The Ostomachion Process

Xuhui Fan, Bin Li, Yi Wang, Yang Wang, Fang Chen
Machine Learning Research Group, National ICT Australia, Eveleigh, NSW 2015, Australia
{xuhui.fan, bin.li, yi.wang, yang.wang, fang.chen}@nicta.com.au

Abstract

Stochastic partition processes for exchangeable graphs produce axis-aligned blocks on a product space. In relational modeling, the resulting blocks uncover the underlying interactions between two sets of entities of the relational data. Although some flexible axis-aligned partition processes, such as the Mondrian process, have been able to capture complex interacting patterns in a hierarchical fashion, they are still in short of capturing dependence between dimensions. To overcome this limitation, we propose the Ostomachion process (OP), which relaxes the cutting direction by allowing for oblique cuts. The partitions generated by an OP are convex polygons that can capture inter-dimensional dependence. The OP also exhibits interesting properties: 1) Along the time line the cutting times can be characterized by a homogeneous Poisson process, and 2) on the partition space the areas of the resulting components comply with a Dirichlet distribution. We can thus control the expected number of cuts and the expected areas of components through hyper-parameters. We adapt the reversible-jump MCMC algorithm for inferring OP partition structures. The experimental results on relational modeling and decision tree classification have validated the merit of the OP.

Introduction

Stochastic partition processes for exchangeable graphs have found broad applications ranging from relational modelling (Kemp et al. 2006; Airoldi et al. 2009), community detection (Nowicki and Snijders 2001; Karrer and Newman 2011), collaborative filtering (Porteous, Bart, and Welling 2008), to random forests (Lakshminarayanan, Roy, and Teh 2014). Most work on the stochastic partition process only considers axis-aligned cuts and the resulting partitions form regular grids (rectangular blocks). In relational modeling, these blocks are able to capture the underlying interactions between two sets of entities of the relational data. The recent advances in irregular grid partitions have introduced more flexibility, such as the Mondrian process (Roy and Teh 2009) and the rectangular tiling process (Nakano et al. 2014b).

Despite the success of these stochastic partition processes in uncovering complex interacting patterns, they are still in short of capturing the dependence between dimensions due to the restriction of axis-aligned cuts. The axis-aligned partitions are based on the assumption that two sets of interacted entities have the same intensity. This is an over-simplified assumption in many scenarios. Take relational modeling of emails in a company for example, a rectangular block implies that the staff involved play an equal role to one another; while in reality staff may play asymmetric roles such that the partition may exhibit a triangular block (e.g., leaders may send emails to many staff while interns may only send emails to her mentor). The limitation of axis-aligned cuts is more obvious if the partition structure is used as a prior for a decision tree, where the decision boundaries are usually a linear combination of multiple dimensions.

In this paper, we relax the axis-aligned partitions by allowing for oblique cuts on a product space (the unit square). Since the resulting components are convex polygons, which resemble a dissection puzzle Ostomachion, the proposed stochastic partition process is named the Ostomachion process (OP). Through this relaxation, the two dimensions can be considered simultaneously to capture more complex partition structures with inter-dimensional dependence (see example in Figure 1).

An OP is generated by recursively bi-partitioning the leaf components (polygons) on the product space with oblique cuts. For each oblique cut, its slope and position are random variables drawn from certain distributions and the OP can thus have the following properties: 1) The times of the cutting events along the time line comply with a homogeneous Poisson process and 2) the areas of the leaf components on the unit square comply with a Dirichlet distribution. Due to these two properties, we can easily control the expected number of cuts and the expected areas of components through hyper-parameters of the OP process.

We adapt the reversible-jump MCMC algorithm (Green 1995) for inferring the partition structure of an OP with three types of cutting operations. In addition to “cut-adding” and “cut-removing”, a new type of proposal “cut-translation” is introduced which can help alleviate inferior local optima and reduce the inference variance.

We demonstrate the advantages of the OP in two applications. Firstly, we apply the OP as a partition prior for relational modeling. The visualization of the partition results and the link prediction performance have validated the merit of oblique cuts in the OP. Secondly, we use the OP partition structure to construct a decision tree classifier and demonstrate its powerful separability against axis-aligned partition structures. The experimental results in both applications show that the OP is more flexible and effective compared to the classical axis-aligned partition processes.

Related Work

Regular-grid stochastic partition processes are constituted by two separate partition processes on each dimension. Due to the separate partition processes, the resulting orthogonal interactions from two sides will exhibit regular grids, whose densities represent the intensities of interactions. Typical regular-grid partition models include infinite relational models (IRM) (Kemp et al. 2006) and mixed-membership stochastic blockmodels (MMSB) (Airoldi et al. 2009). Regular-grid partition models are widely used in real-world applications for modeling graph data (Li, Yang, and Xue 2009; Ishiguro et al. 2010; Ho, Parikh, and Xing 2012; Schmidt and Morup 2013).

To our knowledge, only the Mondrian process (MP) (Roy and Teh 2009) and the rectangular tiling process (RTP) (Nakano et al. 2014b) are able to produce arbitrary grid partitions. MP is a generative process that recursively generates axis-aligned cuts in a unit hypercube. In contrast to stochastic block models, MP can partition the space in a hierarchical fashion known as kd-trees and result in irregular block structures. An MP on the unit square (2-dimensional product space) is started from a random axis-aligned cut on the perimeter and results in two rectangles, in each of which a random cut is made in the same way and so forth. Before cutting on a rectangle, a cost E is drawn from an exponential distribution Exp(perimeter); if $\lambda - E < 0$ (λ is the budget), the recursive procedure halts; otherwise, a random cut is made on the half perimeter of the rectangle and let $\lambda = \lambda - E$. In this way, a larger λ will result in more cuts. There are some interesting extensions and applications of the MP, such as metadata dependent Mondrian Processes (Wang et al. 2015), the ecological network reconstruction (Aderhold, Husmeier, and Smith 2013) and the hidden Markov model (Nakano et al. 2014a). Different from MP based on kd-tree, RTP generates arbitrary rectangular partitions based on projective systems.

The Ostomachion Process

In this section, we introduce the generative process of the Ostomachion process (OP) and show two favorable properties of the OP: One for characterizing the times of cutting events (see Property 1) and the other for characterizing the areas of the leaf components (see Property 2). Note that we use "components" instead of "blocks" since in the OP they are convex polygons.

The Generative Process

The Ostomachion process recursively generates oblique cuts on a unit square[1], with the cutting events arriving along the time line. An OP is denoted as

$$\mathcal{O} \sim OP(\tau, \alpha, [0,1]^2) \tag{1}$$

where τ denotes the time limit[2], which controls the number of cuts in an OP; and α is a concentration parameter, which controls the skewness of the area distribution of the components. The first cut is generated on the unit square and the subsequent cuts are generated in the existing components (polygons). The cutting process proceeds recursively and finally produces a hierarchical partition structure on $[0,1]^2$, on which each leaf component is a polygon. An example OP is illustrated in Figure 1.

To generate an oblique cut in an existing component $\triangle_k$ (the first cut is conducted on the unit square), we first uniformly sample a variable θ_k from $[0, 2\pi]$ to determine the slope of the cut. Then we sample a beta distributed random variable $\gamma_k \sim \text{Beta}(\alpha/2^{d_k}, \alpha/2^{d_k})$, where d_k denotes the depth of $\triangle_k$ in the bi-partition tree structure (the root level $d_k = 1$ is on the unit square). The proposed θ_k-sloped cut is placed on $\triangle_k$ such that the area ratio of the two resulting sub-components satisfies $\gamma_k/(1-\gamma_k)$ (see Figure 2).

Meanwhile, the proposed cut is associated with a waiting time t_k, which is sampled from an exponential distribution with the rate parameter being the area of the component $t_k \sim \text{Exp}(A(\triangle_k))$, where $A(\triangle_k)$ denotes the area of $\triangle_k$. If t_k exceeds the rest time $\tau - \sum_{j \in pre(k)} t_j$, where $pre(k)$ denotes all the predecessor components of $\triangle_k$ in the bi-partition tree structure, the recursive cutting process halts in that branch; otherwise, the proposed cut is accepted and $\triangle_k$ is split into two polygons $\triangle_{k'}$ and $\triangle_{k''}$. Two separate cutting processes continue in $\triangle_{k'}$ and $\triangle_{k''}$ respectively: $\mathcal{O}' \sim OP(\tau - \sum_{j \in pre(k')} t_j, \alpha/2^{d_{k'}+1}, \triangle_{k'})$ and $\mathcal{O}'' \sim OP(\tau - \sum_{j \in pre(k'')} t_j, \alpha/2^{d_{k''}+1}, \triangle_{k''})$. The final partition structure on the unit square is returned when the cutting processes along all the branches of the bi-partition tree reach the time limit τ.

Convex Polygon Components In an OP, all the resulting components are convex polygons. This can be verified by investigating the new angles produced by a cut in the component: All these new angles lie in $[0, \pi]$ and, consequently, the resulting polygons do not contain angles larger than π. This feature enables the OP to capture dependence between two dimensions in individual components and introduces more flexibility in relational modeling compared to the axis-aligned partition processes (Kemp et al. 2006; Roy and Teh 2009).

Partition Prior of Exchangeable Graphs Like the MP (Roy and Teh 2009), the OP can also be a partition prior for exchangeable graphs. The permutation of rows/columns of a graph does not affect the joint probability conditioned on the graphon (Lloyd et al. 2012; Orbanz and Roy 2015) (graph function) $W : [0,1]^2 \to [0,1]$, which is determined

[1] The generative process can be straightforwardly extended to a multi-dimensional product space. For simplicity, we only discuss the 2-dimensional case in this paper.

[2] The time limit τ is analogous to the budget λ in the MP (Roy and Teh 2009).

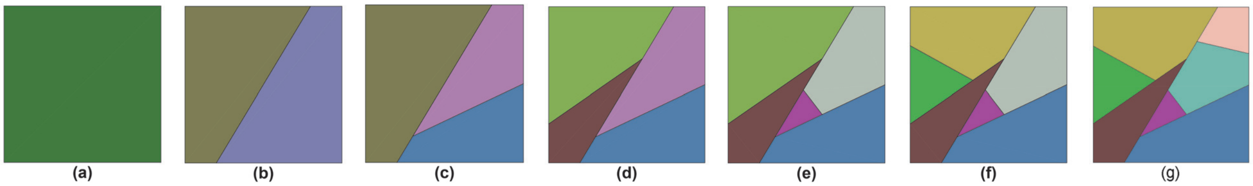

Figure 1: Recursively generate oblique cuts in an example Ostomachion process.

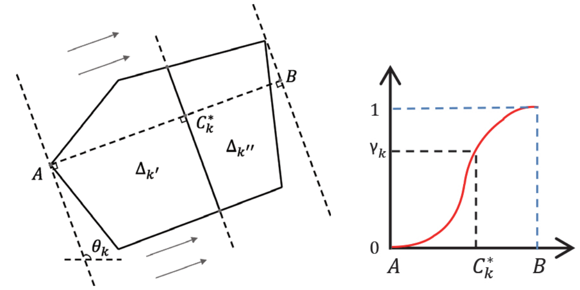

Figure 2: (Left) Generate a θ_k-sloped cut on a component $\triangle_k$ such that the area ratio $A(\triangle_{k'})/A(\triangle_{k''}) = \gamma_k/(1-\gamma_k)$. (Right) The cutting position C_k^* is determined by the area ratio function (red curve), which is usually a step-wise polynomial function describing how the area ratio γ_k changes along the cutting position C_k^*.

by an OP $\mathcal{O} = \{\triangle_k\}$ and the intensity rates for each component in $\mathcal{O}$. The graphon W is a two-dimensional piece-wise constant function and each intensity rate occupies a convex polygon $\triangle_k$ (see Figure 3).

Property of the Cutting Time

In an OP, each proposal of a cut is associated with a waiting time $t_k \sim \text{Exp}(A(\triangle_k))$. By using the area of the component as the parameter of the waiting time distribution, we can have the following property for the cutting events:

Property 1. *In an OP, the times of the cutting events along the time line can be characterized by a homogeneous Poisson process, whose intensity rate is the area of the unit square* $A([0,1]^2)$.

Suppose the current partition structure on the unit square contains K components (polygons) $\triangle_1, \ldots, \triangle_K$. The waiting time for generating a cut in $\triangle_k$ is independently distributed as $t_k \sim \text{Exp}(A(\triangle_k))$. The minimum waiting time t^* among all the components $\triangle_1, \ldots, \triangle_K$ follows the distribution

$$t^* = \min(t_1, \ldots, t_K) \sim \text{Exp}(A([0,1]^2)) \qquad (2)$$

This is because $\Pr(t^* > t) = \Pr(t_1 > t, \ldots, t_K > t) = \prod_{k=1}^K \Pr(t_k \geq t) = \prod_{k=1}^K \exp(-tA(\triangle_k)) = \exp\left(-t\sum_{k=1}^K A(\triangle_k)\right) = \exp\left(-tA([0,1]^2)\right)$, which is the complementary cumulative distribution function of t^*. That is to say, the waiting time for the next cut in $\bigcup_{k=1}^K \triangle_k$ is also distributed as that for the first cut on the unit square, i.e., $t_{next} \sim \text{Exp}(A([0,1]^2))$. Thus, the waiting time of each next cut in an OP is independent to the current partition structure. The arrival times of cutting events in an OP form a homogeneous Poisson process, with the intensity rate being the area of the unit square.

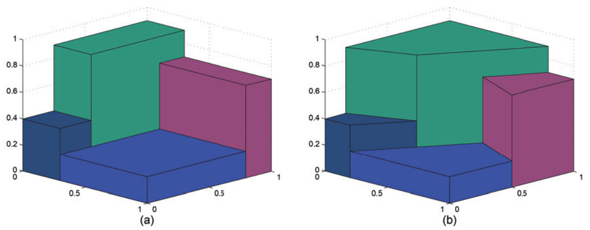

Figure 3: Illustration of graphons for the MP (Roy and Teh 2009) (left) and the OP (right), respectively.

The above result implies that each new cut would be assigned to one of the existing components with a probability proportional to its area; furthermore, the cutting events in each component $\triangle_k$ individually forms a Poisson process with the intensity rate being its own area. The expected number of cuts $N(\triangle_k)$ in $\triangle_k$ equals to the intensity rate along the time line $\mathbb{E}[N(\triangle_k)] = (\tau - \sum_{j\in pre(k)} t_j)A(\triangle_k)$; thus the expected number of cuts in an OP on the unit square is $\mathbb{E}[N([0,1]^2)] = \tau A([0,1]^2) = \tau$.

Figure 4 illustrates the Poisson process of the cutting events along the time line. In the right panel, the solid black lines denote the waiting time intervals of the cutting events and the points on the time line denote the arrival times of the three cutting events. In this example, the intensity rate of the Poisson process is consistent at any time point: $\nu_1(t \in [O,A]) = \nu_2(t \in [A,B]) + \nu_3(t \in [A,B]) = \nu_4(t \in [B,C]) + \nu_5(t \in [B,C]) + \nu_3(t \in [B,C]) = 1$, which is

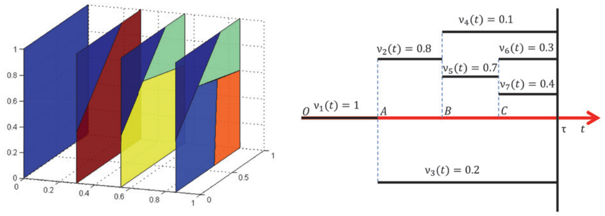

Figure 4: (Left) The cutting events of an OP comply with a homogeneous Poisson process along the time line. (Right) Each cut's waiting time is exponentially distributed with parameter being the area of the unit square $A([0,1]^2)$.

the area of the unit square. It is worth noting that each of the branches of the bi-partition tree structure is also a Poisson process.

Property of the Partition Structure

Although the components $\{\triangle_k\}$ of an OP are generated in a recursive bi-partition fashion, the resulting leaf components have the following interesting property:

Property 2. *In an OP, the areas of the leaf components* $\{A(\triangle_1), \cdots, A(\triangle_K)\}$ *comply with a Dirichlet distribution*

$$[A(\triangle_1), \cdots, A(\triangle_K)] \sim Dir(\alpha_1, \cdots, \alpha_k) \tag{3}$$

where α_k *denotes the concentration parameter for the k-th component and* $\sum_k \alpha_k = \alpha$.

Given α, the first cut partitions the unit square into two polygons whose areas follow $\mathrm{Dir}(\frac{\alpha}{2}, \frac{\alpha}{2})$ (because $\gamma \sim \mathrm{Beta}(\frac{\alpha}{2}, \frac{\alpha}{2})$ and $A([0,1]^2) = 1$). W.l.o.g. assuming the next cut occurs in the first polygon, the cut ratio $\gamma_1 \sim \mathrm{Beta}(\frac{\alpha_1}{2}, \frac{\alpha_1}{2})$, where $\alpha_1 = \frac{\alpha}{2}$ by definition. Let $s_{1,1}, s_{1,2}, s_2$ denote the areas of the three leaf components in the current partition structure, their joint distribution follows

$$\begin{aligned} p(s_{1,1}, s_{1,2}, s_2) &= p(s_1, s_2)p(\gamma_1) \cdot \left| \frac{\partial(s_1, \gamma_1)}{\partial(s_{1,1}, s_{1,2})} \right| \\ &= \frac{\Gamma(\frac{\alpha}{2}\beta_1 + \frac{\alpha}{2}\beta_2 + \frac{\alpha}{2})}{\Gamma(\frac{\alpha}{2}\beta_1)\Gamma(\frac{\alpha}{2}\beta_2)\Gamma(\frac{\alpha}{2})} s_{1,1}^{\frac{\alpha}{2}\beta_1 - 1} s_{1,2}^{\frac{\alpha}{2}\beta_2 - 1} s_2^{\frac{\alpha}{2} - 1} \end{aligned} \tag{4}$$

where $\beta_1 + \beta_2 = 1$. We can see that Eq. (4) is a Dirichlet distribution $\mathrm{Dir}(\frac{\alpha}{2}\beta_1, \frac{\alpha}{2}\beta_2, \frac{\alpha}{2})$. Thus Property 2 can be verified if we recursively apply the above operation to each subsequent cutting event.

For simplicity we let $\beta_1 = \beta_2 = \frac{1}{2}$, which are sufficient for generating flexible partitions. Consequently, we only have one hyper-parameter α to control the expected partition structure: A larger concentration parameter α will result in more evenly distributed areas of the leaf components.

The above construction resembles the stick-breaking construction of the Dirichlet process (Sethuraman 1994), which is to break the unit stick into "left"-biased infinite segments (keeps the left part unchanged and break the right part recursively); while the stick-breaking construction of the OP is to recursively break the stick into two parts.

Inference

In the following, an approximate inference algorithm based on reversible-jump MCMC (Green 1995) is introduced for inferring OP partition structures. We define three types of cutting operations, which can be proposed by the algorithm to change the state space of the partition structure.

Cutting Operations

An OP partition structure on the unit square can be represented by a set of cuts $\{t_j, \theta_j, C_j^*\}_j$, where t_j denotes the waiting time to generate the jth cut, θ_j and C_j^* denote the slope and the location of the cut, respectively.

To infer the partition structure generated by an OP, we need to define the state transition operations between any pair of partition states, such that a partition can be transformed to another in the inference procedure. We propose the following three cutting operations:

- Cut-adding ψ_{add} adds a cut $(t_{j'}, \theta_{j'}, C_{j'}^*)$ in a uniformly sampled component. This operation can be written as $\psi_{add}(\{t_j, \theta_j, C_j^*\}_j, j') = \{t_j, \theta_j, C_j^*\}_j \cup (t_{j'}, \theta_{j'}, C_{j'}^*)$.
- Cut-removing ψ_{rem} deletes a leaf cut $(t_{j'}, \theta_{j'}, C_{j'}^*)$ ("leaf cuts" refer to the cuts generating two leaf components) from the existing partition. As a result, the two corresponding sibling components are merged and returned to their parent component. This operation can be written as $\psi_{rem}(\{t_j, \theta_j, C_j^*\}_j, j') = \{t_j, \theta_j, C_j^*\}_{j \neq j'}$.
- Cut-translation ψ_{tra} adjusts an existing leaf cut $(t_{j'}, \theta_{j'}, C_{j'}^*)$ by resampling $\theta_{j'}$ and $\gamma_{j'}$ (thus $C_{j'}^*$). This operation can be written as $\psi_{tra}(\{t_j, \theta_j, C_j^*\}_j, j') = \{t_j, \theta_j, C_j^*\}_{j \neq j'} \cup (t_{j'}, \theta_{j'}, C_{j'}^*)$. This operation can make the best use of the existing cuts.

By applying a combination of cutting operations, any two partition structures, $\mathcal{O} = \{t_j, \theta_j, C_j^*\}_j$ and $\mathcal{O}' = \{t_{j'}, \theta_{j'}, C_{j'}^*\}_{j'}$, can be transformed to each other by *sequentially* performing the three types of cutting operations

$$\begin{aligned} &\{t_{j'}, \theta_{j'}, C_{j'}^*\}_{j'} = \\ &\psi_{add}(\psi_{tra}(\psi_{rem}(\{t_j, \theta_j, C_j^*\}_j, \mathcal{S}_{rem}), \mathcal{S}_{tra}), \mathcal{S}_{add}) \end{aligned}$$

where $\mathcal{S}_{rem}$, $\mathcal{S}_{tra}$, and $\mathcal{S}_{add}$ denote the sets of removed cuts, translated cuts, and added cuts, respectively. The optimal sets that minimize the number of cutting operations can be determined by the intersection of the two partition structures: $\mathcal{S}_{rem} = \{j | j \in \mathcal{O} \wedge j \notin \mathcal{O}'\}$, $\mathcal{S}_{tra} = \{j | j \in \mathcal{O} \cap \mathcal{O}' \wedge (\theta_j, C_j^*) \neq (\theta_{j'}, C_{j'}^*)\}$, and $\mathcal{S}_{add} = \{j | j \notin \mathcal{O} \wedge j \in \mathcal{O}'\}$. An example of partition structure transform is illustrated in Figure 5(a–d).

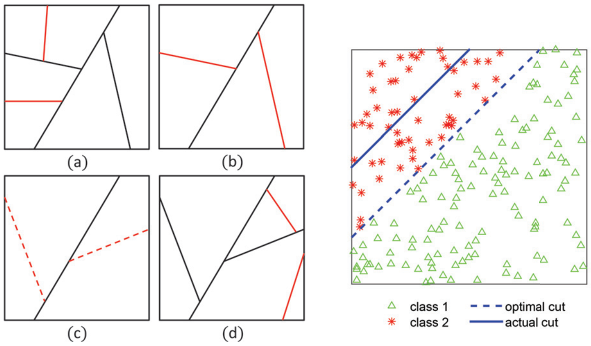

Figure 5: (Left) A partition structure transform via two cut-removing, two cut-translation, and two cut-adding operations in sequence. (Right) An example illustrates the necessity of the cut-translation operation. Since the likelihood ratio for removing the actual cut is very small (9.3×10^{-14}), it has very low probability to accept the cut-removing proposal such that the improper cut could be hardly rectified.

Reversible-Jump MCMC

We adopt the reversible-jump MCMC (Green 1995) algorithm for inferring OP partition structures with three types of

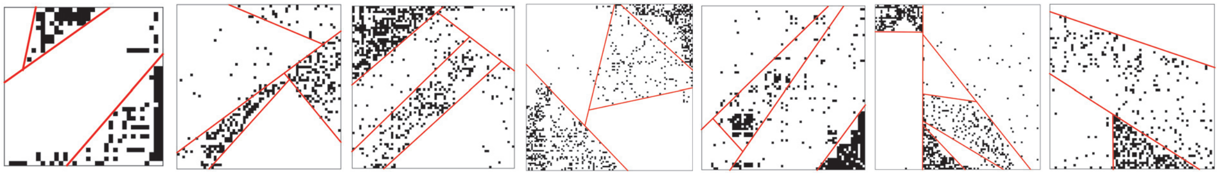

Figure 6: Partition results of the OP on (from left to right) *Foodweb, Dolphin, Lazega, Polbooks, Train, Reality*, and *Wikitalk*.

state-transition proposals, which correspond to three types of cutting operations. From the state of the current partition structure $\mathcal{O}_t$, one of the three types of cutting operations is uniformly selected to perform on $\mathcal{O}_t$, with acceptance ratio ρ the partition structure is transformed to the next state $\mathcal{O}_{t+1}$.

The inference procedure is briefed in Algorithm 1. In the algorithm, $\mathcal{O}_t$ denotes the current partition structure, $\boldsymbol{\zeta}_t$ denotes the parameter set of the current state, $u^{(1:3)}$ refer to the three new generated variables $\{t_k, \theta_k, \gamma_k\}$, $|\cdot|$ denotes the Jacobian matrix; $p(\mathcal{O}_{t+1}|\tau)$, $\mathcal{L}(Y|\mathcal{O}_{t+1}, \xi, \eta, \phi)$, and $Q(\mathcal{O}_{t+1}|\mathcal{O}_t)$ denote the prior probability, the likelihood function, and the proposal probability, respectively.

Algorithm 1 RJMCMC for the Ostomachion process

```
Input: Initial partition $[0,1]^2$, time limit $\tau$, concentration
    parameter $\alpha$, observed data $Y$, iteration number $T$
Output: An OP partition structure $\mathcal{O}$
  Generate the parameter $\boldsymbol{\phi}_k$ of the likelihood distribution
  for each component $\triangle_k$ in $\mathcal{O}_0$
  for $t = 1 : T$ do
    Uniformly propose one of the following actions with
    the acceptance ratio $\min(1, \rho)$
    • Cut-adding: $\rho = \frac{p(\mathcal{O}_{t+1}|\tau)\mathcal{L}(Y|\mathcal{O}_{t+1},\xi,\eta,\phi)}{p(\mathcal{O}_t|\tau)\mathcal{L}(Y|\mathcal{O}_t,\xi,\eta,\phi)} \times \frac{Q(\mathcal{O}_t|\mathcal{O}_{t+1})q(u^{(1:3)})}{Q(\mathcal{O}_{t+1}|\mathcal{O}_t)} \times \left|\frac{\partial\langle\boldsymbol{\zeta}_{t+1}\rangle}{\partial\langle\boldsymbol{\zeta}_t, u^{(1:3)}\rangle}\right|$
    • Cut-removing: $\rho = \frac{p(\mathcal{O}_{t+1}|\tau)\mathcal{L}(Y|\mathcal{O}_{t+1},\xi,\eta,\phi)}{p(\mathcal{O}_t|\tau)\mathcal{L}(Y|\mathcal{O}_t,\xi,\eta,\phi)} \times \frac{Q(\mathcal{O}_t|\mathcal{O}_{t+1})}{Q(\mathcal{O}_{t+1}|\mathcal{O}_t)q(u^{(1:3)})} \times \left|\frac{\partial\langle\boldsymbol{\zeta}_{t+1}, u^{(1:3)}\rangle}{\partial\langle\boldsymbol{\zeta}_t\rangle}\right|$
    • Cut-translation: $\rho = \frac{p(\mathcal{O}_{t+1}|\tau)\mathcal{L}(Y|\mathcal{O}_{t+1},\xi,\eta,\phi)}{p(\mathcal{O}_t|\tau)\mathcal{L}(Y|\mathcal{O}_t,\xi,\eta,\phi)}$
    Update the parameter $\boldsymbol{\phi}_k$ of the likelihood distribution
    for each component $\triangle_k$ in $\mathcal{O}_t$.
  end for
```

It is worth noting that a cut-translation operation in Algorithm 1 is not a simple combination of cut-removing and cut-adding. As an example shown in Figure 5(Right), the proposal of cut-removing has a probability of almost 0 (which happens, for example, when a cut only partially separates the same class of data from the whole data). In contrast, a cut-translation operation can propose to rectify the improper cut, instead of removing, and determines if accepting the rectification based on the ratio of their posterior probabilities. In this way, improper cuts are very likely to be adjusted and the issue of local optimum partition structures can also be largely alleviated.

Inference Complexity The OP retains the same inference complexity as that of the MP (Wang et al. 2011). Each proposal in the MP requires to sample two variables for cost and cut location; while the OP requires to sample three variables for time limit, cut slope and area ratio. Since the complexity of ratio computation is the same for both in each iteration, the computation complexity of the OP is 3/2 times of that of the MP given the same number of iterations.

Applications

We apply the proposed Ostomachion process to two applications: 1) Relational modelling and 2) Decision tree-style classification. We adopt MCMC inference algorithms for all the compared methods. In particular, we let RJMCMC-2 denote the inference algorithm used in (Wang et al. 2011), which only involves two types of operations (cut-adding and cut-removing); and let RJMCMC-3 denote Algorithm 1 introduced above.

In our experiments, we set the time limit $\tau = 5$ (meaning that the expected number of cuts is 5) and the concentration parameter $\alpha = 10$. We set $T = 200$ (200 iterations) for both RJMCMC-2 and RJMCMC-3. The performance of the two applications is evaluated by averaging the prediction on 10 randomly selected (in a ratio of $1/10$) hold-out test sets.

The Ostomachion Relational Model

We apply the OP to relational modeling. The generative process of the Ostomachion relational model (ORM) is as follows

$$\begin{aligned} &\mathcal{O} \sim \text{OP}(5, 10, [0,1]^2); \quad &&\phi_k \sim \text{Beta}(1,1); \\ &\xi_i, \eta_j \sim \text{Uniform}[0,1]; \quad &&Y_{ij} \sim \text{Bernoulli}(\phi_{\hbar(\xi_i,\eta_j)}) \end{aligned} \tag{5}$$

for $k \in \{1, 2, \cdots\}, i, j \in \{1, \cdots, n\}$, where Y denotes the $n \times n$ graph data and $Y_{ij} \in \{0, 1\}$ denotes the link between nodes i and j; ϕ_k denotes the parameter of the beta distribution in the kth component on $\mathcal{O}$, and $\hbar(\xi_i, \eta_j)$ denotes the mapping function from sending and receiving indices (ξ_i, η_j) to the corresponding component index.

In the scenario of social network analysis, Eq. (5) can be interpreted as: An OP partitions the unit square into a number of components; each component $\triangle_k$ represents a community in the social network with the intensity of interaction being ϕ_k, which is independently generated from a beta distribution. Then, sending and receiving indices (ξ_i, η_j) are uniformly generated from $[0, 1]$. For each pair of nodes i and j, it is first mapped to the corresponding component by $\hbar(\xi_i, \eta_j)$ and a link Y_{ij} is generated according to the Bernoulli distribution whose parameter is $\phi_{\hbar(\xi_i,\eta_j)}$.

Table 1: Relational modeling results in AUC (standard deviation)

Data	N.[1]	L.[1]	IRM	Mondrian Relational Model		Ostomachion Relational Model	
				RJMCMC-2	RJMCMC-3	RJMCMC-2	RJMCMC-3
Foodweb (Lichman 2013)	35	160	0.816(0.026)	0.841(0.021)	0.834(0.021)	0.823(0.024)	**0.845**(0.031)
Dolphin (Lichman 2013)	62	318	0.737(0.021)	0.767(0.029)	0.785(0.021)	0.757(0.020)	**0.789**(0.051)
Lazega (Lichman 2013)	71	680	0.746(0.017)	0.743(0.032)	0.776(0.094)	0.746(0.019)	**0.782**(0.029)
Polbooks (Lichman 2013)	105	882	0.752(0.009)	0.702(0.053)	0.692(0.046)	0.772(0.023)	**0.801**(0.022)
Train (Lichman 2013)	70	486	0.767(0.012)	0.734(0.048)	0.738(0.045)	0.746(0.016)	**0.822**(0.039)
Reality (Leskovec et al. 2010)	94	600	0.879(0.018)	0.884(0.031)	0.885(0.085)	0.875(0.013)	**0.894**(0.016)
Wikitalk (Ho et al. 2012)	70	463	0.761(0.005)	0.787(0.084)	0.793(0.061)	0.776(0.034)	**0.842**(0.024)

[1] "N." denotes number of nodes; "L." denotes number of links in the relational data.

Table 2: Decision tree classification results in accuracy (standard deviation)

Data	N.[1]	F.[1]	C.[1]	CART	Mondrian Decision Tree		Ostomachion Decision Tree	
					RJMCMC-2	RJMCMC-3	RJMCMC-2	RJMCMC-3
Seeds (Lichman 2013)	210	7	3	0.818(0.09)	0.695(0.14)	0.813(0.10)	0.749(0.06)	**0.841**(0.04)
Column3c (Lichman 2013)	309	6	2	0.944(0.05)	0.838(0.13)	0.932(0.04)	0.781(0.07)	**0.949**(0.03)
Ecoli (Lichman 2013)	336	7	8	0.704(0.06)	0.579(0.11)	0.665(0.08)	0.571(0.08)	**0.723**(0.08)
Iris (Lichman 2013)	150	4	3	0.869(0.07)	0.709(0.14)	0.844(0.14)	0.632(0.12)	**0.901**(0.07)
Wdbc (Lichman 2013)	567	30	2	0.874(0.04)	0.756(0.11)	0.879(0.07)	0.765(0.07)	**0.890**(0.03)
Wine (Lichman 2013)	178	13	2	0.910(0.06)	0.847(0.11)	0.902(0.07)	0.823(0.07)	**0.921**(0.06)
Banana (Lichman 2013)	5300	2	2	0.704(0.02)	0.586(0.04)	0.720(0.06)	0.591(0.05)	**0.731**(0.04)
Twonorm (Lichman 2013)	7400	20	2	0.926(0.03)	0.829(0.14)	0.943(0.07)	0.823(0.14)	**0.971**(0.04)

[1] "N." denotes number of data; "F." denotes number of feature dimensions; "C." denotes number of classes.

We test our ORM on seven benchmark data sets: *Foodweb, Dolphin, Lazega, Polbooks, Train, Reality, Wikitalk*. We implement the infinite relational model (IRM) (Kemp et al. 2006) and the Mondrian relational model (MRM) (Roy and Teh 2009) as the baseline methods. The link prediction results are reported in Table 1. From the results, we can see that the proposed ORM+RJMCMC-3 achieves the best performance among all the compared methods while keeping competitive small variance at the same time. An interesting observation is that ORM+RJMCMC-2 does not perform well, which may be caused by the high flexibility of the OP. This observation implies that the proposed RJMCMC-3 algorithm is essential for the OP to avoid inferior local maxima due to high flexibility.

Figure 6 visualizes the partition (relational modeling) results of the OP on the seven data sets. The black points denote the observed links and the red lines denote the cuts on the inferred OP partition structures. In general, the convex polygonal components have successfully discovered the underlying asymmetric interactions between the nodes. For example, there are large irregular areas of sparse interactions on *Foodweb, Dolphin, Polbooks, Reality* and there are compact irregular areas of dense interactions on *Dolphin, Lazega, Train, Reality, Wikitalk*.

The Ostomachion Decision Tree

Another interesting application of the OP is decision tree based classification, where the OP partition structure plays the role of decision tree. We refer to this as the Ostomachion decision tree (ODT). The generative process of an ODT is similar to that of an ORM, except that the ODT does not require the generation of indices ξ and η which are actually the 2-D features of the data.

$$\begin{aligned} &\mathcal{O} \sim \text{OP}(5, 10, [0, 1]^2); \quad \phi_k \sim \text{Dir}(1, \cdots, 1); \\ &Y_i \sim \text{Discrete}(\phi_{\hbar(\xi_i, \eta_i)}) \end{aligned} \tag{6}$$

for $i \in \{1, \cdots, n\}$, where (ξ_i, η_i) are the two features of the ith data point. In our experiments, we first use principle component analysis (PCA) to project the data onto a 2-dimensional feature space with the largest eigenvalues. We further normalize the projected data to be in the range of the unit square $[0, 1]^2$.

We test our ODT on eight benchmark data sets: *Seeds, Column3c, Ecoli, Iris, Wdbc, Wine, Banana, Twonorm*, compared to the classical decision tree CART (Quinlan 1986) and the Mondrian decision tree (MDT, by replacing OP in Eq. (6) with MP). For fair comparison, we restrict CART to generate a maximum of 3-level decision tree and it usually results in $6 \sim 7$ leaf components. This number is comparable to the expected number of leaf components in ODT and MDT $\mathbb{E}[N([0, 1]^2)] + 1 = \tau + 1 = 6$.

Table 2 reports the classification results on the eight benchmark data sets. In general, our ODT+RJMCMC-3 outperforms the compared methods in accuracy on all the data sets; the superiority of ODT+RJMCMC-3 is particularly obvious on those small-size data sets. This observation implies that irregular components in the OP are more effective in separating sparse data and describing their geometry. Another observation is that, in both ODT and MDT, RJMCMC-3 outperforms RJMCMC-2 with an average improvement around 0.17 in classification accuracy. This has validated the effectiveness of the proposed cut-translation operation.

Conclusion

In this paper, we propose a stochastic partition process, named the Ostomachion process (OP), which allows for oblique cuts to produce polygonal partitions on the unit square. The two favorable properties of the OP enable ones to easily control the expected number of components and the distribution of areas through a homogeneous Poisson process and a Dirichlet distribution, respectively. Compared to the existing axis-aligned partition processes, the OP is able to capture inter-dimensional dependence and we demonstrate this ability in two applications: relational modeling and decision tree classification. The experimental results show that the OP outperforms the compared methods in both link prediction, by uncovering clear irregular interaction patterns, and decision tree based classification.

References

Aderhold, A.; Husmeier, D.; and Smith, V. A. 2013. Reconstructing ecological networks with hierarchical Bayesian regression and Mondrian processes. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 75–84.

Airoldi, E. M.; Blei, D. M.; Fienberg, S. E.; and Xing, E. P. 2009. Mixed membership stochastic blockmodels. In *Advances in Neural Information Processing Systems (NIPS) 22*, 33–40.

Green, P. J. 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82(4):711–732.

Ho, Q.; Parikh, A. P.; and Xing, E. P. 2012. A multiscale community blockmodel for network exploration. *Journal of the American Statistical Association (JASA)* 107(499):916–934.

Ishiguro, K.; Iwata, T.; Ueda, N.; and Tenenbaum, J. B. 2010. Dynamic infinite relational model for time-varying relational data analysis. In *Advances in Neural Information Processing Systems (NIPS) 23*, 919–927.

Karrer, B., and Newman, M. E. 2011. Stochastic blockmodels and community structure in networks. *Physical Review E* 83(1):016107.

Kemp, C.; Tenenbaum, J. B.; Griffiths, T. L.; Yamada, T.; and Ueda, N. 2006. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence (AAAI)*, volume 3, 381–388.

Lakshminarayanan, B.; Roy, D. M.; and Teh, Y. W. 2014. Mondrian forests: Efficient online random forests. In *Advances in Neural Information Processing Systems (NIPS) 27*, 3140–3148.

Li, B.; Yang, Q.; and Xue, X. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, 617–624.

Lichman, M. 2013. UCI machine learning repository.

Lloyd, J.; Orbanz, P.; Ghahramani, Z.; and Roy, D. M. 2012. Random function priors for exchangeable arrays with applications to graphs and relational data. In *Advances in Neural Information Processing Systems (NIPS) 25*, 1007–1015.

Nakano, M.; Ohishi, Y.; Kameoka, H.; Mukai, R.; and Kashino, K. 2014a. Mondrian hidden markov model for music signal processing. In *2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2405–2409.

Nakano, M.; Ishiguro, K.; Kimura, A.; Yamada, T.; and Ueda, N. 2014b. Rectangular tiling process. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 361–369.

Nowicki, K., and Snijders, T. A. B. 2001. Estimation and prediction for stochastic block structures. *Journal of the American Statistical Association (JASA)* 96(455):1077–1087.

Orbanz, P., and Roy, D. 2015. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(02):437–461.

Porteous, I.; Bart, E.; and Welling, M. 2008. Multi-hdp: A non parametric bayesian model for tensor factorization. In *Proceedings of 23th AAAI Conference on Artificial Intelligence (AAAI)*, volume 8, 1487–1490.

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning* 1(1):81–106.

Roy, D. M., and Teh, Y. W. 2009. The Mondrian process. In *Advances in Neural Information Processing Systems (NIPS) 22*, 1377–1384.

Schmidt, M., and Morup, M. 2013. Nonparametric Bayesian modeling of complex networks: an introduction. *IEEE Signal Processing Magazine* 30(3):110–128.

Sethuraman, J. 1994. A constructive definition of Dirichlet priors. *Statistica Sinica* 4:639–650.

Wang, P.; Laskey, K. B.; Domeniconi, C.; and Jordan, M. I. 2011. Nonparametric Bayesian Co-clustering Ensembles. In *SIAM International Conference on Data Mining (SDM)*, 331–342.

Wang, Y.; Li, B.; Wang, Y.; and Chen, F. 2015. Metadata Dependent Mondrian processes. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 1339–1347.