

Bayesian Nonparametric Space Partitions: A Survey

Xuhui Fan¹, Bin Li², Ling Luo³, Scott A. Sisson¹

¹UNSW Data Science Hub, and School of Mathematics and Statistics, University of New South Wales

²Shanghai Key Laboratory of IIP, School of Computer Science, Fudan University

³School of Computing and Information Systems, University of Melbourne

{xuhui.fan, scott.sisson}@unsw.edu.au, libin@fudan.edu.cn, ling.luo@unimelb.edu.au

Abstract

Bayesian nonparametric space partition (BNSP) models provide a variety of strategies for partitioning a D -dimensional space into a set of blocks, such that the data within the same block share certain kinds of homogeneity. BNSP models are applicable to many areas, including regression/classification trees, random feature construction, and relational modelling. This survey provides the first comprehensive review of this subject. We explore the current progress of BNSP research through three perspectives: (1) *Partition strategies*, where we review the various techniques for generating partitions and discuss their theoretical foundation – self-consistency; (2) *Applications*, where we detail the current mainstream usages of BNSP models and identify some potential future applications; and (3) *Challenges*, where we discuss current unsolved problems and possible avenues for future research.

1 Introduction

Describing and understanding implicit and complex relationships between different features (covariates) in a dataset is typically a key modelling component of many advanced statistical and machine learning methods. Among these approaches, Bayesian Nonparametric Space Partition (BNSP) models provide a flexible and geometrically interpretable way to describe these relationships. The main idea underlying BNSP models is to divide a D -dimensional ($D \geq 2$) space into a number of ‘blocks’, using a specified partition strategy, where the blocks are chosen such that the datapoints within each block exhibit certain types of homogeneity. The resolution of the partition can be arbitrarily fine (i.e. an arbitrary number of blocks in the space), perhaps as a function of the number of datapoints and the complexity of the relationship between the features, resulting in a flexible and adaptive nonparametric model able to describe complex data well. Throughout, we define the random BNSP object, \boxplus , as a set of blocks, $\boxplus = \{\square_k\}_{k \in \mathbb{N}^+}$, where \square_k denotes the k -th block in the space. The space itself is commonly specified as $[0, 1]^D$ with the observed data being rescaled to this domain.

As an illustration, consider the application of regression trees to credit risk assessment: the data $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ are ob-

served on N individuals, where \mathbf{x}_n denotes individual n ’s attributes (e.g. age, monthly expense, monthly income) and y_n her credit risk score. In this setting, one block \square_k may be constructed as: [20, 25] years old \times [\$2000, \$3000] monthly expense \times [\$4000, \$4500] monthly income. We have $\mathbf{x}_n \in \square_k$ if the n -th person belongs to the k -th block \square_k . Individuals located in block \square_k are assumed to share similar behaviours in their risk scores.

To construct regression trees based on BNSP models, an intensity variable ω_k is usually associated with each block \square_k , such that \square_k contributes an impact with intensity ω_k to the labels y_n of all $\mathbf{x}_n \in \square_k$. If we use a Gaussian distribution to describe the credit risk for individual n , the generative process could be written as:

$$\begin{aligned} (1) \quad & \{\square_k\}_k \sim \text{BNSP}([0, 1]^D, -) \\ (2) \quad & \{\omega_k\}_k \sim \mathcal{N}(0, \delta^2) \\ (3) \quad & y_n \sim \mathcal{N}\left(\sum_k \omega_k \cdot \mathbf{1}_{\mathbf{x} \in \square_k}(\mathbf{x}_n), \sigma^2\right) \end{aligned}$$

where $\mathbf{1}_{\mathbf{x} \in \square_k}(\mathbf{x}_n) = 1$ if $\mathbf{x}_n \in \square_k$; otherwise $\mathbf{1}_{\mathbf{x} \in \square_k}(\mathbf{x}_n) = 0$. Step (1) generates the blocks from a BNSP model (with additional parameters denoted by ‘-’) on the space spanned by the (rescaled) feature data $\{\mathbf{x}_n\}_{n=1}^N$. This step is the primary research focus for BNSP models – *the investigation of efficient ways to generate meaningful blocks in the space*. Step (2) generates the intensity values $\{\omega_k\}_k$ for all blocks from a common Gaussian distribution. Step (3) generates the label data from a Gaussian distribution, with mean given by the sum of intensities of all blocks covering \mathbf{x}_n , and with error variance σ^2 . Through posterior inference on these random variables (e.g., the blocks $\{\square_k\}_k$, intensities $\{\omega_k\}_k$, and error variance σ^2), which is typically implemented using Markov chain Monte Carlo (MCMC) methods, insightful structural information can be uncovered.

With different settings of the likelihood function for the label data, regression trees can be built for either regression or classification tasks. E.g., the Bernoulli distribution can be used as the likelihood function, with probability given by the logistic transform of the sum of intensities. Similarly a categorical distribution can be used, with the probability given by a multinomial logistic transform of the sum of intensities.

In contrast to the Dirichlet process [Ferguson, 1973] and its explicit view, the stick-breaking process [Sethuraman,

1994], which are often applied only in one-dimensional-space problem settings, BNSP models are typically implemented in multi-dimensional spaces. For particular definitions of the space and modelling objectives, BNSP models have been successfully implemented in many real-world applications. For example, when the space is spanned by data features and the modelling objective is the labels attached to those features, BNSP models can be applied to regression/classification trees [Chipman *et al.*, 2010; Lakshminarayanan *et al.*, 2014] and online learning [Lakshminarayanan *et al.*, 2014; Lakshminarayanan *et al.*, 2016; Mourtada *et al.*, 2017; Fan *et al.*, 2020]; when the space is spanned by the communities in a social network and the modelling objective is the linkages between the nodes in the network, BNSP models can be implemented in relational modelling [Kemp *et al.*, 2006; Roy and Teh, 2009; Nakano *et al.*, 2014a] and community detection [Nowicki and Snijders, 2001]. Other examples of applications include random feature construction [Balog *et al.*, 2016], voice recognition [Nakano *et al.*, 2014b], co-clustering [Wang *et al.*, 2011], and matrix permutation approximation [Kuck *et al.*, 2019]. Some potential application areas of BNSP models are spatial-temporal modelling, and image detection and segmentation.

In this survey, we will explore the following aspects of BNSP models: (1) *Partition strategies*, where we first introduce an important property in defining the model – self-consistency – and then review the various strategies that have been developed for generating the partitions in the space. These include regular-grid partitions, hierarchical partitions (axis-aligned cuts), hierarchical partitions (sloped cuts), floorplan partitions, and rectangular bounding partitions. We also summarise and contrast the characteristics of each approach as no single strategy will dominate the others in all cases; (2) *Applications*, where we demonstrate how to apply BNSP models to the real-world studies of online learning, random feature construction and relational modelling; (3) *Challenges*, where we discuss the current challenges in BNSP research and avenues for future work, including scalable inference methods, partition flexibility, posterior concentration analysis and analysis of deep neural networks.

2 Partition Strategies

The partition strategies for BNSP models can be divided into five categories. Illustrations of particular implementations of these in 2-dimensional space are shown in Figure 2. A critical aspect underpinning BNSP processes is the concept of *self-consistency* of the process. We review this below.

Self-consistency: The main idea behind self-consistency is to ensure that the probability of any partition remains invariant under the expansion or shrinkage of the space.

More formally, suppose we have a partition \boxplus_Y of a space $Y \in \mathcal{F}(\mathbb{R}^D)$, generated from a BNSP process, where $\mathcal{F}(\mathbb{R}^D)$ denotes the set of all finite rectangular boxes in \mathbb{R}^D . When restricting the BNSP process to a sub-space X , where $X \subset Y$, the resulting partition of \boxplus_Y restricted to X should be distributed as if it were generated via the BNSP process working on X directly. Figure 1 illustrates the self-consistency of the Rectangular Bounding Process (RBP) [Fan *et al.*, 2018b],

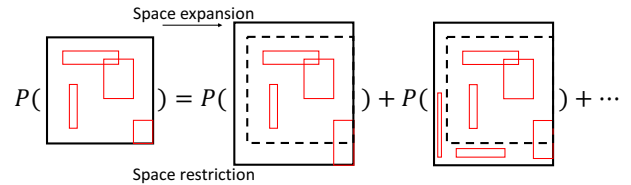


Figure 1: Illustration of self-consistency of the Rectangular Bounding Process (RBP). The solid black boxes denote 2-dimensional spaces; the red boxes denote different blocks generated by the RBP. The *lhs* of the equality denotes the probability of an RBP partition in a smaller space $X \subset Y$, and the *rhs* shows the probabilities of an infinite number of possible partitions expanded to a larger space Y from X (i.e. the *lhs* probability is obtained by marginalizing out all possible partitions in the expanded region Y/X).

which will be reviewed in detail below (Section 2.5). A BNSP process is defined to be self-consistent if and only if the probability of a partition on X equals to the sum of probabilities of all possible partitions on Y extended from the partition on X . Mathematically, this can be represented as $P_{\boxplus}^Y(\pi_{Y,X}^{-1}(\boxplus_X)) = P_{\boxplus}^X(\boxplus_X)$, where $\pi_{Y,X}$ denotes the restriction operator from Y to X . For continuous spaces, the number of possible extensions is usually uncountably infinite.

The self-consistency property ensures that the partition of a smaller space can be safely extended to larger spaces. Online learning is a typical scenario in which self-consistency is essential. When new data points are observed outside the range of the current data (i.e the minimum bounding box of all existing data points, X), the BNSP sample can immediately be extended to include these new data points within a new bounding box (Y). Further, if we make use of the Kolmogorov Extension theorem [Oksendal, 2013], the self-consistency property then allows BNSP models to define partitions in the *infinite* multi-dimensional space.

We now explore the various partition strategies, where the order is approximately in terms of the proposed times and model complexities of these strategies.

2.1 Regular-Grid Partitions

Grid-style BNSP models use one-dimensional partitions on each univariate dimension to construct a regular-grid on the full dimensional space. Regular-grid partitions extend trivially from one-dimensional partitions (e.g. the stick-breaking process), and are a historically earlier and simpler instance of BNSP models. For a D -dimensional space, the model is constructed using D independent stick-breaking processes, each designated as a partition of one dimension. The orthogonal crossover of the partition on these dimensions (Figure 2(a)) produces regular-grid blocks in the full-dimensional space. Due to the self-consistency of the stick-breaking process in each dimension separately, the resulting regular-grid partition is also self-consistent.

The regular-grid partition is an over-simplified partition strategy, as its construction ignores any dependency between dimensions and generates partitions over each dimension independently. Accordingly, this approach is very likely to generate trivial and unneeded blocks in data-sparse regions. Cur-

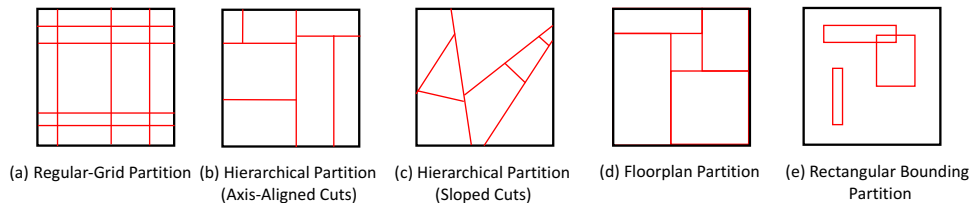


Figure 2: Visualisation of five typical BNSP partition strategies in a 2-dimensional space.

rently, this approach is mainly applied in relational modelling in 2-dimensional space. A specific implementation is the Infinite Relational Model (IRM) [Kemp *et al.*, 2006], which is an infinite-state variant of the Stochastic Block Model (SBM) [Nowicki and Snijders, 2001]. Other models in this category include dynamic IRMs [Ishiguro *et al.*, 2010] for modelling dynamic relational data, and multi-membership relational modelling [Schmidt and Mørup, 2013].

2.2 Hierarchical Partitions (Axis-Aligned Cuts)

Hierarchical partitions follow a top-down strategy to recursively cut an existing block into two new blocks. In this way, the blocks are organized in a binary tree. The Mondrian process (MP) [Roy and Teh, 2009] is the representative construction of hierarchical partitions. In general, the MP recursively generates axis-aligned cuts on a unit hypercube $[0, 1]^D$, and divides the space in a hierarchical fashion known as kd -tree (Figure 2(b)). The kd -tree construction process is regulated by attaching an exponentially distributed cost to each axis-aligned cut. The process is then terminated when the accumulated sum of the costs exceeds a provided budget value. Because the MP can partially consider inter-dimensional dependency, it can produce fewer trivial blocks than regular-grid partitions. The MP retains the self-consistency property by carefully modelling the relationship between the cut cost and the generation of axis-aligned cuts.

2.3 Hierarchical Partitions (Sloped Cuts)

In comparison to the axis-aligned cuts, a related group of tree-structured hierarchical partition models consider sloped (non-axis-aligned) cuts for cutting a space (Figure 2(c)). BNSP processes adopting this strategy include the Ostomachion process [Fan *et al.*, 2016], the Binary Space Partition-Tree (BSP-Tree) process [Fan *et al.*, 2018a] and Random Tessellation Forests (RTF) [Ge *et al.*, 2019]. In contrast to the axis-aligned partitions of the MP, the BSP-Tree process accounts for inter-dimensional dependency by generating sloped cuts, and thereby forming convex polygon-shaped blocks in a 2-dimensional space. Binary Space Partition Forests [Fan *et al.*, 2019] extend the mechanism of producing sloped cuts from 2-dimensional spaces to D -dimensional spaces ($D \geq 2$). This results in the production of convex-polyhedron blocks, with the restriction that the cutting hyperplane is parallel to $D - 2$ dimensions.

Similarly, RTF are constructed by generating arbitrary sloped cutting hyperplanes in a D -dimensional space. These sloped cuts permit a greater focus on describing the multi-dimensional dependence; as a result, the model has the potential to produce partitions more efficiently in the space. Recent

work is able to efficiently generalise cut directions using the iteration stable (STIT) tessellations technique [O’Reilly and Tran, 2020].

2.4 Floorplan Partitions

It is noted that the regular-grid partitions and hierarchical partitions (axis-aligned cuts) are limited to support a subset of floorplan partitions (e.g., they cannot form Figure 2(d)). Baxter Permutation Partitions (BPPs) [Nakano *et al.*, 2020] were recently proposed to produce arbitrary floorplan partitions on a continuous space (Figure 2(d)). The BPP first uses the one-to-one correspondence between Baxter permutations [Dulucq and Guibert, 1998] and floorplan partitions to generate floorplan partitions on 2-dimensional arrays. This is then combined with the block-breaking process – a multi-dimensional extension of the stick-breaking process [Sethuraman, 1994] – to produce a discrete-time Markov process over the set of floorplan partitions on a continuous space. By getting rid of the restriction of a regular-grid or hierarchical structure, the BPPs are able to produce a variety of partition structures. The BPP is self-consistent and has been successfully applied in relational modelling tasks.

A discrete space alternative is the Rectangular Tiling Process (RTP) [Nakano *et al.*, 2014a], which produces a floorplan partition structure on a 2-dimensional array. It typically uses the geometric distribution to generate the length of each block, with the constraint that the length does not violate the rectangular restriction of existing blocks. It is also self-consistent due to the memoryless property of the geometric distribution. However, its generative process is quite complicated for practical usage and the domain of discrete array in RTP restricts it being used in e.g. relational modelling applications only. In contrast, the partition strategies discussed above can be applied to both continuous space and multi-dimensional arrays (with trivial modifications).

2.5 Rectangular Bounding Partitions

In direct contrast to the cutting-based strategies (including regular-grid partitions, hierarchical partitions and floorplan partitions), the Rectangular Bounding Process (RBP) [Fan *et al.*, 2018b] uses a bounding strategy to partition the space. By independently constructing rectangular bounding boxes in the space, the RBP can cover and concentrate on significant regions and avoid data sparse regions (Figure 2(e)). The RBP requires a budget parameter τ to control the number of bounding boxes, and a length parameter λ to control the size and location of the bounding boxes. The probability that any data point occupies a box is the same. The RBP is self-consistent,

and can generate more bounding boxes given a larger budget or a larger space.

The biggest advantage of the RBP is the parsimonious partition of a space. Real data points are typically not evenly distributed over the entire space, but clusters in local regions. The cutting-based strategies, such as regular-grid and hierarchical partitions, inevitably produce too many cuts for sparse regions with few data points while trying to fit data in the dense regions. Recall the credit risk modelling problem, where two dimensions of the feature space are ‘age’ and ‘salary’. Traditional cutting-based models may inevitably cut the regions of young age and very high salary even if there are very few people in those regions, simply by placing cuts in other areas of the space. In contrast, the RBP can place more bounding boxes in the most important regions, and fewer in data-sparse and noisy regions. As a result, the RBP is able to balance the *fitness* and *parsimony* of the partition.

2.6 Non-Self-Consistent Partitions

There are a number of space partition models that do not have the self-consistency property, which implies that their applicability is limited. Bayesian plaid models [Caldas and Kaski, 2008] generate ‘plaid’-like (discrete) partitions on a 2-dimensional array. Usually, the plaids are generated through the Beta-Bernoulli process, or the Indian Buffet Process (IBP) for an infinite number of plaids. The plaid models can produce bounding boxes similar to the continuous-space RBP, although the plaid models are restricted to discrete spaces. Each block is formed through individual row and column permutations. Plaid models are not self-consistent.

Similarly, Matrix Tile Analysis (MTA) models [Givoni *et al.*, 2006] are constructed on a 2-dimensional array, although it is a non-Bayesian method. MTA generates rectangular boxes on discrete arrays with the constraint that the boxes cannot be overlapped. Due to its non-Bayesian nature, the self-consistency property is not applicable to it.

The Bayesian Additive Regression Trees (BART) [Chipman *et al.*, 2010] is a space-partition model using the hierarchical partition strategy, but without the self-consistency property. In general, BARTs assign uniform distributions to the cutting positions and use the Bernoulli distribution to regulate the tree depth. As the parameter in the Bernoulli distribution is inversely related to the depth of the node in the tree, deeper nodes have a lower probability of being split.

2.7 Comparison

Table 1 summarises the similarities and differences of the reviewed space-partition models, in terms of: whether it possesses self-consistency, whether it is applicable to continuous or discrete spaces, the number of dimensions D in which it can be applied, the available inference methods, whether it can account for inter-dimensional dependency, and suitable applications. Apparently, there is not one single model that can dominate others in all modelling aspects, so in practice it is important to select the most appropriate model.

Most of the inference algorithms for BNSP models are based on Markov chain Monte Carlo (MCMC) methods. In particular, Gibbs samplers are used for the regular-grid partition models, Bayesian plaid models and BARTs, as conju-

gacy between the prior and posterior distributions is satisfied by each latent variable. The Reversible-Jump MCMC (RJ-MCMC) and Particle Gibbs (PG) algorithms are required for the hierarchical partition-based models, due to the tree structures of their latent variables. The Metropolis-Hastings (M-H) algorithm is used for the RTP, BPP and RBP models, due to the specially designed distributions of their latent variables. The Iterative Condition Modes (ICM) method [Givoni *et al.*, 2006] is used for the non-Bayesian MTA model.

3 Applications

In addition to regression trees (with static data), BNSP constructions can be applied to other modelling scenarios. In the following we review three of them: online learning, random feature construction, and relational modelling.

3.1 Online Learning

Some BNSP processes can be extended beyond regression-tree modelling for static datasets, to the online learning setting [Lakshminarayanan *et al.*, 2014; Lakshminarayanan *et al.*, 2016; Mourtada *et al.*, 2017; Fan *et al.*, 2020]. Suppose we observe a set of N labelled data points $\{(\mathbf{x}_n, y_n)\}_{n=1}^N \in \mathbb{R}^D \times \mathbb{R}$ which arrive over time, with y_n as the corresponding label of \mathbf{x}_n . When a new data point arrives, it is incorporated into the analysis, and the BNSP structure is updated accordingly without refitting the model using the full dataset.

The online learning application of BNSP processes follows the same spirit as online random forest-type algorithms [Breiman, 2000], which assume that the tree-structured model is generated independently of the data labels. The Mondrian Forest (MF) [Lakshminarayanan *et al.*, 2014; Lakshminarayanan *et al.*, 2016] is the first model to apply BNSP processes to the online learning setting. It uses the Mondrian process [Roy and Teh, 2009] to place a probability distribution over all kd -tree partitions of the space. To regularize the MF to be universally consistent (which ensures the prediction error converges to the Bayes error), the budget parameter is increased with the amount of data [Mourtada *et al.*, 2017], and so the model can achieve the minimax rate in a multi-dimensional space for single decision trees. [Mourtada *et al.*, 2018] shows the advantage of forest-based settings through improved convergence results. The Online Binary Space Partition Forest [Fan *et al.*, 2020] similarly extends the BSP-Tree to the online learning setting, randomly generating sloped hyperplanes to cut the feature space.

3.2 Random Feature Construction

The generative process of BNSP models can support the construction of random features to approximate kernels [Rahimi and Recht, 2008]. Given N data points and K generated blocks, a binary random feature matrix $\Phi \in \{0, 1\}^{N \times K}$ can be used to record the box coverage status for the data points, where the (n, k) -th entry indicates whether the n -th data point is covered by the k -th block. The random features Φ_n for each data point can replace the observed feature vector \mathbf{x}_n , and be used to approximate certain kinds of kernel.

The Mondrian Kernel (MK) [Balog *et al.*, 2016] is the first model to implement the random feature construction based

Models	Self-consistency	Continuous	D	Inference	IDD	Applications
Regular-grid	✓	✓	2	Gibbs	×	RM
MP	✓	✓	≥ 2	RJ-MCMC & PG	✓	RM & RT & RFC
BSP	✓	✓	≥ 2	RJ-MCMC & PG	✓	RM & RT
RTF	✓	✓	≥ 2	SMC	✓	RT
BPP	✓	✓	2	M-H	✓	RM
RBP	✓	✓	≥ 2	M-H	×	RM & RT
Plaid	×	×	2	Gibbs	×	RM
MTA	×	×	2	ICM	×	RM
BART	×	✓	≥ 2	Gibbs	×	RT

Table 1: Comparison of various space-partition strategies (IDD: Inter-Dimensional Dependency; RM: Relational Modelling; RT: Regression Trees; RFC: Random Feature Construction; PG: Particle Gibbs; ICM: Iterative Conditional Modes).

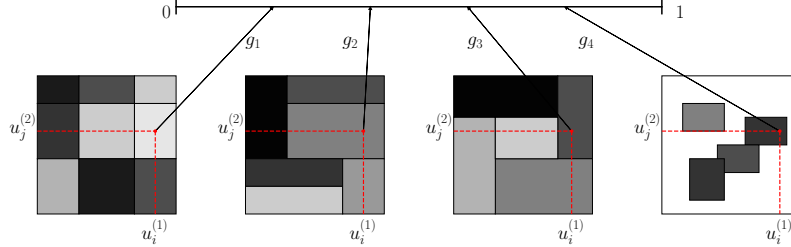


Figure 3: Relational modelling using BNSP models. The relation intensity of any pair of coordinates can be mapped to a unit interval using a graphon function (denoted by g_1, \dots, g_4). From left to right: regular-grid partitions; hierarchical partition (axis-aligned cuts); floorplan partitions, and rectangular bounding partitions. The darker color of the point $(u_i^{(1)}, u_j^{(2)})$ corresponds to a higher intensity in the interval, which means that there is a higher probability to generate a link between $u_i^{(1)}$ and $u_j^{(2)}$.

on BNSP models. In particular, the MK generates a set of M Mondrian process partitions on the feature space and incorporates all M partitions into one random feature matrix $\Phi \in \{0, 1\}^{N \times (MK)}$. As a result, the product $\Phi_n \Phi_{n'}^\top$ represents the count of blocks covering both the n -th and n' -th data points. [Balog *et al.*, 2016] shows that both the expectation of $\Phi \Phi^\top$, and the case of $\Phi \Phi^\top$ as $M \rightarrow \infty$ results in the Laplace kernel. [O'Reilly and Tran, 2020] extends the axis-aligned cuts in the Mondrian process to arbitrary directions, and characterise all possible kernels (including the radial basis function kernel and the Laplace kernel) that the hierarchical partitions can approximate.

3.3 Relational Modelling for Link Prediction

Each data point in relational modelling represents the linkage information of entities, with the value of the link (e.g. R_{ij} between individuals i and j) as the label. Each entity is associated with one latent covariate (e.g. an *a priori* uniformly distributed variable u_i for individual i). The feature information for R_{ij} is then the pair of random variables (u_i, u_j) . The blocks then define community-by-community interactions, and the links located in the same block follow the same Bernoulli distribution.

For relational modelling, the observed linkage data $\mathbf{R} = [R_{ij}]$ is regarded as the label data, which is usually presented as a symmetric (undirected) or asymmetric (directed) matrix $\mathbf{R} \in \{0, 1\}^{N \times N}$, with $R_{ij} = 1$ indicating that entity i interacts with entity j , otherwise $R_{ij} = 0$. Since \mathbf{R} represents pairwise linkage relations, the space of this application is spanned by two community distributions, and en-

coded as a unit square $[0, 1]^2$. The model generates *pseudo* attributes $u_i^{(1)}, u_i^{(2)}$, and for each link R_{ij} , concatenates them as $[u_i^{(1)}, u_j^{(1)}]^\top$ for entities i and j as *pseudo* features. The corresponding generative process is expressed as:

- (1) $\{\square_k\}_k \sim \text{BNSP}([0, 1]^2, -);$
- (2) $\{\omega_k\}_k \sim \text{Beta}(-);$
- (3) $\{u_i^{(1)}, u_i^{(2)}\}_i \sim \text{Uniform}[0, 1];$
- (4) $R_{ij} \sim \text{Bernoulli}(\omega_{k:(u_i^{(1)}, u_j^{(2)}) \in \square_k}).$

where $\{\square_k\}_k$ represents meaningful community-by-community interaction groups, and the intensity ω_k denotes the influence contributing to the links belonging to \square_k .

There are various extensions to the above basic generative process. In the case of overlapping blocks [Fan *et al.*, 2018b], the intensity can be assumed to follow a normal distribution. Positive (negative) intensity would then promote (suppress) links located in the corresponding block. In the case of categorical or real-valued links, different types of likelihood functions can be adopted. E.g., a Gamma distribution likelihood can be used to model non-negative real-valued links.

Connection to graphons: Relational modelling applications are closely related to the graphon (graph function) literature [Orbanz and Roy, 2015]. Given the exchangeable relational data for relational modelling, the Aldous–Hoover theorem [Aldous, 1981] provides the theoretical foundation to model exchangeable multi-dimensional arrays conditioned on a stochastic partition model. A random 2-dimensional array is separately exchangeable if its distribution is invariant

under separate permutations of rows and columns. Specifically, the theorem states that:

Theorem 1. [Orbanz and Roy, 2015] *A random array (R_{ij}) is separately exchangeable if and only if it can be represented as follows: there exists a random measurable function $F : [0, 1]^3 \mapsto \mathcal{X}$ such that $(R_{ij}) \stackrel{d}{=} (F(u_i^{(1)}, u_j^{(2)}, \nu_{ij}))$, where $\{u_i^{(1)}\}_i, \{u_j^{(2)}\}_j$ and $\{\nu_{ij}\}_{i,j}$ are two sequences and an array of i.i.d. uniform random variables in $[0, 1]$, respectively.*

Many of the BNSP models comply with this theorem, with specific forms of mapping function F . For instance, as illustrated in Figure 3, given the uniformly distributed node coordinates $(u_i^{(1)}, u_j^{(2)})$, the regular-grid partition is related to a regular-grid graphon; the hierarchical partitions (axis-aligned cuts) is related to a kd -tree structured graphons; and the RBP is related to a box-wise constant graphon. All these graphons are piece-wise constant in $[0, 1]^2$.

4 Challenges

While BNSP models have developed into highly flexible models that have demonstrated outstanding successes in a number of useful applications, there are still a number of open research questions and remaining challenges.

4.1 Scalable Inference Methods

To the best of our knowledge, most inference methods for BNSP models rely on MCMC simulation (see Table 1). Since MCMC methods often result in long computational times and require convergence assessment, alternative scalable inference methods are necessary to deal with larger-scale data problems. Variational methods, in particular the popular variational auto-encoder methods [Kingma and Welling, 2013], are promising solutions as they produce optimization-based (rather than simulation-based) posterior approximations. However, there has been little progress in developing variational inference methods for BNSP models, with the exception of the regular-grid partition. One possible reason for this might be the complexity of the partition structure, such as the tree structure in a hierarchical partition.

Among hierarchical partition models, the BNSP strategy of using sloped cuts for partitions seems to be more effective and flexible than the axis-aligned cutting strategy. However, the improvement of modelling capability comes at the price of increased computational cost. The cost of current sloped-cut models scales at least quadratically with the number of dimensions of the space, whereas the cost of axis-aligned models often scales linearly with the number of dimensions. Efficient ways of circumventing computational complexity, while retaining self-consistency, is an interesting challenge.

4.2 Flexibility of Partitions

Dependent BNSP partitions: Currently, BNSP models generate space partitions independently of other partitions. One potential for improvement could be to allow for dependence between different partitions, and accordingly extend the application of BNSP models to a greater range of data formats, such as dynamic data, in which data have dependence

across different time points, cross-domain data, in which data may contain dependencies across different domains, and multi-view data, in which data have dependencies through different descriptions. However, such extensions would be nontrivial, as it would be easy to violate self-consistency. Innovative approaches may be needed to incorporate partition dependence while simultaneously retaining self-consistency.

Convex-polygon bounding blocks: In the bounding box-based strategy, it is assumed that the partition is constructed independently in all dimensions, such that rectangular bounding boxes are produced. In practice, the most efficient shape of a block might be a convex or even an irregular polygon. For example, in credit risk modelling (with expense and salary), one high risk block might be formed by a sloped cutting line of expense \geq salary. In this problem, the challenge is to define the convex polygon while retaining self-consistency.

4.3 Posterior Concentration Analysis

The importance of posterior concentration behavior in the Bayesian generative process has been repeatedly emphasised in the literature. Currently, [Rocková and van der Pas, 2017] has developed a series of posterior concentration results for BARTs, based on the work of [Ghosal *et al.*, 2007]. No such results exist for other BNSP models. It would be interesting to see how posterior concentration analysis could be integrated into the hierarchical partition or even rectangular bounding partition BNSP models.

4.4 Analysis of Deep Neural Networks

Recent work [Balestriero and Baraniuk, 2018; Balestriero *et al.*, 2019] explores the possibility of using a Power-Diagram partition – a generalisation of a Voronoi tiling partition – to analyse the piecewise-affine activation functions between the layers of a Deep Neural Network (DNN). However, this work is restricted to analysing piecewise-affine activation functions and is unable to be applied to general DNN architecture with non-linear activation functions. Exploring these ideas with more flexible and advanced partitioning structures could bring great insights and new ideas to DNN research.

5 Conclusions

In this brief survey, we provide a first review on Bayesian Nonparametric Space Partition (BNSP) models, through studying five typical BNSP strategies, introducing current applications of BNSP models, and discussing current challenges of BNSP models.

Acknowledgements

Xuhui Fan and Scott A. Sisson are supported by the Australian Research Council through the Australian Centre of Excellence in Mathematical and Statistical Frontiers (ACEMS, CE140100049), and Scott A. Sisson through the Future Fellow Scheme (FT170100079). Bin Li is supported in part by STCSM Project (20511100400), Shanghai Municipal Science and Technology Major Projects (2018SHZDZX01, 2021SHZDZX0103), and the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

References

- [Aldous, 1981] David J. Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.
- [Balestrierio and Baraniuk, 2018] Randall Balestrierio and Richard Baraniuk. A spline theory of deep learning. In *ICML*, pages 374–383, 2018.
- [Balestrierio *et al.*, 2019] Randall Balestrierio, Romain Cosentino, Behnaam Aazhang, and Richard Baraniuk. The geometry of deep networks: Power diagram subdivision. In *NeurIPS*, pages 15832–15841, 2019.
- [Balog *et al.*, 2016] Matej Balog, Balaji Lakshminarayanan, Zoubin Ghahramani, Daniel M Roy, and Yee Whye Teh. The mondrian kernel. *arXiv preprint arXiv:1606.05241*, 2016.
- [Breiman, 2000] Leo Breiman. Some infinity theory for predictor ensembles. Technical report, Technical Report 579, Statistics Dept. UCB, 2000.
- [Caldas and Kaski, 2008] José Caldas and Samuel Kaski. Bayesian biclustering with the plaid model. In *MLSP 2008. IEEE Workshop on*, pages 291–296. IEEE, 2008.
- [Chipman *et al.*, 2010] Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.
- [Dulucq and Guibert, 1998] S. Dulucq and O. Guibert. Baxter permutations. *Discrete Mathematics*, 180(1):143–156, 1998.
- [Fan *et al.*, 2016] Xuhui Fan, Bin Li, Yi Wang, Yang Wang, and Fang Chen. The Ostomachion Process. In *AAAI*, pages 1547–1553, 2016.
- [Fan *et al.*, 2018a] Xuhui Fan, Bin Li, and Scott A. Sisson. The binary space partitioning-tree process. In *AISTATS*, volume 84, pages 1859–1867, 2018.
- [Fan *et al.*, 2018b] Xuhui Fan, Bin Li, and Scott A. Sisson. Rectangular bounding process. In *NeurIPS*, pages 7631–7641, 2018.
- [Fan *et al.*, 2019] Xuhui Fan, Bin Li, and Scott A. Sisson. Binary space partitioning forests. In *AISTATS*, volume 89, pages 3022–3031, 2019.
- [Fan *et al.*, 2020] Xuhui Fan, Bin Li, and Scott A. Sisson. Online binary space partitioning forests. In *AISTATS*, 2020.
- [Ferguson, 1973] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, pages 209–230, 1973.
- [Ge *et al.*, 2019] Shufei Ge, Shijia Wang, Yee Whye Teh, Liangliang Wang, and Lloyd Elliott. Random tessellation forests. In *NeurIPS*, pages 9571–9581, 2019.
- [Ghosal *et al.*, 2007] Subhashis Ghosal, Aad Van Der Vaart, et al. Convergence rates of posterior distributions for non-iid observations. *The Annals of Statistics*, 35(1):192–223, 2007.
- [Givoni *et al.*, 2006] Inmar Givoni, Vincent Cheung, and Brendan Frey. Matrix tile analysis. In *UAI*, pages 200–207, 2006.
- [Ishiguro *et al.*, 2010] Katsuhiko Ishiguro, Tomoharu Iwata, Naonori Ueda, and Joshua B. Tenenbaum. Dynamic infinite relational model for time-varying relational data analysis. In *NIPS*, pages 919–927, 2010.
- [Kemp *et al.*, 2006] Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, pages 381–388, 2006.
- [Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [Kuck *et al.*, 2019] Jonathan Kuck, Tri Dao, Hamid Rezatofghi, Ashish Sabharwal, and Stefano Ermon. Approximating the permanent by sampling from adaptive partitions. In *NeurIPS*, pages 8860–8871. 2019.
- [Lakshminarayanan *et al.*, 2014] Balaji Lakshminarayanan, Daniel M. Roy, and Yee Whye Teh. Mondrian forests: Efficient online random forests. In *NIPS*, pages 3140–3148, 2014.
- [Lakshminarayanan *et al.*, 2016] Balaji Lakshminarayanan, Daniel M Roy, and Yee Whye Teh. Mondrian forests for large-scale regression when uncertainty matters. In *AISTATS*, pages 1478–1487, 2016.
- [Mourtada *et al.*, 2017] Jaouad Mourtada, Stéphane Gaïffas, and Erwan Scornet. Universal consistency and minimax rates for online mondrian forests. In *NIPS*, pages 3758–3767, 2017.
- [Mourtada *et al.*, 2018] Jaouad Mourtada, Stéphane Gaïffas, and Erwan Scornet. Minimax optimal rates for mondrian trees and forests. *arXiv preprint arXiv:1803.05784*, 2018.
- [Nakano *et al.*, 2014a] Masahiro Nakano, Katsuhiko Ishiguro, Akisato Kimura, Takeshi Yamada, and Naonori Ueda. Rectangular tiling process. In *ICML*, pages 361–369, 2014.
- [Nakano *et al.*, 2014b] Masahiro Nakano, Yasunori Ohishi, Hirokazu Kameoka, Ryo Mukai, and Kunio Kashino. Mondrian hidden Markov model for music signal processing. In *ICASSP*, pages 2405–2409, 2014.
- [Nakano *et al.*, 2020] Masahiro Nakano, Akisato Kimura, Takeshi Yamada, and Naonori Ueda. Baxter permutation process. In *NeurIPS*, pages 8648–8659, 2020.
- [Nowicki and Snijders, 2001] Krzysztof Nowicki and Tom A.B. Snijders. Estimation and prediction for stochastic block structures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.
- [Oksendal, 2013] Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.
- [Orbanz and Roy, 2015] Peter Orbanz and Daniel M. Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(02):437–461, 2015.

- [O'Reilly and Tran, 2020] Eliza O'Reilly and Ngoc Tran. Stochastic geometry to generalize the mondrian process. *arXiv preprint arXiv:2002.00797*, 2020.
- [Rahimi and Recht, 2008] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2008.
- [Rocková and van der Pas, 2017] Veronika Rocková and Stéphanie van der Pas. Posterior concentration for bayesian regression trees and forests. *arXiv preprint arXiv:1708.08734*, 2017.
- [Roy and Teh, 2009] Daniel M. Roy and Yee Whye Teh. The Mondrian process. In *NIPS*, pages 1377–1384, 2009.
- [Schmidt and Mørup, 2013] Mikkel N. Schmidt and Morten Mørup. Nonparametric Bayesian modeling of complex networks: An introduction. *IEEE Signal Processing Magazine*, 30(3):110–128, 2013.
- [Sethuraman, 1994] Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650, 1994.
- [Wang *et al.*, 2011] Pu Wang, Kathryn B. Laskey, Carlotta Domeniconi, and Michael I. Jordan. Nonparametric Bayesian co-clustering ensembles. In *SDM*, pages 331–342, 2011.