Rectangular Tiling Process

Masahiro Nakano NAKANO.MASAHIRO@LAB.NTT.CO.JP
Katsuhiko Ishiguro ISHIGURO.KATSUHIKO@LAB.NTT.CO.JP
Akisato Kimura KIMURA.AKISATO@LAB.NTT.CO.JP
Takeshi Yamada YAMADA.TAK@LAB.NTT.CO.JP
Naonori Ueda UEDA.NAONORI@LAB.NTT.CO.JP
NTT communication science laboratories, Morinosato Wakamiya 3-1, Atsugi-shi, Kanagawa

Abstract

This paper proposes a novel stochastic process that represents the arbitrary rectangular partitioning of an infinite-dimensional matrix as the conditional projective limit. Rectangular partitioning is used in relational data analysis, and is classified into three types: *regular grid*, *hierarchical*, and *arbitrary*. Conventionally, a variety of probabilistic models have been advanced for the first two, including the product of Chinese restaurant processes and the Mondrian process. However, existing models for arbitrary partitioning are too complicated to permit the analysis of the statistical behaviors of models, which places very severe capability limits on relational data analysis. In this paper, we propose a new probabilistic model of arbitrary partitioning called the rectangular tiling process (RTP). Our model has a sound mathematical base in projective systems and infinite extension of conditional probabilities, and is capable of representing partitions of infinite elements as found in ordinary Bayesian nonparametric models.

1. Introduction

Relational data is now being used in various applications in order to represent richly structured, real-world data. In particular, pairwise relations represented by matrices, i.e., $(Y_{i,j})_{m\times n}$, are subjects of intense study. For example, in the context of binary relational data, entry $Y_{i,j}$ represents an on/off connection between the i-th and j-th elements. One of the most important problems associated with relational data analysis is to discover clusters that are hiding in the relational data. In the context of Bayesian relational data analysis, a telling example is the stochastic block model (SBM) (Nowicki & Snijders, 2001), which extracts hidden clusters through the rectangular partitioning of the matrix (Wasserman & Anderson, 1987). The goal of SBM is to find permutations of rows and columns, and their rectangular partitioning. The central problem is how to construct generative models to yield rectangular partitioning.

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

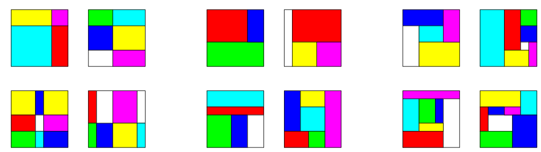

Figure 1. **Left:** Regular grid partitioning. **Middle:** Hierarchical partitioning. **Right:** Arbitrary partitioning.

Kemp et al. (2006) presented a Bayesian nonparametric model, called the infinite relational model (IRM), based on the product of Chinese restaurant processes (CRP) on both rows and columns. The partitions obtained by IRM are restricted to regular grids. Splitting one area of the matrix requires the other parts to be divided, even if the data do not imply such structure. Motivated by this, Roy & Teh (2009) proposed the Mondrian process (MP), which can be regarded as a multi-dimensional generalization of Poisson processes (Roy, 2011). MP allows more flexible partitioning of the matrix by creating inner rectangles, not splitting entire rows or columns. However, even the MP can generate only a limited class of rectangular partitionings.

In general, there are three types of rectangular partitionings (Muthukrishnan et al., 1999) (Fig. 1):

- *Regular grid*: The rows and columns are partitioned into clusters. Each block is characterized by the product of the row and column clusters,
- *Hierarchical*: Partitionings are expressed as binary trees where nodes represent a vertical or horizontal separation of a rectangle into two disjoint rectangles,
- *Arbitrary*: No restrictions are required.

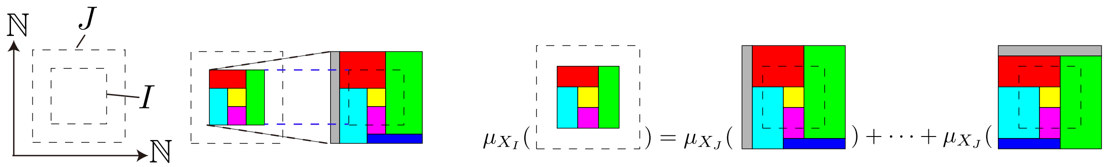

Figure 2. **Left:** Projector of rectangular partitioning. For any sub-arrays I, J ($I \prec J \in \mathcal{F}(E)$), the projection $P_{J,I}$ restricts the partition of J by keeping the I entries unchanged, and removing the remaining entries. **Right:** Illustration of *projectivity*. Left side shows the probability of a sample partitioning of array I. Right side shows the probability of the partitioning of array J ($I \prec J$) whose projection (i.e., restriction) onto I is equivalent to the left side. For any I, J ($I \prec J$), this equation should hold.

IRM and MP correspond to *regular grid* and *hierarchical* types, respectively. We are interested in general *arbitrary* partitioning, and propose a new model accordingly.

To the best of our knowledge, arbitrary partitioning has been little discussed in machine learning literature. However, the field of probability theory knows of probabilistic models for arbitrary rectangular partitioning. One well-known model is the Gilbert tessellation model (1967) with axis-parallel cuts (i.e., cracks) described below (Mackisack & Miles, 1996). First, consider points, (called *seeds*), drawn from a stationary Poisson process on a plane. At a given time, each of these seeds initiates the growth of a line. The lines are confined to two orthogonal directions, i.e., “vertical” or “horizontal”. Each seed uniformly and independently chooses a direction from {“vertical”, “horizontal”}, and each line grows bidirectionally from its seed at the same rate. When the current line encounters and intersects another line that has grown from another seed, the growth of that line stops. As a result, Gilbert tessellations can generate arbitrary partitionings. However, it is notoriously difficult to analyze statistical behaviors of Gilbert tessellations (Burridge et al., 2013). For example, no analytic solution yielding the ray-length distribution (or the expected length, height/width of each block) has been found.

Our goal is to construct a new probabilistic model for arbitrary rectangular partitioning where the distribution of the height/width of each block can be easily analyzed. As a result, we can easily employ it for SBM-based relational data analysis. We call it *rectangular tiling process* (RTP), analogous to *rectangular tiling problem* in combinatorics.

2. Preliminaries

Notations: Random events are modeled by the abstract probability space $(\Omega, \mathcal{A}, \mathbb{P})$, where Ω is a point set, $\mathcal{A}$ is a σ-algebra on Ω, and $\mathbb{P}$ is a probability measure. A random variable (e.g., X) is a measurable mapping from Ω into some space of observations (e.g., $\mathcal{X}$), such as $X : \Omega \to \mathcal{X}$. Their distribution is denoted as $\mu_X := X(\mathbb{P}) = \mathbb{P} \circ X^{-1}$.

2.1. Construction of infinite models

In machine learning, there are two main approaches to obtaining infinite models: (1) the use of well-known stochastic processes, including Dirichlet processes, beta processes, and Bernoulli processes; (2) applying infinite extension theorems to a family of finite-dimensional models. As with most Bayesian nonparametric models in machine learning, the former strategy makes it much easier to construct infinite models if it is possible. However, the former strategy, sometimes, is not applicable to the problem. This makes it essential to consider the latter approach. That is exactly the case for the arbitrary rectangular partitionings.

2.1.1. Kolmogorov’s theorem (used in Sec. 3.2)

Sketch: A **projective** family of finite-dimensional models can be extended to an infinite-dimensional model.

It follows from Kolmogorov’s extension theorem that we can construct infinite-dimensional models via a family of finite-dimensional marginals, namely, the *projective system* (Crane, 2012). This paper deals with rectangular partitionings of infinite-dimensional matrix $E := \mathbb{N} \times \mathbb{N}$. A **projective system** is a family indexed by the elements of index set $\mathcal{F}(E)$, where we use $\mathcal{F}(E)$ as the set of all finite subarrays of E (i.e., for any $m \le m' \in \mathbb{N}$, and $n \le n' \in \mathbb{N}$, $\{m, m+1, \ldots, m'\} \times \{n, n+1, \ldots, n'\}$). We consider a family of measurable spaces $(\mathcal{X}_I, \mathcal{B}_I)$ with $I \in \mathcal{F}(E)$, where $\mathcal{X}_I$ means a set of rectangular partitionings of I. The index set $\mathcal{F}(E)$ has partial order relation $\preceq$, and, whenever $I \prec J \in \mathcal{F}(E)$, there exists $K \in \mathcal{F}(E)$ such that $I \preceq K$ and $J \preceq K$. As Fig. 2 shows, the component spaces $(\mathcal{X}_I)_{I \in \mathcal{F}(E)}$ are related via *projection*. The projection operator from $\mathcal{X}_J$ to $\mathcal{X}_I$ will be denoted $P_{J,I}$. Projection $P_{J,I}$ restricts the partition of J by keeping the I entries unchanged, and removing the remaining entries. For sets $B_I \subset \mathcal{X}_I$, the preimage under projection is denoted as $P^{-1}_{J,I} B_I = \{X_J \in \mathcal{X}_J | P_{J,I} X_J \in B_I\}$. The projection of a measure is defined, by means of a push-forward, as $(P_{J,I}\mu_{X_J})(B_I) := \mu_{X_J}(P^{-1}_{J,I} B_I)$. This family defines measurable space $(\mathcal{X}_E, \mathcal{B}_E)$, called the *projective limit*.

Theorem 2.1 (Bochner, 1955) *Let* $(\mathcal{X}_I, \mathcal{B}_I, \mu_{X_I})_{I \in \mathcal{F}(E)}$ *be a projective system of measurable spaces such that, for projection* $P_{J,I} : \mathcal{X}_J \to \mathcal{X}_I$, $\mu_{X_I}(B_I) = \mu_{X_J}(P^{-1}_{J,I} B_I)$ *holds for all* $B_I \in \mathcal{B}_I$*. Then* μ_{X_I} $(I \in \mathcal{F}(E))$ *can be uniquely extended to measure* μ_{X_E} *on* $(\mathcal{X}_E, \mathcal{B}_E)$ *as the projective limit measurable space.*

2.1.2. ORBANZ'S THEOREM (USED IN SEC. 3.3)

Sketch: A **conditionally projective** family of finite "Bayesian" models can be extended to an infinite model.

"Conditioning" is useful in the context of probabilistic data modeling since it can lead to the hierarchical structure of probabilistic models. We may wonder if we can extend a family of finite hierarchical models to an infinite model by analogy to the Kolmogorov extension. Recently, Orbanz gave a positive answer to this question (2008; 2009).

In the context of Bayesian modeling, hierarchical models typically correspond to parametric models. Consider parameter variables $\Theta_I : \Omega \to \mathcal{Y}_I$, and the parametric model of X_I with parameter Θ_I is the conditional distribution $\mu^I(X_I|\Theta_I)$. The family $\mu^I(X_I|\Theta_I)$ with $I \in \mathcal{F}(E)$ is called *conditionally projective* if $(P_{J,I}\mu_{X_J})(\ .\ |\Theta_J) := \mu_{X_J}(P_{J,I}^{-1}\ .\ |\Theta_J) = \mu_{X_I}(\ .\ |\Theta_I)$ for all $I \prec J \in \mathcal{F}(E)$.

Theorem 2.2 (Orbanz, 2009) *Let $\mu^I(X_I|\Theta_I)$ be a family of regular conditional probabilities. If the family is conditionally projective, and the parameter variables are also projective (i.e., $P_{J,I}\Theta_J = \Theta_I$), there exists a conditional probability $\mu^E(X_E|\Theta_E)$ with projective limit Θ_E.*

Briefly, unlike the standard Kolmogorov extension setting, we have to take a projective limit with respect to both the sample variable and the parameter variable. For more details refer to (Orbanz, 2008; 2009). Moreover, Orbanz (2011) (lemma 2 and 3) provides applicable criteria to construct an infinite model by means of the above theorem.

2.2. Discrete Mondrian process (used in Sec. 3.3)

We describe here a special case of the Mondrian process (MP) (Roy & Teh, 2009; Roy, 2011), a Markov process with values in the space of hierarchical partitioning. The original MP generates partitioning of planes ($\in \mathbb{R} \times \mathbb{R}$), while the discrete MP deals with arrays ($\in \mathbb{N} \times \mathbb{N}$).

Consider stochastic rectangular partitioning of input array $I \in \mathbb{N} \times \mathbb{N}$: $\mathcal{M}_I \sim \mathrm{dMP}(\lambda, I)$, where $\lambda > 0$ is a *budget* parameter. Let e_I be the sum of the number of boundaries of rows and columns of I. That is, $e_I = \#\mathrm{row}(I) + \#\mathrm{column}(I) - 2$, where $\#\mathrm{row}(I)$ and $\#\mathrm{column}(I)$ are the number of rows and columns of I, respectively. Rectangular partitioning is recursively constructed. Let $\lambda' = \lambda - e'$, where $e' \sim \mathrm{Exp}(e_I)$. If $\lambda' < 0$, the process halts, and returns the current I. Otherwise, an axis-aligned cut splits I into two sub-arrays I' and I''. The cut is chosen uniformly at random from all possible boundaries of the rows and columns of I. The partition $\mathcal{M}$ is recursively generated from independent dMPs with the diminished budget λ' on both sides of the cut: $\mathcal{M} = \mathcal{M}_< \cup \mathcal{M}_>$, where $\mathcal{M}_< \sim \mathrm{dMP}(\lambda', I')$ and $\mathcal{M}_> \sim \mathrm{dMP}(\lambda', I'')$.

2.3. Aldous-Hoover exchangeable array (used in Sec. 4)

As an attractive application, rectangular partitionings can be used in relational data analysis. In the sense of infinite data analysis (finite observations of potentially infinite data), rows and columns should be infinitely exchangeable:

Theorem 2.3 (Aldous, 1981; Hoover, 1979) *Random array $(Y_{i,j})$ is separately exchangeable if and only if it can be represented as follows: There is a random measurable function $G : [0,1]^3 \to \boldsymbol{Y}$ such that $(Y_{i,j}) = G(U_i^{\mathrm{row}}, U_j^{\mathrm{column}}, U_{i,j})$, where U_i^{row}, U_j^{column} and $U_{i,j}$ are $\mathrm{Uniform}[0,1]$ random variables.*

This provides a natural way to construct the rectangular-partitioning-based exchangeable array (Orbanz & Roy, 2013): We first generate a rectangular partitioning of $[0,1]^2$, then each (i,j)-entry is assigned to a rectangle based on a geometrical interpretation of U_i^{row} and U_j^{column} on $[0,1]^2$, and finally, for example, categorical data $Y_{i,j}$ is generated from each Dirichlet-categorical model on the assigned rectangle. Since our RTP itself does not have exchangeability of rows and columns, we use this idea.

3. Rectangular tiling process

Our goal is to obtain probabilistic models for arbitrary rectangular partitioning. First, Sec. 3.1 explains a key strategy for projective rectangular partitionings, and presents a *local growth algorithm* (LGA) for a matrix with 2 rows and 2 columns. Second, Sec. 3.2 shows a *repeated local growth algorithm* (RLGA) that extends LGA to matrices of any finite size, which can be naively extended to an infinite model in the sense of Kolmogorov's theorem. However, this naive approach has both positive and negative properties. A positive side is that it is easy to analyze ray-length distributions unlike Gilbert tessellation. A negative side is that a parameter used in LGA (*direction of growth* described later) undesirably biases the resulting arrangements of blocks. To reduce the negative influence, Sec 3.3 presents a more sophisticated construction based on Orbanz's theorem. Proofs of theorems are described in supplementary material.

In the following, we consider a stochastic rectangular partitioning of a matrix with m rows and n columns ($m, n \in \mathbb{N}$), denoted by "$(m \times n)$-*array*". We call each tile of the rectangular partitioning a "*block*".

3.1. Local growth algorithm for (2×2)-array

Strategy: As a prototype of the projective partitioning, we embed projectivity among the relationships between a (2×2)-array , and a (2×1)- or a (1×2)-array.

As discussed in Sec. 2.1, we deal with a **projective** family

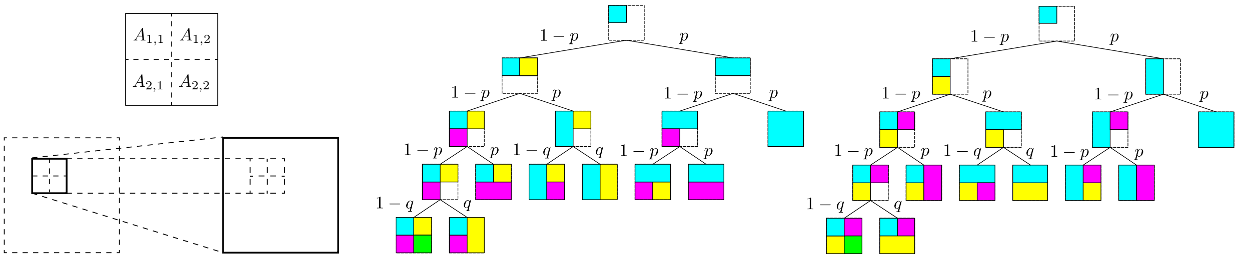

Figure 3. (Best viewed in color.) **Left:** (2×2)-array (top), and projectivity between (2×2)-array and array of any size (bottom). **Center:** Rectangular partitioning of (2×2)-array by LGA($\rightarrow\downarrow$). **Right:** LGA($\downarrow\rightarrow$). *Projectivity* implies that all probabilities that the adjacent two entries are assigned into the same block should be same. For example, if $A_{1,1}$ is assigned to the same block as $A_{1,2}$ with probability p, $A_{2,1}$ is assigned to the same block as $A_{2,2}$ with probability p, which is achieved by LGA with $q = p/(p^2 - p + 1)$.

of rectangular partitionings (Fig. 3 left). That is, we need to consider all pairs of finite sub-arrays I, J $(I \prec J \in \mathbb{N}^2)$. Thus, as a primitive example, we first consider the case with J as a $(2 \times 2)-$array and its stochastic rectangular partitioning. Naively, we assume that the top two entries ($A_{1,1}$ and $A_{1,2}$ in Fig. 3 left) are assigned to the same block with probability p. Projectivity implies that, if we remove the top entries and focus only on the bottom two entries ($A_{2,1}$ and $A_{2,2}$), they can be regarded as "top entries" of the current (1×2)-array, and their assignment must be done in a similar manner (that is, the probability that $A_{2,1}$ and $A_{2,2}$ are assigned to a same block is p). Similarly, this is just as valid for the left or right entries. In short, the probabilities that the top/left/right/bottom two entries are assigned to the same block are all p.

To obtain the projective partitionings, we present a *local growth algorithm* (LGA), that randomly generates the rectangular partitioning of a (2×2)-array. The input of this algorithm consists of real variables $p, q \in (0, 1)$ and a *direction of growth* chosen from eight patterns $\{\rightarrow\downarrow, \rightarrow\uparrow, \leftarrow\uparrow, \leftarrow\downarrow, \uparrow\rightarrow, \uparrow\leftarrow, \downarrow\rightarrow, \downarrow\leftarrow\}$. It consists of two stages (**prestage** and **main body**). Algorithm 1 shows LGA with $\rightarrow\downarrow$, and Fig. 3 illustrates LGA with $\rightarrow\downarrow$ and $\downarrow\rightarrow$. The direction of growth determines the order of merging/partitioning of the 4 possible pairs of adjacent entries (Fig. 3). All patterns are summarized as follows (supplementary material describes a formal description):

$\rightarrow\downarrow$	$(A_{1,1}, A_{1,2}) \Rightarrow (A_{1,1}, A_{2,1}) \Rightarrow (A_{2,1}, A_{2,2}) \Rightarrow (A_{1,2}, A_{2,2})$
$\downarrow\rightarrow$	$(A_{1,1}, A_{2,1}) \Rightarrow (A_{1,1}, A_{1,2}) \Rightarrow (A_{1,2}, A_{2,2}) \Rightarrow (A_{2,1}, A_{2,2})$
$\leftarrow\downarrow$	$(A_{1,2}, A_{1,1}) \Rightarrow (A_{1,2}, A_{2,2}) \Rightarrow (A_{2,2}, A_{2,1}) \Rightarrow (A_{1,1}, A_{2,1})$
$\downarrow\leftarrow$	$(A_{1,2}, A_{2,2}) \Rightarrow (A_{1,2}, A_{1,1}) \Rightarrow (A_{1,1}, A_{2,1}) \Rightarrow (A_{2,2}, A_{2,1})$
$\rightarrow\uparrow$	$(A_{2,1}, A_{2,2}) \Rightarrow (A_{2,1}, A_{1,1}) \Rightarrow (A_{1,1}, A_{1,2}) \Rightarrow (A_{2,2}, A_{1,2})$
$\uparrow\rightarrow$	$(A_{2,1}, A_{1,1}) \Rightarrow (A_{2,1}, A_{2,2}) \Rightarrow (A_{2,2}, A_{1,2}) \Rightarrow (A_{1,1}, A_{1,2})$
$\leftarrow\uparrow$	$(A_{2,2}, A_{2,1}) \Rightarrow (A_{2,2}, A_{1,2}) \Rightarrow (A_{1,2}, A_{1,1}) \Rightarrow (A_{2,1}, A_{1,1})$
$\uparrow\leftarrow$	$(A_{2,2}, A_{1,2}) \Rightarrow (A_{2,2}, A_{2,1}) \Rightarrow (A_{2,1}, A_{1,1}) \Rightarrow (A_{1,2}, A_{1,1})$

What is to be noted is that, if the procedure reaches the final (fourth) merging/partitioning of the above table, the probability of merger is q. To obtain the projective partitionings, we must appropriately choose the value of q as follows:

Algorithm 1 LOCAL GROWTH ALGORITHM (LGA)

Input: p, q $(0 < p < 1, 0 < q < 1)$ and *direction of growth* (as an example, assume $\rightarrow\downarrow$.)

Prestage

· We start from the top-left entry as a singleton.

· The top-right entry is assigned to the same block with probability p, otherwise, to a new block.

· The bottom-left entry is assigned to the same block with probability p, otherwise, to a new block.

Main body

· If the top-right and bottom-left entries are assigned to the same block, the bottom-right entry is also assigned to the same block. Otherwise, go to the next line.

· With probability p, the bottom-right entry is assigned to the block to which the bottom-left entry belongs. With probability $(1 - p)q$, it is assigned to the block to which the top-right entry belongs. With probability $(1-p)(1-q)$, it is assigned to a new block.

Output: Rectangular partitioning of a (2×2)-array.

Remark 3.1 *Consider stochastic rectangular partitionings of a* (2×2)*-array based on LGA with* $q = p/(p^2 - p + 1)$*. The probabilities that the top/left/right/bottom two entries are assigned to the same block are all* p *regardless of the choice of the direction of growth. For example, in Fig. 3 center, the right two entries are assigned to the same block with the sum of the probabilities of the (from left) 2nd* ($(1-p)^3q$)*, 5th* ($(1-p)pq$) *and 8th* (p^2) *leaves.*

3.2. Repeated LGA for array with any finite size

Strategy: We construct a **projective** (Sec. 2.1.1.) family of rectangular partitionings by applying LGA repeatedly. Theorem 3.3 gives us an infinite model based on Kolmogorov's theorem.

In the previous subsection, we construct a projective rectangular partitioning of a (2×2)-array. Recall that projectivity means that all two adjacent entries of a (2×2)-array are assigned to the same block with probability p.

Algorithm 2 Repeated LGA (RLGA)

Input: $(m \times n)-$array A, p, q $(0 < p < 1, 0 < q < 1)$ and *direction of growth* (as an example, $\rightarrow\downarrow$)
Prestage
· We start from the top-left entry $A_{1,1}$ as a singleton.
for $n' = 2$ to n **do**
  · With prob. p, $A_{1,n'}$ is assigned to the same block as $A_{1,n'-1}$; with prob. $(1-p)$, to a new block.
end for
for $m' = 2$ to m **do**
  · With prob. p, $A_{m',1}$ is assigned to the same block as $A_{m'-1,1}$; with prob. $(1-p)$, to a new block.
end for
Main body
for $m' = 2$ to m **do**
  for $n' = 2$ to n **do**
    · Given the block of $A_{m'-1,n'-1}$, $A_{m',n'-1}$, $A_{m'-1,n'}$, the assignment of $A_{m',n'}$ is determined by the main body of the LGA.
  end for
end for
Output: Rectangular partitioning of A.

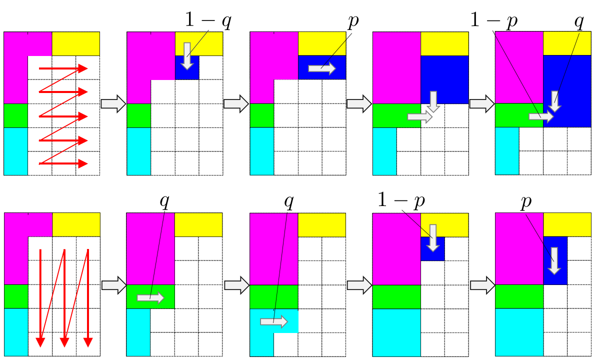

Figure 4. (Best viewed in color.) **Top:** Illustration of RLGA($\rightarrow\downarrow$) for (7×4)-array. (From left to right.) (1) The prestage of RLGA is completed. According to the direction of growth, the remaining entries are assigned to blocks. (2) The current block failed to grow vertically with probability $(1-p)$. (3) The block succeeded in growing horizontally with probability p. (4)-(5) The block first failed to grow horizontally with probability $(1-p)$, then succeeded in growing vertically with probability q. **Bottom:** Illustration of RLGA($\downarrow\rightarrow$).

Now, we extend this projectivity to matrices of any finite size. That is, the probability that adjacent two entries of a $(m \times n)$-array $(m, n \in \mathbb{N})$ are assigned into a same block are all p. To obtain such a projective rectangular partitioning, we can apply LGA repeatedly. Consider the rectangular partitioning of any finite-dimensional array, $(m \times n)$-array $(m \in \mathbb{N}, n \in \mathbb{N})$. As Fig. 4 shows, we can generate the rectangular partitioning by applying the *repeated local growth algorithm* (RLGA). Algorithm 2 and Fig. 4 shows the $\mathrm{RLGA}(\rightarrow\downarrow)$. The uppermost and leftmost entries are first assigned to blocks (**prestage**). Then we apply LGA to the remaining entries (**main body**). The key property of RLGA is described as follows:

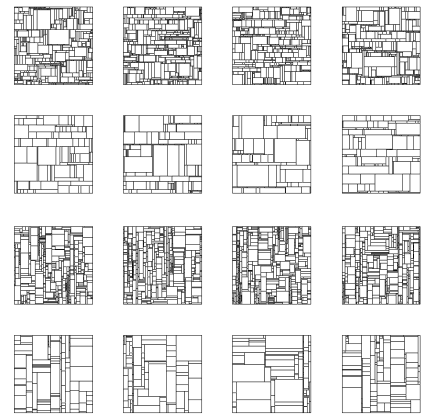

Figure 5. Undesirable property of naive RLGA. **From top.** Four samples drawn from $\mathrm{RLGA}(0.8, \rightarrow\downarrow)$, $\mathrm{RLGA}(0.9, \rightarrow\downarrow)$, $\mathrm{RLGA}(0.8, \downarrow\rightarrow)$, and $\mathrm{RLGA}(0.9, \downarrow\rightarrow)$.

Proposition 3.2 *For any sub-array $I \in \mathcal{F}(E)$, consider a space of rectangular partitionings of I (denoted by $\mathcal{T}_I$). Let $\mu_{\mathcal{T}_I}$ be a measure for $T_I \in \mathcal{T}_I$ provided by RLGA with $q = p/(p^2 - p + 1)$. Consider a distribution of the height (width) of each block of T_I drawn from $\mu_{\mathcal{T}_I}$, none of whose growing edges reach the edges of I. It is equivalent to a distribution of the number of successful trials with success probability p until failure.*

Kolmogorov's theorem gives us an infinite model, since the above proposition directly leads to a projective family of rectangular partitionings. Again, as the index set $\mathcal{F}(E)$, we employ the set of all sub-arrays of $\mathbb{N} \times \mathbb{N}$, partially ordered by inclusion. The projection operator $P_{J,I}$ from $\mathcal{T}_J$ to $\mathcal{T}_I$ (with $I \prec J$) restricts $T_J \in \mathcal{T}_J$ to the rectangular partitioning by keeping the I entries unchanged, and removing the remaining entries. It follows from Kolmogorov's theorem that RLGA provides an infinite model:

Theorem 3.3 *The family $\mu_{\mathcal{T}_I}$ $(I \in \mathcal{F}(E))$ can be uniquely extended to a measure on the projective limit measurable space $\mathcal{T}_E$. Moreover, the height (width) of each block is k with probability $p^{k-1}(1-p)$.*

Unlike Gilbert tessellations, it is a new infinite model for arbitrary rectangular partitioning, where the distribution of the height/width of each block can be easily analyzed. This makes us interested in the properties of the resulting rectangular partitionings drawn from the RLGAs. Intuitively, p controls the size of blocks. However, the role of the direction of growth is still unclear.

To visualize the role of the direction of growth, we show some samples drawn from the RLGAs. Recall that RLGA requires a direction of growth chosen from $\{\downarrow\rightarrow, \downarrow\leftarrow, \uparrow\rightarrow$

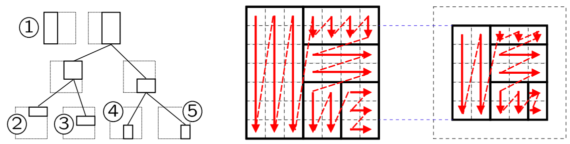

Figure 6. **Left:** Order of *boxes* generated from discrete Mondrian process and total order (e.g., $\searrow$). **Right:** Projectivity of parameter variables that leads to **conditional projectivity** of RTP, which is required for Orbanz's theorem.

$,\uparrow\leftarrow, \rightarrow\uparrow, \rightarrow\downarrow, \leftarrow\uparrow, \leftarrow\downarrow\}$. Fig. 5 shows the cases of $\rightarrow\downarrow$ and $\downarrow\rightarrow$. The direction of growth controls the entire block arrangement. In other words, specific choice of the direction of growth results in largely biased outputs, which is usually undesirable. Next we reduce this undesirable influence. Supplementary material provides more details.

3.3. Final construction as conditional projective limit

Strategy: To overcome the undesirable property of the naive construction discussed in the previous subsection, we construct a **conditionally projective** (Sec. 2.1.2.) family of rectangular partitionings by combining discrete MP (Sec. 2.2.) and RLGA; it leads to infinite partitionings according to Orbanz's theorem.

We now consider how to suppress the undesirable characteristic of the naive construction discussed in the previous subsection. The problem is the use of one RLGA with a specific direction of growth. Thus, to overcome this problem, roughly speaking, we first divide the input array into sub-arrays (called *boxes*), and then apply RLGAs with different (randomized) directions of growth to each of them. This idea is exactly two-step modeling, i.e., "conditioning" in the context of Bayesian modeling.

The key insight is that our strategy is to focus on the order of entries on which the merging/splitting decision is made. Each direction of growth in RLGA decides the order of the entries in "each box". Thus, to obtain a total order for the whole input array, we additionally require the "order of the boxes". That is, we first generate the order of the boxes, and then we independently generate a direction of growth for each box. This is followed by deciding the order of applying merging/splitting procedures for the entries in each box based on each direction of growth. Finally we obtain the total order of applying merging/splitting procedures.

To obtain a suitable order of boxes, we have to address the following conditions:

- We can apply RLGA only to a box where one upper/lower column at most and one right/left row at most have already been assigned to blocks. In other words, we cannot apply it to a box where more than two sides have already been assigned.

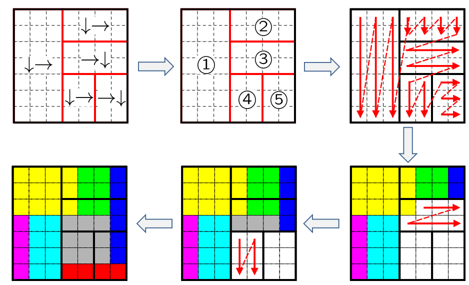

Figure 7. (Best viewed in color.) Flow of rectangular tiling process. (Clockwise from top left.) (1) The discrete Mondrian process first generates a binary space partitioning. Each box independently chooses a direction of growth. (2)-(3) Then, the order of applying merging/splitting procedures is determined. (4)-(5) The RLGAs sequentially generate partitionings. (6) The resulting object shows a sample of rectangular partitioning.

- The order of boxes should be projective. That is, if we restrict the input array to any sub-array, the restricted order of boxes must be preserved (Fig. 6).

dMP is sufficient for these conditions. We show a dMP-RLGA hierarchical model that satisfies the above conditions in Fig. 7. We first introduce a global direction from $\{\nearrow, \searrow, \swarrow, \nwarrow\}$ (typically, it is uniformly chosen in advance). As discussed below, this makes it possible to assign a randomized direction of growth to each box. Specifically, for example, when we chose $\searrow$, each box can choose, independently a direction of growth from $\{\rightarrow\downarrow, \downarrow\rightarrow\}$. Then, dMP generates a binary tree (binary space partitioning). Note that the leaves of the binary tree correspond to boxes. Along the binary tree, box orders are recursively generated as follows: Suppose that a current node of the binary tree is divided vertically/horizontally; We give priority to the side of the starting points of the arrow of the global direction, e.g., $\searrow$ means that we give priority to left or upper leaves. For all paths of the binary tree, we apply the above procedure. By construction, the above procedure must ensure that each untreated box has more than two residual neighboring sides in any step. All boxes can choose independently and uniformly their direction of growth from two candidates, e.g., for the case of $\searrow$, each box chooses it from $\{\rightarrow\downarrow, \downarrow\rightarrow\}$ (similarly, $\nearrow$: $\{\uparrow\rightarrow, \rightarrow\uparrow\}$, $\swarrow$: $\{\leftarrow\downarrow, \downarrow\leftarrow\}$, $\nwarrow$: $\{\leftarrow\uparrow, \uparrow\leftarrow\}$). Thus this hierarchical model suppresses the undesirable influence discussed in Sec. 3.2.

Finally Orbanz's theorem provides an infinite model as a conditional projective limit. In summary, our rectangular tiling process is based on the RLGAs conditioned by dMP (Fig. 7). The input consists of an input array, real value p $(0 < p < 1)$, and the global direction chosen from $\{\nearrow, \searrow, \swarrow, \nwarrow\}$. We can construct a family of $\mu^I(T_I|\Theta_I)$ for $I \in \mathcal{F}(E)$ as follows: Using dMP, we generate parameter variables Θ_I, which consist of a hierarchical partitioning of I, and directions of growth. According to Θ_I and the

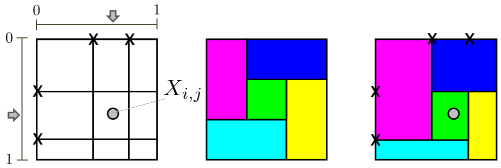

Figure 8. Illustration of RTP-based relational model. **Left:** First we make grid-style pre-clusters based on the product of Poisson processes on $[0,1]$, and generate the coordinates for each row/column from $\text{Uniform}[0,1]$. **Middle:** We then regard the pre-clusters ID's as a matrix, and apply the RTP to obtain a rectangular partitioning. **Right:** Finally each observation is generated from the assigned block.

global direction, the suitable order of RLGAs is automatically determined. Then, the sequential RLGAs generate rectangular partitioning $T_I \in \mathcal{T}_I$.

Theorem 3.4 *RTP uniquely defines probability measure $\mu^E(T_E|\Theta_E)$ as the conditional projective limit of the family $\mu^I(T_I|\Theta_I)$ $(I \in \mathcal{F}(E))$. Moreover, the height/width of each block is k with probability $p^{k-1}(1-p)$.*

4. Application: relational data analysis

We show an application of the proposed RTP to be used as the priors for SBMs. Note that RTP itself does not have the exchangeability of rows and columns similar to MP. Thus, we require additional models for the infinitely exchangeable permutations of the rows and the columns needed to obtain the RTP-based SBM. Obviously, it is preferable that the model leads to a tractable inference algorithm. Specifically, in the sense of Bayesian inference (e.g., Markov chain Monte Carlo methods or variational methods), it is preferable to perform (conditionally) independent updates for a partition and two permutations.

4.1. Relational model based on RTP

Strategy: We use a hierarchical structure. First we generate a grid-style partition, which leads to exchangeability of rows and columns similar to IRM. Then the grid-style partition is translated to a final rectangular partitioning by RTP, which belongs to the "arbitrary" class.

We here describe a Bayesian relational model based on the combination of RTP and the Aldous-Hoover representation (1981; 1979). Our strategy is to make grid-style clusters (called *pre-clusters*) based on the product of Poisson processes, and then to apply RTP to the pre-clusters. Fig. 8 provides an illustration. First, each row/column is represented as a vertical/horizontal coordinates in $[0,1]$, and two (vertical and horizontal) independent Poisson processes (PP) on $[0,1]$ divide the row and column coordinates into pre-clusters that provide a grid-style partition. For the observation matrix $(Y_{i,j})_{m\times n}$, the pre-cluster IDs of each row/column (denoted by ξ_i or η_j) are generated from the PP-based clustering (PPC): $\xi_i \sim \text{PPC}(\mu)$ $(i = 1,\ldots,m)$, $\eta_j \sim \text{PPC}(\mu)$ $(j = 1,\ldots,n)$, where μ denotes the rate parameter of PP. We use a matrix to represent the grid-style pre-cluster ID's. Note that we fix the ID's (we do not consider the permutations of the ID's). Thus, the permutations of the rows/columns of the observation are indirectly expressed as assignment of the fixed pre-cluster ID's. We then apply RTP to this matrix, consisting of the pre-cluster IDs, and obtain a rectangular partitioning: $\theta \sim \text{dMP}(\lambda) \times (\text{Bernoulli}(1/2))^{\#\text{leaf}}$, $T \mid \theta \sim \text{RLGAs}(p,\theta)$, where $\#\text{leaf}$ means the number of leaf nodes of the dMP. Finally, observation data is generated from the Dirichlet-categorical models: $\phi_r \sim \text{Dirichlet}(\boldsymbol{\alpha})$ $(r \in T)$, $Y_{i,j} \mid R, \boldsymbol{\xi}, \boldsymbol{\eta} \sim \text{Categorical}(\phi_{r_{i,j}})$, where $r_{i,j}$ denotes the block such that $(\xi_i, \eta_j) \in r_{i,j}$.

For Bayesian inference, we can use a Markov chain Monte Carlo (MCMC) method that iterates over draws from posteriors for rectangular partitioning T (Metropolis-Hastings), pre-clusters $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ (Gibbs), and intermediate variables θ (reversible jump (Wang et al., 2011)). See the supplementary material for details.

4.2. Experiments

For inference, we set the real variable $p = 0.7$, set the Mondrian budget $\lambda = 1$, and let the intensity of the PPs and the MP be Lebesgue measures. In practice, we found that it was better to increase the frequency of the Metropolis-Hastings (MH) updates for rectangular partitioning since MH has lower acceptance rate than Gibbs (100% acceptance) for row and column entries. Thus, we performed one MH update (for rectangular partitioning) and one Gibbs update (for one row and one column) per iteration. To examine the influence of MCMC initialization, we also employed 3 types of manually-generated regular grid partitionings as initialization: (7×7) (referred as RTPs), (15×15) (RTPm), and (30×30) (RTPl). We evaluated the models using perplexity: $\text{perp}(\hat{X}) = \exp(-(\log\, p(\hat{X}))/N)$, where N is the number of non-missing entries in $\hat{X}$. Roughly speaking, small perplexity means that the model fits the data better.

We used the following three real relational data sets: (a) **Animal feature** (50×85 binary data). We used a animal-feature matrix for 50 animals with 85 features (Kemp et al., 2006). (b) **Donations** (14×111 binary data). We used a political dataset for 14 countries with 111 binary features (Kemp et al., 2006). (c) **Cities** (55×46 categorical ($\in \{0,1,2,3\}$) data). This dataset consists of the distribution of offices for 46 service firms over 55 world cities. Service values for a firm in a city are given as 3, 2, 1 or 0 (Beaverstock et al., 2000).

Visualization of partitioning. (a) **Animal feature.** Fig.

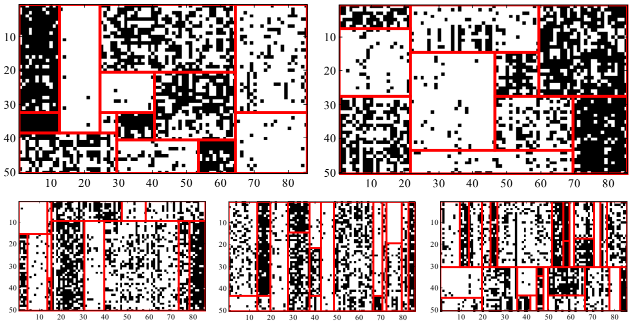

Figure 9. Animal-feature data analysis. (Supplementary material provides larger figures with animal-feature labels.) **Top:** two samples of RTP-based analysis. **Bottom:** three samples of MP-based analysis. The RTP provides more parsimonious explanations. As an example, for the right analysis of the RTP, two dense blocks on the right side indicate that {active, fast, smart} are included in a cluster for all animals, but {big, strong, group} are included in a cluster only for approximately half the animals.

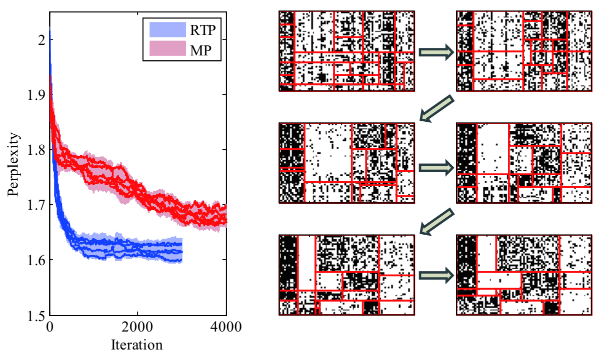

Figure 10. MCMC performances on animal-feature dataset. **Left:** Evolution of perplexity. Blue and red lines show 5 runs of the RTP- and MP-based model, respectively. **Right:** Evolution of RTP-based partitionings. (Left to right, top to bottom.) Each represents the sample on the 50/100/200/500/1000/2000-th iteration.

9 shows samples of rectangular partitionings for the animal dataset, and Fig. 10 (left) plots the training perplexity evolution for 5 RTP runs and 5 MP runs. Fig. 10 (right) represents an example of the MCMC evolution of RTP-based partitioning. Fig. 11 shows perplexity and number of blocks of 10 RTPs runs, 10 RTPm runs, 10 RTPl runs, and 10 MP runs. For each run, we focused on the sample that obtained the highest likelihood. As Fig. 11 shows, RTP tends to find partitionings that have a smaller number of blocks than MP with similar perplexity, or to use a similar number of blocks to MP with better perplexity than MP. Supplementary material provides the visualizations of b) **Nations** and (c) **Cities**.

Perplexity comparison on test datasets. For model comparison, we held out 20% of the data for testing. Table 1 lists the average perplexity over 5 runs, with the standard deviation of each average given in parentheses. For all

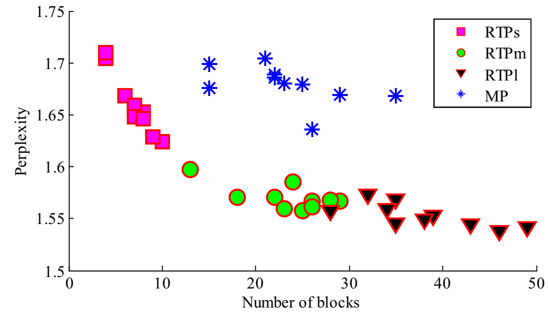

Figure 11. Number of blocks and perplexity of RTP and MP. RTPs (red square) typically find partitionings that have a smaller number of blocks than MP (blue asterisk), even though RTPs match the perplexity of MP. RTPm (green circle) use a similar number of blocks to MP. However, RTPm fits the data better than MP in the sense of perplexity. Although RTPl uses many blocks to obtain better perplexity, the final experiment (test perplexity comparison) implies that it avoids over-fitting.

datasets, our primitive sampler for RTP shows at least comparable performance to the reversible jump MCMC for MP. However, the performance depends on the initialization of MCMC, which should be improved in the near future.

Table 1. Perplexity comparison on test datasets

	Animal	Dnations	Cities
MP	1.806 (0.032)	1.858 (0.000)	2.582 (0.138)
RTPs	1.749 (0.070)	**1.840** (0.029)	2.560 (0.095)
RTPm	1.741 (0.049)	1.913 (0.086)	**2.495** (0.218)
RTPl	**1.688** (0.061)	2.367 (0.266)	2.783 (0.154)

5. Discussion

One of the new generation of Bayesian nonparametrics must involve array- and graph-valued random variables (Lloyd et al., 2012; Choi & Wolfe, 2014). It also involves classical Aldous-Hoover theorem (1981; 1979) and recent graph limit theory (Lovász, 2009; Airoldi et al., 2013). We believe that this paper provides a significant contribution in the context of rectangular partitionings. Moreover, our strategy involves Orbanz's extension theorem beyond Kolmogorov's well-known extension theorem, which will lead to various new stochastic processes in the near future.

One of the most important future directions is to construct more sophisticated inference methods. Our primitive sampler requires more computation time than MP, since it includes MP inference as a subroutine, and the performance strongly depends on the initialization. We are currently interested in improving MCMC schemes for the RTP-based relational model by combining the essence of recent methods for combinatorial problems, including measure factorization (Bouchard-Côté & Jordan, 2010), and MCMC via bridging (Lin & Fisher, 2012).

References

Airoldi, E. M., Costa, T. B., and Chan, S. H. Stochastic blockmodel approximation of a graphon: Theory and-consistent estimation. In *Advances in Neural Information Processing Systems*, 2013.

Aldous, D. J. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11:581–598, 1981.

Beaverstock, J. V., Smith, R. G., Taylor, P., Walker, D. R. F., and Lorimer, H. Relational studies of globalization and world cities: Three measurement methodologies. *Applied Geography*, (20), 2000.

Bochner, S. *Harmonic analysis and the theory of probability*. University of California Press, 1955.

Bouchard-Côté, A. and Jordan, M. Variational inference over combinatorial spaces. In *Advances in Nueral Information Processing Systems*, 2010.

Burridge, J., Cowan, R., and Ma, I. Full and half Gilbert tessellations with rectangular cells. *Advances in Applied Probability*, 1:1–19, 2013.

Choi, D. S. and Wolfe, P. J. Co-clustering separately exchangeable network data. *Annals of Statistics*, 42:29–63, 2014.

Crane, H. *Infinitely exchangeable partition, tree and graph-valued stochastic process*. PhD thesis, Department of Statistics, The University of Chicago, 2012.

Gilbert, E. N. *Surface films of needle-shaped crystals*. Applications of Undergraduate Mathematics in Engineering, noble, macmillan edition, 1967.

Hoover, D. N. Relations on probability spaces and arraysof random variables. Technical report, Institute of Advanced Study, Princeton, 1979.

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., and Ueda, N. Learning systems of concepts with an infinite relational model. In *Proceedings of the National Conference onArtificial Intelligence*, 2006.

Lin, D. and Fisher, J. Efficient sampling from combinatorial space via bridging. In *International Conference on Artificial Intelligence and Statistics*, 2012.

Lloyd, J., Orbanz, P., Ghahramani, Z., and Roy, D. M. Random function priors for exchangeable arrays with applications to graphs and relational data. In *Advances in Neural Information Processing Systems*, 2012.

Lovász, L. Very large graphs. *Current Developments in Mathematics*, 11:67–128, 2009.

Mackisack, M. S. and Miles, R. E. Homogeneous rectangular tessellation. *Advances on Applied Probability*, 28: 993, 1996.

Muthukrishnan, S., Poosala, V., and Suel, T. On rectangular partitionings in two dimensions: algorithms, complexity, and applications. In *The International Conference on Database Theory*, 1999.

Nowicki, K. and Snijders, T. A. B. Estimation and prediction for stochastic block structures. *Journal of the American Statistical Association*, 96:1077–1087, 2001.

Orbanz, P. *Infinite-dimensional exponential families in the cluster analysis of structured data*. PhD thesis, Eidgenössische Technische Hochschule Zürich, 2008.

Orbanz, P. Construction of nonparametric Bayesian models from parametric Bayes equations. In *Advances in Neural Information Processing Systems*, 2009.

Orbanz, P. Conjugate projective limits. Technical report, arXiv, 2011.

Orbanz, P. and Roy, D. M. Bayesian models of graphs, arrays and other exchangeable random structures. Technical report, arXiv, 2013.

Roy, D. M. *Computability, inference and modeling in probabilistic programming*. PhD thesis, Massachusetts Institute of Technology, 2011.

Roy, D. M. and Teh, Y. W. The Mondrian process. In *Advances in Neural Information Processing Systems*, 2009.

Wang, P., Laskey, K. B., Domeniconi, C., and Jordan, M. I. Nonparametric bayesian co-clustering ensembles. In *SIAM International conference on Data Mining*, 2011.

Wasserman, S. and Anderson, C. Stochastic a posteriori block models: Construction and assessment. *IBM Journal of Research and Development*, 9(1):1–36, 1987.