defined as $p(X|H) = f(H)$ and H is the latent features of the data and decomposed as $H = S \odot Z$. BP is used as the prior for Z, and an additional regulariser is introduced to the target variational inference optimisation function to control the similar and dissimilar data pairs.

7 APPLICATIONS IN REAL-WORLD TASKS

The first study on BNL in the machine learning community concerned document modelling [167]. However, as BNL has developed over the years, it has attracted attention from researchers in other fields, such as computer vision and robots. In turn, the increasing number of scenarios BNL is being applied to is prompting further development of theories and techniques in this area. A summary of the main fields of application for BNL is described in this section and Table 4.

7.1 Text mining

The goal of text mining is to understand a document corpus. Below, we have summarised the various text mining applications for BNL according to the different types of document corpora to be mined.

- *Single corpus.* This task seeks to learn the knowledge shared between different documents in a corpus. HDP [167] was designed to deal with such tasks by assigning a random measure to each document using a generated global random measure as a base measure. However, with an HDP approach, the word vocabulary needs to be fixed in advance, which means new words cannot be incorporated if subsequent documents are added to the corpus. Latent IBP Compound Dirichlet Allocation [4] relaxes this constraint through a four-parameter IBP. Further, by using dependent GaPs or a mixed GaP-NBP, additional information from the corpus can be introduced into a model. [191] incorporated the links (e.g., citations between scientific papers) between the documents using a dependent GaPs and authorship information using a mixed GaP-NBP [190].
- *Multiple corpora.* This task aims to learn the knowledge shared across different corpora. The first attempt at such a task was undertaken by a three-level HDP [167] that assigned each corpus with a random measure using a generated global random measure as the base measure. Through this hierarchical method, the topics shared by different corpora could be inferred. Moving beyond shared topics, the difference between topics at different corpora is also learned by Differential Topic Models [27] based on PYP.
- *Multiple time-varying corpora.* This task aims to learn the shared knowledge across time-varying corpora. Beyond the idea of HDP, an Evolutionary HDP [194] was proposed to deal with this task, which added another time-varying dependency between the random measures for different corpora. The base measure of a corpus is the (convex) combination of two measures: the global measure and the measure at the previous time stamp.

7.2 Natural language processing

The main applications of BNL in the field of natural language processing are detailed below.

- *Word segmentation.* This task is to identify word boundaries in continuous speech. CRP is used to capture the word sequence generation process [75] with two options: if a novel lexical item arrives, generate a phonemic form; if not, choose an existing lexical form. This idea has subsequently been applied to HDP and performs better than DP.
- *Phrase alignment.* This task aims to find frequent phrase pairs from bilingual texts for the benefit of phrase-based translation systems. Since bilingual texts do not come already segmented, and the number of aligned phrase pairs is unknown, DP and HDP are used as priors to identify each aligned phrase pair in a probabilistic model for this task [41].
- *Unsupervised part-of-speech (PoS) tagging.* Pos tagging is to mark the words in a text with its corresponding part-of-speeches, which is the basis of text analysis. Infinite HMM is used to model the word sequence with the help of an HMM and the hidden states (PoS tags) are unbounded with the help of HDP [66].

7.3 Computer vision

In the field of computer vision (i.e., image processing and video processing), BNL has been successfully applied to the following problems:

- *Image interpolation.* It is one of the most common image processing tasks and mainly includes image resizing and remapping. The goal is to factor an image by a linear, but finite, combination of dictionary atoms [83], which are represented as a matrix. BP and IBP [198] are used as the prior for this matrix to avoid the need to predetermine the number of dictionary atoms with an additional sparsity property. Zhou et al. [201] proposed a dependent HBP to incorporate a constraint on the patch positions in an image.
- *Motion capture segmentation.* This task aims to identify a finite number of hidden dynamic behaviours in multiple time series given that each time series may contain multiple dynamic behaviours. The mapping relationship between time series and dynamic behaviours is expressed as a matrix, which is given a BP prior to make it infinite [63].
- *Background subtraction.* This task aims to delineates background information in video streams. The background is modelled by a probability density and performs better as a multi-mode probability density with dynamic backgrounds. The appropriate number of modes in the distribution tends to depend on the target video stream. Hence, [81] proposed a DP-based Gaussian Mixture Model to allow the number of nodes to be determined by the data.

7.4 Biology

The field of biology is fortunate to boast a massive amount of data, much of which has been derived from clinical trials and personalised medicines. BNL provides an efficient paradigm in a fully model-based probabilistic framework that is highly flexible and adaptable [49]. Therefore, BNL has the potential to be extremely useful in biology, because of the lack of knowledge on the parametric model establishing.

- *Brain MRI tissue classification.* MRIs are an effective and routine diagnostic tool, and, as such, the accurate automatic classification of brain MRI can assist doctors' diagnoses. Mixture model clustering algorithms have dominated this task for some time due to their relatively good performance. However, more recently, nonparametric extensions to these algorithms, such as DP [35] and HDP [94], have performed even better.
- *Positive selection detection.* This task aims to detect the positive natural selection from alignments of protein-coding DNA. Traditional methods assume that the non-synonymous/synonymous rate ratio at a site as a random variable that satisfies an underlying distribution. DP is used as the prior of the non-synonymous/synonymous rate ratios of site clusters [92].
- *Expressed sequence tag (EST) analysis.* It plays a crucial role in gene analysis in molecular biology, like gene identification in organisms. One aim of this task is to predict the number of new genes in a new coming sample which can benefit the experimental design and redundancy measuring of an EST library [49, 109]. PYP [109] is used to estimate the gene proportion in the EST library and the number of new genes, while the probability of discovering new genes estimated by additional samples [55].

7.5 Music analysis

Music analysis is another interesting application scenario for BNL. Such applications include teaching music, analysing the human perception of sounds, and designs for music searches [138, 169]. Music data is stored as acoustic waveforms. Among a number of methodologies for music analysis, Bayesian techniques have been found to be very effective, which has paved the way for further advancements using BNL. The following applications are highlighted.

- *Musical similarity computation.* This task is to estimate the timbral similarity between recorded songs, which could be further applied to music retrieval. HDP is used for this task through representing each song as a (stick weight) posterior distribution. The similarity between two songs can then be evaluated by the symmetrised KL divergence of two corresponding distributions. Compared to the traditional single Gaussian-based or GMM-based approaches, HDP-based approaches perform better [89]. A similar idea was also adopted by [124] with a 2-dimensional tree structure learned to represent each song.
- *Blind source separation.* This task is to separate different types of sounds, e.g., instruments or people, from an audio clip. One challenge for this task is that the number of sound sources is unknown; however, Blei et al. [6] proposed a GP-based NMF to resolve this problem.

7.6 Robots

Using BNL to teach robots how to complete specific tasks is arguably one of the most pioneering tasks in current machine learning. Some examples are given below:

- *Robot teaching.* This task aims to train robots, such as autonomous quadrotor flights or self-driving cars, to learn actions from hand-held demonstrations [24, 119]. Experts demonstrate good or perfect actions for a specific task, and the robots learn from these demonstrations through reinforcement learning (also known as imitation learning).
- *Robot object identification.* This task aims to enhance the a robot's ability to identify (often complex) objects, such as clothes [104] and tactile surfaces [38].
- *Robot navigation.* This task requires a robot to build a map, locate itself, and find a route to a target according to its sensor data, which has been modelled as a regression task using GP in [95, 132].
- *Robot introspection.* This task helps a robot understand what it is doing by identifying the actions and sub-tasks it executes [184], which makes it able to react to unstructured environments.

8 CONCLUDING REMARKS

Bayesian nonparametric learning (BNL) is becoming a hot topic in machine learning due to its unique characteristics. BNL offers a strong theoretical foundation and the ability to generate powerful models in a highly flexible setting. Starting from the basic motivations and definitions, this article reviews the latest state-of-the-art research in this field following a standard procedure of BNL. A standard procedure of BNL comprises two steps: model construction and inference. Model construction can be likened to playing with Legos, where basic stochastic processes are the bricks, and the model is built by manipulating those bricks, while model inference is a parameter adjustment procedure according to the observed data. The recent advances on both steps have been reviewed, including the popular stochastic processes and their manipulations; the sampling-based and optimisation-based inference algorithms. The major applications of BNL in machine learning, e.g., relational learning and transfer learning, and real-world tasks, e.g., biology and computer vision, have also been summarised. From this survey, we find that BNL is still in its development stage, with a large gap between the current theories and techniques and the demand for solutions that address more complicated real-world tasks. However, rather than seeing this gap as presenting challenges to current developments, we see them as a rich source of possibilities deserving of further studies in this field. A selection are highlighted below.

- *Truncation-free variational inference.* Most of the variational inference algorithms for BNL require truncation, which not only introduces errors into the posterior approximation but also discards the asymptotic property in the original Bayesian nonparametric model. Some scholars have attempted to resolve this issue [106, 175], but a general truncation-free variational inference method for Bayesian nonparametric models has yet to be developed. Integrating all the state-of-the-art models to form a unified general method of truncation would be especially significant for practical platforms.

- *Deep Bayesian nonparametric model.* Inspired by deep learning, stacking stochastic processes in more layers would pave the way for deep Bayesian nonparametric models. The challenges such an innovation would bring, in both constructing such a model and to the model's inference, represent great opportunities for further study and progress in machine learning for real-world scenarios.
- *High-dimensionality data modelling.* High-dimensionality is a common challenge in data mining and machine learning. BNL normally assumes the features of this type of data are interchangeable, which is inappropriate for some tasks. Hence, it would be interesting to study ways of building reasonable models for high-dimensional data without sacrificing too many correlations.
- *New stochastic processes and their manipulations.* BNL's powerful modelling ability depends on an abundance of stochastic processes and how those processes are manipulated. As the complexity of data increases, new and ingenious stochastic processes and manipulations will be in greater demand. One motivating example is the Hawkes process [52], which implies a relationship between the index and parameters that are independent in PP. This relationship may be valuable for some machine learning tasks, e.g., modelling the relationships between data and their labels.

ACKNOWLEDGMENTS

This work is supported by the Australian Research Council (ARC) under Discovery Grant DP170101632.

REFERENCES

[1] Amr Ahmed, Linagjie Hong, and Alexander J. Smola. 2013. Nested Chinese restaurant franchise processes: Applications to user tracking and document modeling. In *Proceedings of the 30th International Conference on International Conference on Machine Learning (ICML'13).* JMLR.org, III–1426–III–1434.

[2] David J. Aldous. 1985. *Exchangeability and Related Topics.* Springer.

[3] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. 2003. An introduction to MCMC for machine learning. *Machine Learning* 50, 1 (2003), 5–43.

[4] Cédric Archambeau, Balaji Lakshminarayanan, and Guillaume Bouchard. 2015. Latent IBP compound Dirichlet allocation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 321–333.

[5] Behnam Babagholami M., Seyed M. Roostaiyan, Ali Zarghami, and Mahdieh S. Baghshah. 2014. Multi-modal distance metric learning: A Bayesian non-parametric approach. In *Proceedings of the 13th European Conference on Computer Vision Workshops (ECCV '14).* Zurich, Switzerland, 63–77.

[6] David M. Blei, Perry R. Cook, and Matthew Hoffman. 2010. Bayesian nonparametric matrix factorization for recorded music. In *Proceedings of the 27th International Conference on Machine Learning (ICML '10).* Haifa, Israel, 439–446.

[7] David M. Blei and Peter I. Frazier. 2011. Distance dependent Chinese restaurant processes. *Journal of Machine Learning Research* 12 (2011), 2461–2488.

[8] David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. 2010. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *J. ACM* 57, 2 (2010), 7.

[9] David M. Blei and Michael I. Jordan. 2006. Variational inference for Dirichlet process mixtures. *Bayesian Analysis* 1, 1 (2006), 121–143.

[10] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *J. Amer. Statist. Assoc.* 112, 518 (2017), 859–877.

[11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.

[12] Phil Blunsom and Trevor Cohn. 2011. A hierarchical Pitman-Yor process HMM for unsupervised part of speech induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (ACL '11).* Portland, Oregon, USA, 865–874.

[13] Tamara Broderick, Michael I. Jordan, and Jim Pitman. 2012. Beta processes, stick-breaking and power laws. *Bayesian Analysis* 7, 2 (2012), 439–476.

[14] Tamara Broderick, Lester Mackey, John Paisley, and Michael I. Jordan. 2015. Combinatorial clustering and the beta negative binomial process. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 290–306.

[15] Michael Bryant and Erik B. Sudderth. 2012. Truly nonparametric online variational inference for hierarchical Dirichlet processes. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*. Lake Tahoe, Nevada, USA, 2699–2707.

[16] Trevor Campbell, Julian Straub, John W. Fisher III, and Jonathan P. How. 2015. Streaming, distributed variational inference for Bayesian nonparametrics. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS '15)*. Montreal, Quebec, Canada, 280–288.

[17] Kevin R. Canini and Thomas L. Griffiths. 2011. A nonparametric Bayesian model of multi-level category learning. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI '11)*. San Francisco, California, 307–312.

[18] Kevin R. Canini, Mikhail M. Shashkov, and Thomas L. Griffiths. 2010. Modeling transfer learning in human categorization with the hierarchical Dirichlet process. In *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*. Haifa, Israel, 151–158.

[19] Lawrence Carin, David M. Blei, and John W. Paisley. 2011. Variational inference for stick-breaking beta process priors. In *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*. Bellevue, Washington, USA, 889–896.

[20] Francois Caron, Manuel Davy, and Arnaud Doucet. 2007. Generalized polya urn for time-varying Dirichlet process mixtures. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI '07)*. Vancouver, BC, Canada, 33–40.

[21] FranÇois Caron, Manuel Davy, Arnaud Doucet, Emmanuel Duflos, and Philippe Vanheeghe. 2008. Bayesian inference for linear dynamic models with Dirichlet process mixtures. *IEEE Transactions on Signal Processing* 56, 1 (2008), 71–84.

[22] FranÇois Caron and Emily B. Fox. 2017. Sparse graphs using exchangeable random measures. *Journal of the Royal Statistical Society B* 79, 5 (2017), 1295–1366.

[23] Jason Chang and John W. Fisher III. 2013. Parallel sampling of DP mixture models using sub-cluster splits. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS '13)*. Lake Tahoe, Nevada, USA, 620–628.

[24] Sotirios P. Chatzis, Dimitrios Korkinof, and Yiannis Demiris. 2012. A nonparametric Bayesian approach toward robot learning by demonstration. *Robotics and Autonomous Systems* 60, 6 (2012), 789 – 802.

[25] Sotirios P. Chatzis and Gabriel Tsechpenakis. 2010. The infinite hidden markov random field model. *IEEE Transactions on Neural Networks* 21, 6 (2010), 1004–1014.

[26] Bo Chen, Gungor Polatkan, Guillermo Sapiro, Lawrence Carin, and David B. Dunson. 2011. The hierarchical beta process for convolutional factor analysis and deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*. Bellevue, Washington, USA, 361–368.

[27] Changyou Chen, Wray Buntine, Nan Ding, Lexing Xie, and Lan Du. 2015. Differential topic models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 230–242.

[28] Changyou Chen, Nan Ding, and Wray L. Buntine. 2012. Dependent hierarchical normalized random measures for dynamic topic modeling. In *Proceedings of the 29th International Conference on Machine Learning (ICML '12)*. Edinburgh, Scotland, UK.

[29] Changyou Chen, Vinayak Rao, Wray Buntine, and Yee W. Teh. 2013. Dependent normalized random measures. In *Proceedings of the 30th International Conference on Machine Learning (ICML '13)*. Atlanta, GA, USA, 969–977.

[30] Yi Chen, X L Wang, Xin Xiang, Buzhou Tang, and Junzhao Bu. 2015. Network structure exploration via Bayesian nonparametric models. *Journal of Statistical Mechanics: Theory and Experiment* 2015, 10 (2015), P10004.

[31] Jen-Tzung Chien. 2018. Bayesian nonparametric learning for hierarchical and sparse topics. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26, 2 (2018), 422–435.

[32] Jaedeug Choi and Kee-Eung Kim. 2012. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*. Lake Tahoe, Nevada, USA, 305–313.

[33] Jaedeug Choi and Kee-Eung Kim. 2013. Bayesian nonparametric feature construction for inverse reinforcement learning. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI '13)*. Beijing, China, 1287–1293.

[34] Yeonseung Chung and David B. Dunson. 2009. The local Dirichlet process. *Annals of the Institute of Statistical Mathematics* 63, 1 (2009), 59–80.

[35] Adelino R. Ferreira da Silva. 2007. A Dirichlet process mixture model for brain MRI tissue classification. *Medical Image Analysis* 11, 2 (2007), 169–182.

[36] Andrew M. Dai and Amos J. Storkey. 2015. The supervised hierarchical Dirichlet process. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 243–255.

[37] Patrick Dallaire, Camille Besse, Stephane Ross, and Brahim Chaib-draa. 2009. Bayesian reinforcement learning in continuous POMDPs with Gaussian processes. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '09)*. St. Louis, MO, USA, 2604–2609.

[38] Patrick Dallaire, Philippe Giguère, Daniel Émond, and Brahim Chaib-draa. 2014. Autonomous tactile perception: A combined improved sensing and Bayesian nonparametric approach. *Robotics and Autonomous Systems* 62, 4 (2014), 422–435.

[39] P. Damlen, John Wakefield, and Stephen Walker. 1999. Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61, 2 (1999), 331–344.

[40] Marc P. Deisenroth, Carl E. Rasmussen, and Jan Peters. 2008. Model-based reinforcement learning with continuous states and actions. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN '08)*. Bruges, Belgium, 19–24.

[41] John DeNero, Alexandre Bouchard-Côté, and Dan Klein. 2008. Sampling alignment structure under a Bayesian translation model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*. Honolulu, Hawaii, USA, 314–323.

[42] Nan Ding, Rongjing Xiang, Ian Molloy, and Ninghui Li. 2010. Nonparametric Bayesian matrix factorization by Power-EP. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS '10)*. Sardinia, Italy, 169–176.

[43] Kjell Doksum. 1974. Tailfree and neutral random probabilities and their posterior distributions. *Annals of Probability* 2, 2 (1974), 183–201.

[44] Finale Doshi, Kurt Miller, Jurgen V. Gael, and Yee W. Teh. 2009. Variational inference for the Indian buffet process. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS '09)*. Clearwater Beach, Florida, USA, 137–144.

[45] Finale Doshi-velez. 2009. The infinite partially observable Markov decision process. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*. Vancouver, British Columbia, Canada, 477–485.

[46] Finale Doshi-Velez, David Pfau, Frank Wood, and Nicholas Roy. 2015. Bayesian nonparametric methods for partially-observable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 394–407.

[47] Finale Doshi-Velez, David Wingate, Nicholas Roy, and Joshua B. Tenenbaum. 2010. Nonparametric Bayesian policy priors for reinforcement learning. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS '10)*. Vancouver, British Columbia, Canada, 532–540.

[48] Kumar Dubey, Sinead Williamson, and Eric P. Xing. 2014. Parallel Markov chain Monte Carlo for Pitman-Yor mixture models. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI '14)*. Quebec City, Quebec, Canada, 142–151.

[49] David B. Dunson. 2010. *Bayesian Nonparametrics.* Cambridge University Press, Chapter Nonparametric Bayes applications to biostatistics, 223–273.

[50] David B. Dunson and Ju-Hyun Park. 2008. Kernel stick-breaking processes. *Biometrika* 95, 2 (2008), 307–323.

[51] Clément Elvira, Pierre Chainais, and Nicolas Dobigeon. 2017. Bayesian nonparametric principal component analysis. *arXiv preprint arXiv:1709.05667* (2017).

[52] Paul Embrechts, Thomas Liniger, and Lu Lin. 2011. Multivariate Hawkes processes: an application to financial data. *Journal of Applied Probability* 48A (2011), 367–378.

[53] Ali Faisal, Jussi Gillberg, Gayle Leen, and Jaakko Peltonen. 2013. Transfer learning using a nonparametric sparse topic model. *Neurocomputing* 112 (2013), 124 – 137. Advances in artificial neural networks, machine learning, and computational intelligenceSelected papers from the 20th European Symposium on Artificial Neural Networks (2012).

[54] Xuhui Fan, Longbing Cao, and Richard D.Y. Xu. 2015. Dynamic infinite mixed-membership stochastic blockmodel. *IEEE Transactions on Neural Networks and Learning Systems* 26, 9 (2015), 2072–2085.

[55] Stefano Favaro, Antonio Lijoi, and Igor Prünster. 2012. A new estimator of the discovery probability. *Biometrics* 68, 4 (2012), 1188–1196.

[56] Paul Fearnhead. 2004. Particle filters for mixture models with an unknown number of components. *Statistics and Computing* 14, 1 (2004), 11–21.

[57] Thomas S. Ferguson. 1973. A Bayesian analysis of some nonparametric problems. *Annals of Statistics* 1, 2 (1973), 209–230.

[58] Thomas S. Ferguson, Eswar G. Phadia, and Ram C. Tiwari. 1992. Bayesian nonparametric inference. *Lecture Notes-Monograph Series* 17 (1992), 127–150.

[59] Nicholas J. Foti, Joseph D. Futoma, Daniel N. Rockmore, and Sinead Williamson. 2013. A unifying representation for a class of dependent random measures. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS '13)*. Scottsdale, AZ, USA, 20–28.

[60] Nicholas J. Foti and Sinead A. Williamson. 2015. A survey of non-exchangeable priors for bayesian nonparametric models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 359–371.

[61] Emily Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. 2011. Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing* 59, 4 (2011), 1569–1585.

[62] Emily B. Fox. 2009. *Bayesian Nonparametric Learning of Complex Dynamical Phenomena.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[63] Emily B. Fox, Michael C. Hughes, Erik B. Sudderth, and Michael I. Jordan. 2014. Joint modeling of multiple time series via the beta process with application to motion capture segmentation. *Annals of Applied Statistics* 8, 3 (2014), 1281–1313.

[64] Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. 2011. A sticky HDP-HMM with application to speaker diarization. *Annals of Applied Statistics* 5, 2A (2011), 1020–1056.

[65] Jurgen V. Gael, Yee W. Teh, and Zoubin Ghahramani. 2008. The infinite factorial hidden Markov model. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS '08)*. Vancouver, British Columbia, Canada, 1697–1704.

[66] Jurgen V. Gael, Andreas Vlachos, and Zoubin Ghahramani. 2009. The infinite HMM for unsupervised PoS tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '09)*. Singapore, 678–687.

[67] Zekai Gao, Yangqiu Song, Shixia Liu, Haixun Wang, Hao Wei, Yang Chen, and Weiwei Cui. 2011. Tracking and connecting topics via incremental hierarchical Dirichlet processes. In *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM '11)*. Vancouver, Canada, 1056–1061.

[68] Hong Ge, Yutian Chen, Moquan Wan, and Zoubin Ghahramani. 2015. Distributed inference for Dirichlet process mixture models. In *Proceedings of the 32nd International Conference on Machine Learning (ICML '15)*. Lille, France, 2276–2284.

[69] Alan E. Gelfand, Athanasios Kottas, and Steven N. MacEachern. 2005. Bayesian nonparametric spatial modeling with Dirichlet process mixing. *J. Amer. Statist. Assoc.* 100, 471 (2005), 1021–1035.

[70] Samuel J. Gershman and David M. Blei. 2012. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology* 56, 1 (2012), 1 – 12.

[71] Samuel J. Gershman, Peter I. Frazier, and David M. Blei. 2015. Distance dependent infinite latent feature models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 334–345.

[72] Zoubin Ghahramani, Michael I. Jordan, and Ryan P. Adams. 2010. Tree-structured stick breaking for hierarchical data. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS '10)*. Vancouver, British Columbia, Canada, 19–27.

[73] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al. 2015. Bayesian reinforcement learning: A survey. *Foundations and Trends in Machine Learning* 8, 5-6 (2015), 359–483.

[74] Jayanta K. Ghosh and R.V. Ramamoorthi. 2002. *Bayesian Nonparametrics.* Springer.

[75] Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2009. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition* 112, 1 (2009), 21 – 54.

[76] Jim E. Griffin, Michalis Kolossiatis, and Mark F.J. Steel. 2013. Comparing distributions by using dependent normalized random-measure mixtures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 75, 3 (2013), 499–529.

[77] Thomas L. Griffiths and Zoubin Ghahramani. 2005. Infinite latent feature models and the Indian buffet process. In *Proceedings of the 19th Annual Conference on Neural Information Processing Systems (NIPS '05)*. Vancouver, British Columbia, Canada, 475–482.

[78] Thomas L. Griffiths and Zoubin Ghahramani. 2011. The indian buffet process: An introduction and review. *Journal of Machine Learning Research* 12 (2011), 1185–1224.

[79] Sunil K. Gupta, Dinh Phung, and Svetha Venkatesh. 2012. A Bayesian nonparametric joint factor model for learning shared and individual subspaces from multiple data sources. In *Proceedings of the 12th International Conference on Data Mining (SDM '12)*. Anaheim, California, USA, 200–211.

[80] Sunil K. Gupta, Dinh Q. Phung, and Svetha Venkatesh. 2012. A slice sampler for restricted hierarchical beta process with applications to shared subspace learning. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI '12)*. Catalina Island, CA, USA, 316–325.

[81] Tom S. F. Haines and Tao Xiang. 2014. Background subtraction with Dirichlet process mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 4 (2014), 670–683.

[82] Lauren A. Hannah, David M. Blei, and Warren B. Powell. 2011. Dirichlet process mixtures of generalized linear models. *Journal of Machine Learning Research* 12 (2011), 1923–1953.

[83] Li He, Hairong Qi, and Russell Zaretzki. 2013. Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*. Portland, OR, USA, 345–352.

[84] Creighton Heaukulani, David A. Knowles, and Zoubin Ghahramani. 2014. Beta diffusion trees. In *Proceedings of the 31th International Conference on Machine Learning (ICML '14)*. Beijing, China, 1809–1817.

[85] Jennifer L. Hill. 2011. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics* 20, 1 (2011), 217–240.

[86] Geoffrey E. Hinton, Simon Osindero, and Yee W. Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 7 (2006), 1527–1554.

[87] Nils L. Hjort. 1990. Nonparametric Bayes estimators based on beta processes in models for life history data. *Annals of Statistics* 18, 3 (1990), 1259–1294.

[88] Nils L. Hjort, Chris Holmes, Peter Müller, and Stephen G. Walker. 2010. *Bayesian Nonparametrics*. Vol. 28. Cambridge University Press.

[89] Matthew D. Hoffman, David M. Blei, and Perry R. Cook. 2008. Content-based musical similarity computation using the hierarchical Dirichlet process. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR '08)*. Philadelphia, PA, USA, 349–354.

[90] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *Journal of Machine Learning Research* 14, 1 (2013), 1303–1347.

[91] Yuening Hu, Ke Zhai, Vladimir Eidelman, and Jordan L. Boyd-Graber. 2014. Polylingual tree-based topic models for translation domain adaptation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL '14)*. Baltimore, MD, USA, 1166–1176.

[92] John P. Huelsenbeck, Sonia Jain, Simon W.D. Frost, and Sergei L. Kosakovsky Pond. 2006. A Dirichlet process model for detecting positive selection in protein-coding DNA sequences. *Proceedings of the National Academy of Sciences* 103, 16 (2006), 6263–6268.

[93] Tomoharu Iwata, James R. Lloyd, and Zoubin Ghahramani. 2016. Unsupervised many-to-many object matching for relational data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 3 (2016), 607–617.

[94] Saad Jbabdi, Mark Woolrich, and Timothy E.J. Behrens. 2009. Multiple-subjects connectivity-based parcellation using hierarchical Dirichlet process mixture models. *NeuroImage* 44, 2 (2009), 373 – 384.

[95] Yun Jiang and Ashutosh Saxena. 2013. Infinite latent conditional random fields for modeling environments through humans. In *Proceedings of Robotics: Science and Systems IX*. Berlin, Germany.

[96] Michael I. Jordan. 2010. Bayesian nonparametric learning: Expressive priors for intelligent systems. *Heuristics, Probability and Causality: A Tribute to Judea Pearl* 11 (2010), 167–185.

[97] Maria Kalli, Jim E. Griffin, and Stephen G. Walker. 2011. Slice sampling mixture models. *Statistics and Computing* 21, 1 (2011), 93–105.

[98] Jeon-Hyung Kang, Jun Ma, and Yan Liu. 2012. Transfer topic modeling with ease and scalability. In *Proceedings of the 12th SIAM International Conference on Data Mining (SDM '12)*. Anaheim, California, USA, 564–575.

[99] Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. 2006. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1 (AAAI '06)*. Boston, Massachusetts, USA, 381–388.

[100] John F. C. Kingman. 1967. Completely random measures. *Pacific J. Math.* 21, 1 (1967), 59–78.

[101] John F. C. Kingman. 1982. The coalescent. *Stochastic Processes and their Applications* 13, 3 (1982), 235 – 248.

[102] John F. C. Kingman. 1992. *Poisson Processes*. Vol. 3. Oxford university press.

[103] David A. Knowles and Zoubin Ghahramani. 2015. Pitman Yor diffusion trees for Bayesian hierarchical clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 271–289.

[104] Nishanth Koganti, Tomoya Tamei, Kazushi Ikeda, and Tomohiro Shibata. 2017. Bayesian Nonparametric Learning of Cloth Models for Real-Time State Estimation. *IEEE Transactions on Robotics* 33, 4 (2017), 916–931.

[105] Kenichi Kurihara, Max Welling, and Yee W. Teh. 2007. Collapsed variational Dirichlet process mixture models. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI '07)*, Vol. 7. Hyderabad, India, 2796–2801.

[106] Kenichi Kurihara, Max Welling, and Nikos A. Vlassis. 2006. Accelerated variational Dirichlet process mixtures. In *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS '06)*. Vancouver, British Columbia, Canada, 761–768.

[107] Sergey Levine, Zoran Popovic, and Vladlen Koltun. 2011. Nonlinear inverse reinforcement learning with Gaussian processes. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS '11)*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger (Eds.). Granada, Spain, 19–27.

[108] Dawen Liang, Matthew D. Hoffman, and Daniel P.W. Ellis. 2013. Beta process sparse nonnegative matrix factorization for music.. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR '13)*. Curitiba, Brazil, 375–380.

[109] Antonio Lijoi, Ramsés H. Mena, and Igor Prünster. 2007. A Bayesian nonparametric method for prediction in EST analysis. *BMC Bioinformatics* 8, 1 (2007), 1–10.

[110] Kar W. Lim, Wray Buntine, Changyou Chen, and Lan Du. 2016. Nonparametric Bayesian topic modelling with the hierarchical Pitman-Yor processes. *International Journal of Approximate Reasoning* 78, C (2016), 172–191.

[111] Dahua Lin. 2013. Online learning of nonparametric mixture models via sequential variational approximation. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS '13)*. Lake Tahoe, Nevada, USA, 395–403.

[112] Dahua Lin, Eric Grimson, and John W. Fisher III. 2010. Construction of dependent Dirichlet processes based on Poisson processes. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS '10)*. Vancouver, British Columbia, Canada, 1396–1404.

[113] Dan Lovell, Jonathan Malmaud, Ryan P. Adams, and Vikash K. Mansinghka. 2013. ClusterCluster: Parallel Markov Chain Monte Carlo for Dirichlet Process Mixtures. *arXiv preprint arXiv:1304.2302* (2013).

[114] Steven N. MacEachern. 1999. Dependent nonparametric processes. In *Proceedings of the section on Bayesian statistical science*. American Statistical Association, 50–55.

[115] Steven N. MacEachern. 2000. *Dependent Dirichlet Processes.* Technical Report. Department of Statistics, The Ohio State University.

[116] Steven N. MacEachern, Merlise Clyde, and Jun S. Liu. 1999. Sequential importance sampling for nonparametric Bayes models: The next generation. *Canadian Journal of Statistics* 27, 2 (1999), 251–267.

[117] M. Mahmud. 2010. Constructing states for reinforcement learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*. Haifa, Israel, 727–734.

[118] Bernard Michini and Jonathan P. How. 2012. Bayesian nonparametric inverse reinforcement learning. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML '12)*. Bristol, UK, 148–163.

[119] Bernard Michini, Thomas J. Walsh, Ali-Akbar Agha-Mohammadi, and Jonathan P. How. 2015. Bayesian nonparametric reward learning from demonstration. *IEEE Transactions on Robotics* 31, 2 (2015), 369–386.

[120] Kurt Miller, Michael I. Jordan, and Thomas L. Griffiths. 2009. Nonparametric latent feature models for link prediction. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*. Vancouver, British Columbia, Canada, 1276–1284.

[121] Thomas Minka. 2004. *Power EP.* Technical Report. Microsoft Research, Cambridge.

[122] Morten Mørup, Mikkel N. Schmidt, and Lars K. Hansen. 2011. Infinite multiple membership relational modeling for complex networks. In *Proceedings of the 21st IEEE International Workshop on Machine Learning for Signal Processing (MLSP '11)*. Beijing, China, 1–6.

[123] Peter Müller, Fernando A. Quintana, Alejandro Jara, and Tim Hanson. 2015. *Bayesian Nonparametric Data Analysis.* Springer.

[124] Masahiro Nakano, Yasunori Ohishi, Hirokazu Kameoka, Ryo Mukai, and Kunio Kashino. 2012. Bayesian nonparametric music parser. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '12)*. Kyoto, Japan, 461–464.

[125] Radford M. Neal. 2003. Density modeling and clustering using Dirichlet diffusion trees. *Bayesian Statistics* 7 (2003), 619–629.

[126] Radford M. Neal. 2003. Slice sampling. *Annals of Statistics* 31, 3 (2003), 705–767.

[127] Willie Neiswanger, Chong Wang, and Eric Xing. 2014. Embarrassingly parallel variational inference in nonconjugate models. In *Workshop on Advanced Variational Inference, Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPSW '14)*. Montreal, Quebec, Canada, 1–18.

[128] Peter Orbanz and Yee Whye Teh. 2010. *Encyclopedia of Machine Learning*. Springer US, Boston, MA, Chapter Bayesian nonparametric models, 81–89.

[129] John Paisley, Chong Wang, David M. Blei, and Michael I. Jordan. 2015. Nested hierarchical Dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 256–270.

[130] Konstantina Palla, David A. Knowles, and Zoubin Ghahramani. 2015. Relational learning and network modelling using infinite latent attribute models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 462–474.

[131] Sinno J. Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.

[132] Christian Plagemann, Kristian Kersting, Patrick Pfaff, and Wolfram Burgard. 2007. Gaussian beam processes: A nonparametric Bayesian measurement model for range finders. In *Proceedings of the Robotics: Science and Systems*. Atlanta, Georgia, USA.

[133] S. C. Williams R. Daniel Mauldin, William D. Sudderth. 1992. Polya trees and random distributions. *Annals of Statistics* 20, 3 (1992), 1203–1221.

[134] Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286.

[135] Natraj Raman and S.J. Maybank. 2016. Non-parametric hidden conditional random fields for action classification. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN '16)*. 3256–3263.

[136] Pravesh Ranchod, Benjamin Rosman, and George Konidaris. 2015. Nonparametric Bayesian reward segmentation for skill discovery using inverse reinforcement learning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '15)*. Hamburg, Germany, 471–477.

[137] Carl E. Rasmussen. 1999. The infinite Gaussian mixture model. In *Proceedings of the 13th Annual Conference on Neural Information Processing Systems (NIPS '99)*. Denver, CO, USA, 554–560.

[138] Lu Ren, David Dunson, Scott Lindroth, and Lawrence Carin. 2010. Dynamic nonparametric Bayesian models for analysis of music. *J. Amer. Statist. Assoc.* 105, 490 (2010), 458–472.

[139] Lu Ren, David B. Dunson, and Lawrence Carin. 2008. The dynamic hierarchical Dirichlet process. In *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*. Helsinki, Finland, 824–831.

[140] Lu Ren, Yingjian Wang, Lawrence Carin, and David B. Dunson. 2011. The kernel beta process. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS '11)*. Granada, Spain, 963–971.

[141] Abel Rodriguez, David B. Dunson, and Alan E. Gelfand. 2008. The nested Dirichlet process. *J. Amer. Statist. Assoc.* 103, 483 (2008), 1131–1154.

[142] Daniel M. Roy and Leslie Pack Kaelbling. 2007. Efficient Bayesian task-level transfer learning.. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI '07)*, Vol. 7. Hyderabad, India, 2599–2604.

[143] Daniel M. Roy and Yee W. Teh. 2008. The mondrian process. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS '08)*. Vancouver, British Columbia, Canada, 1377–1384.

[144] Jason Roy, Kirsten J. Lum, Michael J. Daniels, Bret Zeldow, Jordan Dworkin, and Vincent Lo Re III. 2017. Bayesian nonparametric generative models for causal inference with missing at random covariates. *arXiv preprint arXiv:1702.08496* (2017).

[145] Anirban Roychowdhury and Brian Kulis. 2015. Gamma processes, stick-breaking, and variational inference. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS '15)*. San Diego, California, USA, 800–808.

[146] Ruslan Salakhutdinov, Joshua B. Tenenbaum, and Antonio Torralba. 2011. One-shot learning with a hierarchical nonparametric Bayesian model. In *Workshop on Unsupervised and Transfer Learning - Proceedings of the 28th International Conference on Machine Learning (ICMLW '11)*. Bellevue, Washington, USA, 195–206.

[147] Issei Sato, Kenichi Kurihara, and Hiroshi Nakagawa. 2012. Practical collapsed variational bayes inference for hierarchical dirichlet process. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '12)*. Beijing, China, 105–113.

[148] Ken-iti Sato. 1999. *Lévy Processes and Infinitely Divisible Distributions*. Cambridge University Press.

[149] Mikkel N. Schmidt and Morten Morup. 2013. Nonparametric Bayesian modeling of complex networks: an introduction. *IEEE Signal Processing Magazine* 30, 3 (2013), 110–128.

[150] Peter Schulam and Suchi Saria. 2017. Reliable decision support using counterfactual models. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS '17)*. Long Beach, CA, USA, 1697–1708.

[151] Matthias Seeger. 2004. Gaussian processes for machine learning. *International Journal of Neural Systems* 14, 02 (2004), 69–106.

[152] Jayaram Sethuraman. 1994. A constructive definition of Dirichlet priors. *Statistica Sinica* 4, 2 (1994), 639–650.

[153] Babak Shahbaba and Radford Neal. 2009. Nonlinear models using Dirichlet process mixtures. *Journal of Machine Learning Research* 10 (2009), 1829–1850.

[154] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354.

[155] Padhraic Smyth, Max Welling, and Arthur U. Asuncion. 2009. Asynchronous distributed learning of topic models. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*. Vancouver, British Columbia, Canada, 81–88.

[156] Nitish Srivastava and Ruslan R. Salakhutdinov. 2013. Discriminative transfer learning with tree-based priors. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS '13)*. Lake Tahoe, Nevada, United States, 2094–2102.

[157] Jacob Steinhardt and Zoubin Ghahramani. 2012. Flexible martingale priors for deep hierarchies. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS '12)*. La Palma, Canary Islands, 1108–1116.

[158] Amit Surana and Kunal Srivastava. 2014. Bayesian nonparametric inverse reinforcement learning for switched Markov decision processes. In *Proceedings of the 13th IEEE International Conference on Machine Learning and Applications (ICMLA '14)*. Detroit, MI, USA, 47–54.

[159] Alex Tank, Nicholas Foti, and Emily Fox. 2015. Streaming variational inference for Bayesian nonparametric mixture models. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS '15)*. San Diego, California, USA, 968–976.

[160] Martin A. Tanner and Wing H. Wong. 2010. From EM to data augmentation: The emergence of MCMC Bayesian computation in the 1980s. *Statist. Sci.* 25, 4 (2010), 506–516.

[161] Yee W. Teh. 2006. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ACL '06)*. Stroudsburg, PA, USA, 985–992.

[162] Yee W. Teh. 2010. *Encyclopedia of Machine Learning*. Springer US, Chapter Dirichlet process, 280–287.

[163] Yee W. Teh, Charles Blundell, and Lloyd Elliott. 2011. Modelling genetic variations using fragmentation-coagulation processes. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS '11)*. Granada, Spain, 819–827.

[164] Yee W. Teh, Dilan Görür, and Zoubin Ghahramani. 2007. Stick-breaking construction for the Indian buffet process. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS '07)*. San Juan, Puerto Rico, 556–563.

[165] Yee W. Teh, Hal Daume III, and Daniel M. Roy. 2007. Bayesian agglomerative clustering with coalescents. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS '07)*. Vancouver, British Columbia, Canada, 1473–1480.

[166] Yee W. Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2005. Sharing clusters among related groups: Hierarchical Dirichlet processes. In *Proceedings of the 19th Annual Conference on Neural Information Processing Systems (NIPS '05)*. Vancouver, British Columbia, Canada, 1385–1392.

[167] Yee W. Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical Dirichlet processes. *J. Amer. Statist. Assoc.* 101, 476 (2006), 1566–1581.

[168] Yee W. Teh, Kenichi Kurihara, and Max Welling. 2007. Collapsed variational inference for HDP. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS '07)*. Vancouver, British Columbia, Canada, 1481–1488.

[169] David Temperley. 2007. *Music and Probability*. MIT Press.

[170] Romain Thibaux and Michael I. Jordan. 2007. Hierarchical beta processes and the Indian buffet process. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS '07)*. San Juan, Puerto Rico, 564–571.

[171] Bruce Thompson. 2004. *Exploratory and Confirmatory Factor Analysis: Understanding Concepts and Applications*. American Psychological Association.

[172] David A Van Dyk and Xiao-Li Meng. 2001. The art of data augmentation. *Journal of Computational and Graphical Statistics* 10, 1 (2001), 1–50.

[173] Sara Wade, Silvia Mongelluzzo, and Sonia Petrone. 2011. An enriched conjugate prior for Bayesian nonparametric inference. *Bayesian Analysis* 6, 3 (2011), 359–385.

[174] Chong Wang and David M. Blei. 2009. Variational inference for the nested Chinese restaurant process. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*. Vancouver, British Columbia, Canada, 1990–1998.

[175] Chong Wang and David M. Blei. 2012. Truncation-free online variational inference for Bayesian nonparametric models. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*. Lake Tahoe, Nevada, USA, 413–421.

[176] Chong Wang, John W. Paisley, and David M. Blei. 2011. Online variational inference for the hierarchical Dirichlet process. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS '11)*. Fort Lauderdale, USA, 752–760.

[177] Yingjian Wang and Lawrence Carin. 2012. LéVy measure decompositions for the beta and gamma processes. In *Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML '12)*. Edinburgh, Scotland, UK, 499–506.

[178] Christopher K.I. Williams and Carl Edward Rasmussen. 2006. *Gaussian Processes for Machine Learning*. Vol. 2. MIT Press. 4 pages.

[179] Sinead Williamson, Avinava Dubey, and Eric Xing. 2013. Parallel Markov chain Monte Carlo for nonparametric mixture models. In *Proceedings of the 30th International Conference on Machine Learning (ICML '13)*. Atlanta, GA, USA, 98–106.

[180] Sinead A. Williamson. 2016. Nonparametric network models for link prediction. *Journal of Machine Learning Research* 17, 1 (2016), 7102–7121.

[181] Alan S. Willsky, Erik B. Sudderth, Michael I. Jordan, and Emily B. Fox. 2009. Nonparametric Bayesian learning of switching linear dynamical systems. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*. Vancouver, British Columbia, Canada, 457–464.

[182] Frank Wood and Thomas L. Griffiths. 2006. Particle filtering for nonparametric Bayesian matrix factorization. In *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS '06)*. Vancouver, British Columbia, Canada, 1513–1520.

[183] Frank Wood and Yee W. Teh. 2009. A hierarchical nonparametric Bayesian approach to statistical language model domain adaptation. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS '09)*. Clearwater Beach, Florida, USA, 607–614.

[184] Hongmin Wu, Juan Rojas, Hongbin Lin, and Kensuke Harada. 2017. Robot introspection with bayesian nonparametric vector autoregressive hidden markov models. *arXiv preprint arXiv:1705.08661* abs/1705.08661 (2017). arXiv:1705.08661

[185] Tianbing Xu, Zhongfei Zhang, Philip S. Yu, and Bo Long. 2008. Evolutionary clustering by hierarchical Dirichlet process with hidden Markov state. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08)*. Pisa, Italy, 658–667.

[186] Zhao Xu, Volker Tresp, Achim Rettinger, and Kristian Kersting. 2010. Social network mining with nonparametric relational models. In *Proceedings of the 2nd International Workshop on Advances in Social Network Mining and Analysis (SNAKDD '08)*. Las Vegas, NV, USA, 77–96.

[187] Zhao Xu, Volker Tresp, Kai Yu, and Hans-Peter Kriegel. 2006. Infinite hidden relational models. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI '06)*. Arlington, Virginia, United States, 544–551.

[188] Zhao Xu, Volker Tresp, Kai Yu, Shipeng Yu, and Hans-Peter Kriegel. 2005. Dirichlet enhanced relational learning. In *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*. New York, NY, USA, 1004–1011.

[189] Zenglin Xu, Feng Yan, and Yuan Qi. 2015. Bayesian nonparametric models for multiway data analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 475–487.

[190] Junyu Xuan, Jie Lu, Guangquan Zhang, Richard Y.D. Xu, and Xiangfeng Luo. 2015. Infinite author topic model based on mixed gamma-negative binomial process. In *Proceedings of the 15th IEEE International Conference on Data Mining (ICDM '15)*. Atlantic City, New Jersey, USA, 489–498.

[191] Junyu Xuan, Jie Lu, Guangquan Zhang, Richard Y.D. Xu, and Xiangfeng Luo. 2017. Bayesian nonparametric relational topic model through dependent Gamma processes. *IEEE Transactions on Knowledge and Data Engineering* 29, 7 (2017), 1357–1369.

[192] Junyu Xuan, Jie Lu, Guangquan Zhang, Richard Y.D. Xu, and Xiangfeng Luo. 2018. Doubly nonparametric sparse nonnegative matrix factorization based on dependent Indian buffet processes. *IEEE Transactions on Neural Networks and Learning Systems* 29, 5 (2018), 1835–1849.

[193] Cheng Zhang, Carl Henrik Ek, Xavi Gratal, Florian T. Pokorny, and Hedvig Kjellstrom. 2013. Supervised hierarchical Dirichlet processes with variational inference. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW '13)*. Sydney, Australia, 254–261.

[194] Jianwen Zhang, Yangqiu Song, Changshui Zhang, and Shixia Liu. 2010. Evolutionary hierarchical Dirichlet processes for multiple correlated time-varying corpora. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*. Washington, DC, USA, 1079–1088.

[195] Jiangchuan Zheng, Siyuan Liu, and Lionel M. Ni. 2014. Effective mobile context pattern discovery via adapted hierarchical Dirichlet processes. In *Proceedings of the 15th IEEE International Conference on Mobile Data Management (MDM '14)*, Vol. 1. Brisbane, Australia, 146–155.

[196] Mingyuan Zhou and Lawrence Carin. 2012. Augment-and-conquer negative binomial processes. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*. Lake Tahoe, Nevada, USA, 2546–2554.

[197] Mingyuan Zhou and Lawrence Carin. 2015. Negative binomial process count and mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 2 (2015), 307–320.

[198] Mingyuan Zhou, Haojun Chen, Lu Ren, Guillermo Sapiro, Lawrence Carin, and John W. Paisley. 2009. Non-parametric Bayesian dictionary learning for sparse image representations. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*. Vancouver, British Columbia, Canada, 2295–2303.

[199] Mingyuan Zhou, Yulai Cong, and Bo Chen. 2015. Gamma belief networks. *arXiv preprint arXiv:1512.03081* (2015).

[200] Mingyuan Zhou, Lauren Hannah, David B. Dunson, and Lawrence Carin. 2012. Beta-negative binomial process and poisson gactor snalysis. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS '12)*. La Palma, Canary Islands, 1462–1471.

[201] Mingyuan Zhou, Hongxia Yang, Guillermo Sapiro, David B. Dunson, and Lawrence Carin. 2011. Dependent hierarchical beta process for image interpolation and denoising. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI '11)*. Barcelona, Catalonia, Spain, 883–891.

[202] Jun Zhu, Ning Chen, and Eric P. Xing. 2011. Infinite latent SVM for classification and multi-task learning. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS '11)*. Granada, Spain, 1620–1628.