Nonparametric Relational Models with Superrectangulation

Masahiro Nakano **Ryo Nishikimi** **Yasuhiro Fujiwara**
Akisato Kimura **Takeshi Yamada** **Naonori Ueda**
NTT Communication Science Laboratories, NTT Corporation

Abstract

This paper addresses the question, "What is the smallest object that contains all rectangular partitions with n or fewer blocks?" and shows its application to relational data analysis using a new strategy we call SUPER BAYES as an alternative to Bayesian nonparametric (BNP) methods. Conventionally, standard BNP methods have combined the Aldous-Hoover-Kallenberg representation with *parsimonious* stochastic processes on rectangular partitioning to construct BNP relational models. As a result, conventional methods face the great difficulty of searching for a parsimonious random rectangular partition that fits the observed data well in Bayesian inference. As a way to essentially avoid such a problem, we propose a strategy to combine an extremely *redundant* rectangular partition as a deterministic (non-probabilistic) object. Specifically, we introduce a special kind of rectangular partitioning, which we call *superrectangulation*, that contains all possible rectangular partitions. Delightfully, this strategy completely eliminates the difficult task of searching around for random rectangular partitions, since the superrectangulation is deterministically fixed in inference. Experiments on predictive performance in relational data analysis show that the super Bayesian model provides a more stable analysis than the existing BNP models, which are less likely to be trapped in bad local optima.

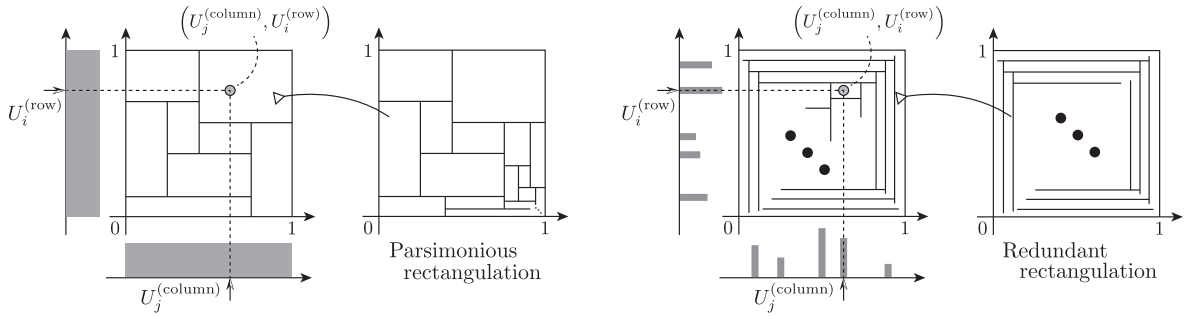
1 INTRODUCTION

Parsimony plays a central role in Bayesian nonparametric (BNP) machine learning. Since BNP models are typically defined as stochastic processes with infinite dimensional parameter spaces (Orbanz and Teh, 2010; Teh, 2010; Teh and Jordan, 2010; Hjort et al., 2010; Orbanz and Roy, 2013), it is not easy to represent them accurately on a computer with only finite resources. However, if we can provide the BNP model[1] with *parsimony*, we can induce it to behave in such a way that it tries to represent itself with as few active dimensions as possible out of the infinite dimensional parameters. Thus, the remaining redundant infinite dimensions of the parameter space can be safely ignored, and as a result, the BNP model can be approximated with high accuracy using a finite number of resources on the computer. This benefit also works for Bayesian inference in the data analysis phase. It is commonly formulated as the problem of finding the value of a parameter and its active dimension in an infinite-dimensional parameter space (or, more precisely, learning their posterior probability), given the observed data. Thanks to *parsimony*, the search for parameters can be typically freed from the vast space of infinite dimensions, and restricted to only a small finite number of active dimensions. The essence of the BNP model is that the active dimension of the parameter space can adapt to the input data; if the observed data desires a richer parameter, it will make it active while saving the necessary but *parsimonious* amount of dimensions. Indeed, the BNP model has taken advantage of *parsimony* in its development, yet we would like to return to this basic principle and consider a completely opposite *redundant* model.

Redundancy has received particular attention in machine learning in recent years. An emblematic example of this is the *lottery ticket hypothesis* (LTH) (Frankle and Carbin, 2019) in deep neural networks (NNs): various forms of LTH claims have been proposed theoretically (Malach et al., 2020; Frankle et al., 2020; Diffenderfer and Kailkhura, 2021), empirically (Raj and Mishra, 2020; Chen et al., 2020, 2021b), and experimentally (Brix et al., 2020; Chen et al., 2021a; Girish et al., 2021), and here we would like to refer specifically to one of the theoretical results. Very roughly speaking

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

[1] BNP models do not always have parsimony; an example of non-parsimonious BNP models would be the Pólya tree (see Corollary 1.5 for details in Orbanz (2011)).



(a) Nonparametric Bayes **(b) Super Bayes**

Figure 1: **Nonparametric Bayes vs. Super Bayes** in terms of generative probabilistic models for relational data. **(a)** The standard Bayesian nonparametric models are modeled in terms of a decomposition into a random rectangular partition of $[0,1] \times [0,1]$ and virtual random coordinates on $[0,1]$ for each row and column, owing to the Aldous-Hoover-Kallenberg representation (Aldous, 1981; Hoover, 1979; Kallenberg, 1989). More specifically, which block the ith row and jth column elements of the input matrix belong to is determined by the following generative model: the virtual coordinate on $[0,1]$ of the ith row is drawn from $U_i^{(\text{row})} \sim \text{Uniform}([0,1])$, the virtual coordinate on $[0,1]$ of the jth column is drawn from $U_j^{(\text{column})} \sim \text{Uniform}([0,1])$, and the rectangular partition sample of $[0,1] \times [0,1]$ is generated from some BNP model, such as the Mondrian process (Roy and Teh, 2009) and the block-breaking process (Nakano et al., 2020). In this case, the cluster to which the ith row and jth column elements of the input matrix belong is determined by the rectangular block to which $(U_j^{(\text{column})}, U_i^{(\text{row})})$ belong. **(b)** The new strategy is to use an extremely redundant and deterministic superrectangulation instead of the parsimonious random rectangular partition. Some readers may be concerned that this extremely redundant superrectangulation may cause overfitting. To counter this, we use a sparse atomic random measure prior on $[0,1]$ (e.g., the Dirichlet process (Ferguson, 1973) with the base measure $\text{Uniform}([0,1])$) for $U_i^{(\text{row})}$ and $U_j^{(\text{column})}$.

(letting us skip all the detailed assumptions since LTH itself is not the purpose of this paper), a sufficiently large random NN is guaranteed to contain some desired NN (called a *winning ticket*) in its subnet (See Theorem 2.1 in Malach et al. (2020) for a more precise statement). As a result, LTH strongly encourages the new paradigm in NN learning. While it is standard practice to iteratively update the parameters of an NN in order to learn a desired NN, LTH suggests that after creating a sufficiently redundant random NN, simply pruning the network can be an alternative to learning. This is a striking example of the usefulness of model *redundancy* in a nutshell. Inspired by LTH, the genesis of this research lies in the idea that, even in the BNP model, by daring to remove *parsimony* and using *redundancy*, we could avoid the iterative search and update of parameters in the training of the BNP model, and replace the learning alternative with only pruning-like operations.

According to the above considerations, this paper proposes a new Bayesian machine learning strategy called Super Bayes (SB), which removes *parsimony* in the BNP model and analyzes the data using a model with *redundancy*. As an application that best illustrates the difference between BNP and SB, this paper will focus in particular on the problem of rectangular partitioned clustering (RPC) of relational data. The goal of RPC is to find a partition and an order of rows and columns of the input matrix such that all blocks are rectangular, so that the elements in each block are as homogeneous as possible, given matrix-type relational data (Figure 2). Here, to clarify the motivation behind the genesis of SB, we would like to discuss the serious affliction of the BNP methods and the key insight of the SB method.

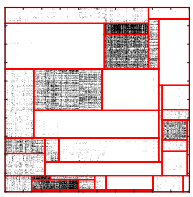


Figure 2: RPC

Serious affliction of BNP methods - The standard strategy of generic BNP methods for relational data analysis is to use the Aldous-Hoover-Kallenberg representation theorem (Aldous, 1981; Hoover, 1979; Kallenberg, 1989) for the construction of BNP relational models (Roy and Teh, 2009; Roy, 2011; Choi and Wolfe, 2014; Rodriguez and Ghosh, 2009; Shan and Banerjee, 2008; Miller et al., 2009; Ishiguro et al., 2016; Caldas and Kaski, 2008; Airoldi et al., 2013; Fan et al., 2018a, 2016; Lloyd et al., 2012; Lovász, 2009; Ge et al., 2019; Fan et al., 2018b, 2020). Figure 1 (a) illustrates the case of using random rectangular partitioning specifically as a BNP relational model, which consists of a random rectangular partition on $[0,1] \times [0,1]$ and uniform random variables corresponding to the coordinates of each row and column on $[0,1]$. As a result, Bayesian inference is typically based on an iterative algorithm that repeats two steps: (1) update

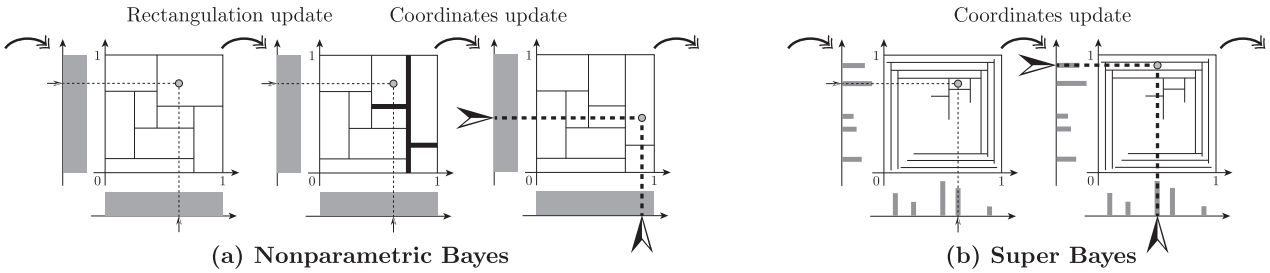


Figure 3: **Nonparametric Bayes vs. Super Bayes** for Bayesian inference (e.g., Markov chain Monte Carlo). **(a)** The BNP model requires alternating updates of both the rectangular partition and the coordinates of each row and each column of the input matrix. As a result, if the chain is trapped in a bad rectangular partition, it will lead to bad coordinates induced by it, and these bad coordinates will induce even worse rectangular partitions, often making it easy to get trapped in bad local optima. **(b)** The SB model, on the other hand, only needs to update the coordinates of each row and column sequentially, while keeping the rectangular partition fixed.

the rectangular partition on $[0,1] \times [0,1]$, and (2) update the coordinates of each row and column. See also Figure 3 (a). *Parsimony* imposed by the BNP model often has a particularly negative impact on the former. Specifically, the former part of the inference algorithm is continually suppressed in a conservative way as it searches for what partitions are appropriate for the data. Consequently, it often gets trapped in bad local optima. The LTH way of explaining this is that the BNP model should always have a winning ticket (because it must have positive probability for every possible lottery), but it is too difficult to find out where it is because the winning ticket is buried in an infinite number of losing tickets.

Key insight of SB method - The crux of the SB method is to eliminate the need to look for the winning ticket. In other words, we want all lotteries to be visible as being there. To be more specific, we consider the following object, which we call *superrectangulation*. It is a rectangular partition, one that contains all rectangular partitions with n or fewer blocks by cutting out parts of itself. If the superrectangulation is used instead of the random rectangular partition for the relational model, as shown in Figure 1 (b), we can greatly simplify Bayesian inference. It means that (1) updating the rectangular partitioning is no longer necessary in Bayesian inference, and training can be completed only by (2) updating the coordinates of each row and column of the input matrix, as shown in Figure 3 (b). Determining the coordinates of each row and column means that the rectangular partitioning needed to describe the data can be achieved at the same time by cutting a part of it out of the fixed superrectangulation. Our contributions can be summarized as follows.

(1) Super Bayes framework - The first contribution of this paper is the SB framework itself, as an alternative to the conventional BNP methods. Inspired by the recent LTH for NNs, this is a method for analyzing data using extremely redundant deterministic object in Bayesian statistics, as Figures 1 and 3 illustrate all of its key insights. We will deal with relational data analysis based on rectangular partitioning as an example of the remarkable effect of the SB framework.

(2) Superrectangulation - Our second contribution is to introduce the notion of the superrectangulation as a universal object that contains any rectangular partition as its part. To the best of our knowledge, this is the first time that the superrectangulation has been introduced in machine learning and related fields. Aiming at the ultimate goal of constructing a superrectangulation useful for machine learning, we collect in Section 2 the great wisdom scattered in various fields that we expect to be useful for its realization. In Section 3, we give the definition of a superrectangulation and discuss some observations and results from the perspective of combinatorics. In particular, Proposition 3.2 gives the (potentially) constructive existence of a superrectangulation, although it is somewhat less practical. Therefore, we further explore more practically useful constructions of a superrectangulation with two strategies. The first strategy continues to be based on the findings of combinatorics; specifically, we describe the intrinsic difficulty of its construction in Remark 3.4, and propose an object that we hope may overcome this difficulty, the *zigzag rectangulation*, in Conjecture 3.5. The second strategy is more daring, inspired by LTH, and hypothesizes that *in a random partition with an extremely huge number of blocks, any small rectangular partition may appear with high probability.* In Section 4, we investigate how useful these strategies are as relational data analysis.

2 PRELIMINARIES

For the discussion of superrectangulation, it is very useful to survey the existing wisdom on closely related research topics scattered in various research fields.

2.1 Permutation

Word and permutation - A *word* is a sequence of *letters* selected from a certain *alphabet*. For example, the word $w = 3123$ is constructed as a sequence of letters 3, 1, 2, and 3 chosen from the alphabet $\mathbb{N}$. Throughout this paper, we will use the set of natural numbers (i.e., $\mathbb{N} = \{1, 2, \dots\}$) or subsets (e.g., $[n] := \{1, 2, \dots, n\}$ and $[n+1]$) of it as the alphabet. The *length* of the word w, denoted $|w|$, is its number of letters. For example, the word $w = 3123$ has $|w| = 4$. If a word w has at least length i, then we denote by $w(i)$ its ith letter. For $w = 3123$, we have $w(1) = 3$ and $w(3) = 2$. As a benefit of having the alphabet as natural numbers, we can immediately introduce the notion of *order-isomorphism* for words from the total order in the natural numbers. Specifically, two words $w, w' \in \mathbb{N}^n$ of length n are *order-isomorphic* if, for all indices $i, j \in [n]$, we have

$$w(i) > w(j) \iff w'(i) > w'(j). \tag{1}$$

Note that this also implies that $w(i) = w(j) \iff w'(i) = w'(j)$. For example, $w = 3123$ and $w' = 7257$ are order-isomorphic. Needless to say, $w = 3123$ and $w' = 7256$ are not order-isomorphic. A *permutation* of length n is a word consisting of letters chosen from the alphabet $[n]$, each occurring precisely once. For example, $w = 4123$ is a permutation. Needless to say, $w = 3123$ and $w = 5234$ are words but not permutations. Since permutations are also words, we can consider order-isomorphism between permutations according to Equation (1).

Geometric representation - The geometric interpretation of words and permutations is often very useful in revealing their properties. For a word or permutation w, if we arrange the indices i vertically in the order $1, 2, \dots,$ and horizontally in the order $w(1), w(2), \dots,$ the word w (e.g., $w = 45132$) can be represented as in Figure 4.

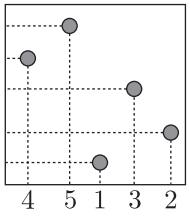


Figure 4: 45132

2.2 Universal permutation problem

Superpermutation - A historically important question surrounding permutations is "What is the smallest object (typically a word of minimal length) that can *contain* all permutations of length n?" The same kind of problem has been defined and proposed in various ways, and they are collectively called the *universal permutation problem* (UPP) (Engen and Vatter, 2021). For a more precise description of UPP, we must clarify the notion of *containment*, i.e., how words contain permutations. Conventionally, two different interpretations of the containment have been considered:

- **Factor** - The word w contains the permutation π as a *factor*, which means that w is represented in the form $w = u\sigma v$, and σ is order-isomorphic to π. For example, the word $w = 23724515$ has a factor $\pi = 4123$ since w can be expressed as $w = u\sigma v$ where $u = 23$, $\sigma = 7245$, and $v = 15$, and $\sigma = 7245$ is order-isomorphic to $\pi = 4123$.

- **Subsequence** - The word w contains the permutation π as a *subsequence*, which means that there are indices $i_1, i_2, \dots, i_n$ so that the word $\sigma = w(i_1)w(i_2)\dots w(i_n)$ is order-isomorphic to π. For example, the word $w = 23724518$ has a Subsequence $\pi = 12345$ since there are $i_1 = 1$, $i_2 = 2$, $i_3 = 5$, $i_4 = 6$, and $i_5 = 8$, and the word $\sigma = w(i_1)w(i_2)w(i_3)w(i_4)w(i_5) = 23458$ is order-isomorphic to $\pi = 12345$.

While these two different notions of containment give rise to essentially different UPPs, another element known to have a significant impact on UPP is the choice of alphabet. The most elementary problem is the setting where the alphabet size is taken to be equal to the length n of the permutation. In this case, UPP becomes a problem of finding a word $w \in [n]^m$ of length m that contains all permutations of length n as a factor or subsequence, and whose length m is as small as possible. On the other hand, a more flexible setting could be to use $\mathbb{N}$ for the alphabet. And interestingly, the case where $[n+1]$ is used for the alphabet is also well studied, as a form of in-between these two extreme cases, $[n]$ and $\mathbb{N}$. In summary, UPP has been studied in a total of six different ways, two for the interpretation of containment and three for the choice of alphabet. We will simply call the object that contains all permutations of length n a *superpermutation*. The upper bounds on the superpermutation size currently known (Johnson, 2009; Gao et al., 2019; Radomirovic, 2012; Miller, 2009) are summarized in Table 1 (Engen and Vatter, 2021).

An excellent recent survey paper for all cases can be found in Engen and Vatter (2021). In this paper we will only mention two cases in particular, "factor $\times$ alphabet $[n+1]$" and "subsequence $\times$ alphabet $[n+1]$". For the former, the existence of a superpermutation called the *universal cycle* is shown constructively[2]:

Theorem 2.1. *(See Theorem 1 in Johnson (2009)) There is a word over the alphabet $[n+1]$ of length*

[2]The specific construction of universal cycles is complicated and will not be discussed here. If you are interested, please refer to Section 2 in (Johnson, 2009).

Table 1: Current best upper bounds for superpermutation length in six types of UPPs.

Containment	Alphabet		
	Words over $[n]$	Words over $[n+1]$	Words over $\mathbb{N}$
Factor	$n-3+\sum_{i=0}^{3}(n-i)!$	$n!+n-1$	$n!+n-1$
Subsequence	$\lceil n^2-\frac{7}{3}n+\frac{19}{3}\rceil$	$\frac{n^2+n}{2}$	$\lceil\frac{n^2+1}{2}\rceil$

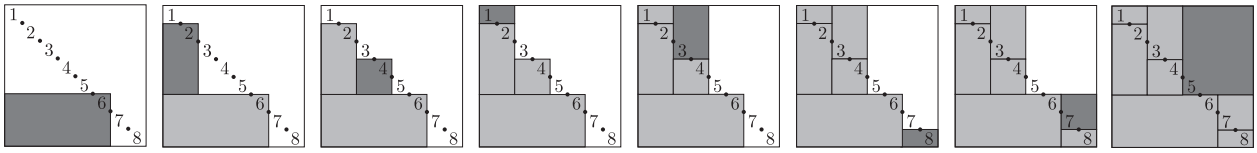


Figure 5: Map from permutations (e.g., 62413875) to diagonal rectangulations (Law and Reading, 2012). We first draw $n+1=9$ distinct *diagonal points* on the diagonal, with one of the points being the top-left corner and another being the bottom-right corner. Let T be the union of the rectangles drawn in the first $i-1$ steps. To draw the ith rectangle, we consider the label i on the diagonal. If the diagonal point p on the diagonal immediately above or left of the label i is not in T, then the upper left corner of the new rectangle is the rightmost point of T immediately to the left of p. If the diagonal point p immediately below or right of the label i is not in T, then the lower right corner of the new rectangle is the highest point of T immediately below p. If p is in T, then the lower-right corner of the new rectangle is the rightmost point of T immediately to the right of p.

$n!+n-1$ *containing factors order-isomorphic to every permutation of length* n.

For the latter, the existence of a superpermutation called the *zigzag word* is shown constructively:

Theorem 2.2. *(See Theorem* 3.1 *in Miller (2009) and Theorem* 9 *in Engen and Vatter (2021)) For all* $n \geq 1$*, there is a word over the alphabet* $[n+1]$ *of length* $(n^2+n)/2$ *containing subsequences order-isomorphic to every permutation of length* n*.*

Delightfully, the proof of this theorem is constructive, and the word mentioned (i.e., the superpermutation in this UPP setting) can be obtained concretely as follows. We first introduce the *infinite zigzag word* (IZG), which is defined as a word[3] formed by alternating the ascending *runs* of the odd natural numbers $1357\ldots$ and the descending *runs* of the even natural numbers $\ldots 8642$. Specifically, IZG has the form of

$$\underbrace{1357\ldots}_{\text{1st run}}\ \underbrace{\ldots 8642}_{\text{2nd run}}\underbrace{1357\ldots}_{\text{3rd run}}\ \underbrace{\ldots 8642}_{\text{4th run}}\underbrace{1357\ldots}_{\text{5th run}}\ \ldots. \quad (2)$$

Surprisingly, a slight modification of IZG provides a specific configuration of the superpermutation described in Theorem 2.2:

Corollary 2.3. ***(Superpermutation)*** *The restriction of the first* n *runs of IZG to the alphabet* $[n+1]$ *contains all permutations with length* n *as subsequences, whose length is* $(n^2+n)/2$*.*

[3]Strictly speaking, this does not meet the definition of a word, but for the sake of simplicity, we will relax the definition a bit and consider it a word in the broad sense.

For example, when $n=8$, the superpermutation w is

$$\underbrace{13579}_{\text{1st run}}\ \underbrace{8642}_{\text{2nd run}}\ \underbrace{13579}_{\text{3rd run}}\ \underbrace{8642}_{\text{4th run}}\ \underbrace{13579}_{\text{5th run}}\ \underbrace{8642}_{\text{6th run}}\ \underbrace{13579}_{\text{7th run}}\ \underbrace{8642}_{\text{8th run}}.$$

2.3 Rectangulation

Rectangulation - A rectangular partition, also called rectangulation, is a partition of a rectangle (or a matrix) in which all blocks are disjoint rectangles. This is a research subject that has received a great deal of attention in recent years, especially as an intersection of multiple research areas in combinatorics (Reading, 2012; Merino and Mütze, 2021; Hong et al., 2000; Mackisack and Miles, 1996; Ackerman et al., 2006), machine learning (Kemp et al., 2006; Roy and Teh, 2009; Roy, 2011; Orbanz and Roy, 2013; Nakano et al., 2014), and probability and statistics (Maazoun, 2019; Borga and Maazoun, 2020; Merino and Mütze, 2021).

Relationship between rectangulation and permutation - In recent years, it has become clear that there is a close relationship between permutations and rectangular partitions, and in particular, the one-to-one correspondence between special classes of both has been studied in depth. For a bird's eye survey of recent developments, see for example (Merino and Mütze, 2021). Here we would particularly like to mention two closely related facts: the existence of the surjective map from permutations to *diagonal rectangulations* and the surjective map from permutations to *generic rectangulations*. A rectangulation is a *diagonal rectangulations* if it has a representative in which each rectangle's interior intersects the diagonal.

Proposition 2.4. *(See Proposition* 6.2 *in Law and*

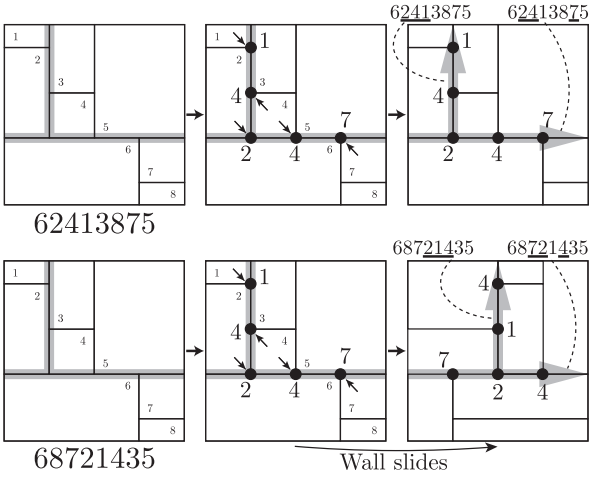


Figure 6: Map from permutations, e.g., 62413875 (top) and 68721435 (bottom), to generic rectangulations. The first step is to convert from permutation to diagonal rectangulation (left). Then we assign to the vertex on the *walls* (colored gray) the label of the block that contains that vertex as its own upper left or lower right corner (middle). Finally, the order of the vertices on the wall will be rearranged according to the permutation. Specifically, vertices on the horizontal wall are aligned in permutation order from left to right, and vertices on the vertical wall are aligned in permutation order from bottom to top (right).

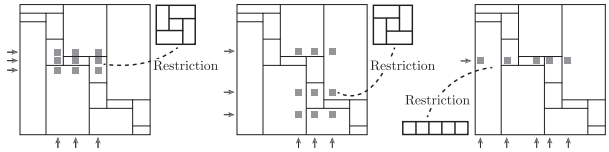


Figure 7: Restriction operation (with three examples).

Reading (2012)) There is a surjective map from permutations to diagonal rectangulations.

Figure 5 shows how to transform permutations into diagonal rectangulations, which is specifically given in Law and Reading (2012). Similarly, for *generic rectangulations*, that is, rectangular partitioning without any restrictions, the result below is also known.

Proposition 2.5. *(See Proposition* 4.2 *in Reading (2012)) There is a surjective map from permutations to generic rectangulations.*

Figure 6 shows how to transform permutations into generic rectangulations, provided in Reading (2012).

3 SUPERRECTANGULATION

This section will start with the definition of the superrectangulation and then continue with the key observations and results. All the details of the proofs will be given in Appendix B. As an intuitive definition sketch,

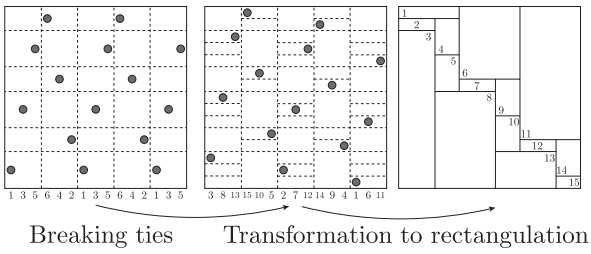


Figure 8: Illustration of *zigzag rectangulation*. **Left**: Zigzag word ($n = 5$). **Middle**: Corresponding permutation to the zigzag word obtained by ranking the ties that appear in the zigzag word. **Right**: *Zigzag rectangulation* obtained by applying the mapping from permutations to generic rectangulations described in Section 2.3 to the zigzag word. Interestingly, the resulting rectangulation is a diagonal rectangulation.

the superrectangulation can be viewed as a rectangular partition that *contains* all rectangular partitions with n blocks. However, this expression is still vague; by analogy with the factors and subsequences in the superpermutation (Section 2.1), the superrectangulation also requires the notion of *containment* to be realized first. We introduce the following *restriction* operation on rectangular partitions (See also intuitive illustrations in Figure 7):

Restriction - We consider a rectangular partition R on $[0,1] \times [0,1]$. Given a matrix $\boldsymbol{Z} = (Z)_{L\times M}$ and the $[0,1]$ variables $U_l^{(\mathrm{row})}$ $(l = 1, \ldots, L)$ and $U_m^{(\mathrm{column})}$ $(m = 1, \ldots, M)$ corresponding to each of its rows and columns, we introduce the following *restriction* on the rectangular partition R. Each element $Z_{l,m}$ of the matrix $\boldsymbol{Z}$ is assigned to the block to which the coordinates $(U_m^{(\mathrm{column})}, U_l^{(\mathrm{row})})$ point in the rectangular partition R on $[0,1]\times[0,1]$, and we obtain the rectangular partition of the matrix $\boldsymbol{Z}$. We will call the resulting rectangular partition the *restriction* of the original partition R.

Definition 3.1. ***(Superrectangulation)*** *A rectangular partition R on $[0,1] \times [0,1]$ is said to be a superectangulation if, for any rectangular partition R' of matrix $\boldsymbol{Z} = (Z)_{L\times M}$ with n blocks, there exists $U_l^{(\mathrm{row})}$ $(l = 1, \ldots, L)$ and $U_m^{(\mathrm{column})}$ $(m = 1, \ldots, M)$ and its corresponding restriction of R coincides with the specified rectangular partition R' of $\boldsymbol{Z}$.*

Our first observation is the existence of a superrectangulation with a huge number of blocks that (allowing for duplication) counts all rectangular partitions:

Proposition 3.2. ***(First naive observation)*** *There is a superrectangulation that has $n \cdot n!$ blocks.*

This result follows immediately from the fact that there is the surjective mapping from permutations to generic rectangulations (Section 2.3). That is, there are only $n!$ permutations of length n, and the rectangu-

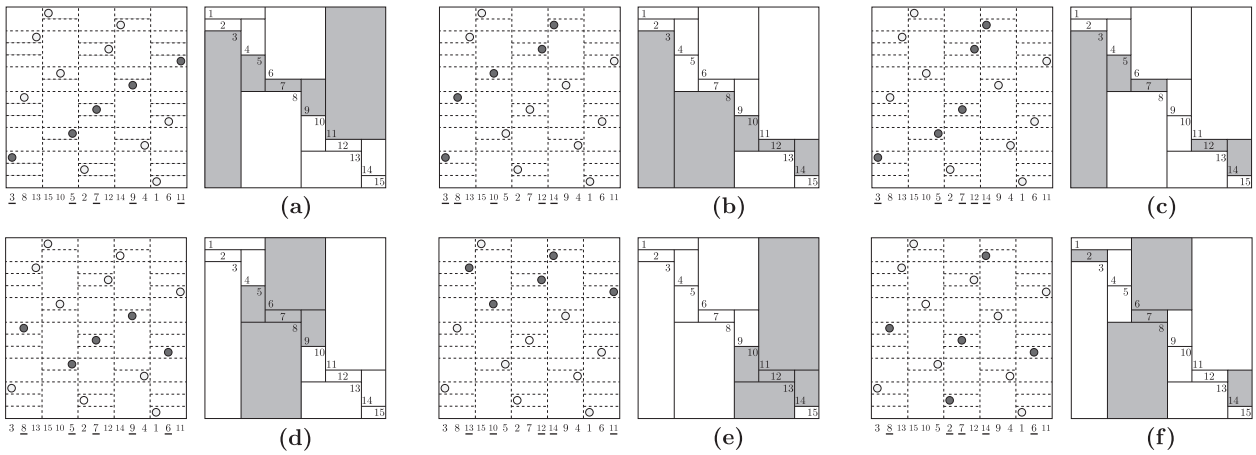


Figure 9: **(a, b, c)** Subsequences that are order-isomorphic to 12345 extracted from the zigzag word ($n = 5$), and the corresponding blocks in the zigzag rectangulation (right). For (a) and (b), a rectangular partition consisting of only the selected blocks (as shown in the example on the right side of Figure 7, a rectangular partition with five blocks lined up horizontally) can be constructed by restriction operation, but for (c), a rectangular partition consisting of the selected five blocks cannot be created. **(d, e, f)** The same analysis is performed for subsequences that are order-isomorphic to the permutation 41352. For (d) and (e), the rectangular partition on the left side of Figure 7 can be created from the selected blocks by restriction, but for (f), this is not possible.

lations obtained by transforming these permutations cover all rectangulations with n blocks. Needless to say, this example would not be very useful in practice because the number of blocks required as a superrectangulation is too large. Therefore, we need a strategy to reduce the size of a superrectangulation more dramatically. The subsequent development of this paper is to find a superrectangulation of smaller size.

Now that we have confirmed the existence of superrectangulation, how can we dramatically reduce its size (the number of blocks)? What comes to mind here is that instead of representing the enumeration of rectangular partitions with permutations of length n, we can represent them with a single superpermutation (Section 2.2). That is, the set of all permutations is compressed by a superpermutation. Recalling here the sizes of the two superpermutations introduced in Section 2.2, the length of a universal cycle is $n!+n-1$ and the length of a zigzag word is $(n^2+n)/2$. It is worth noting that the zigzag word can be represented by a dramatically shorter length. Thus, the strategy that we come up with is to convert the zigzag word into a rectangular partition, which may be used as a superrectangulation, as shown in Figure 8. We will call the rectangulation generated from the zigzag word a *zigzag rectangulation*. Since the zigzag partition has $(n^2+n)/2$ blocks, this is a partition of dramatically smaller size than the superrectangulation described in Proposition 3.2. Then, does zigzag partitioning really satisfy the requirements of superrectangulation?

Remark 3.3. ***(Second observation)*** *Unfortunately, the zigzag rectangulation does not satisfy the requirement of the superrectangulation because there is a rectangular partition with n blocks that cannot be generated by the restriction operation. For example, when $n = 5$, the zigzag rectangulation does not contain the rectangulation of 5 blocks arranged vertically (perpendicularly). However, delightfully, it is also easy to confirm that the zigzag rectangulation contains most of the 116 possible generic rectangulations with 5 blocks.*

We will continue to discuss this observation in more detail. From this observation, we can expect that, with very few exceptions, the zigzag word will contain most of the rectangulations with n blocks. Some readers may think that this positive observation can be immediately extended to all $n \in \mathbb{N}$. We also believed that at first, however, as we tried to prove it, we have gradually come to realize that it is not an easy task. We were initially optimistic that since the zigzag word contains all short permutations as subsequences (Theorem 2.2), the zigzag rectangulation corresponding to the zigzag word must also contain all the smaller rectangular partitions. However, unfortunately, this is not true. The reason for this is that the subsequence extraction from the zigzag word does not necessarily correspond to the restriction operation in the zigzag rectangulation (See also Figure 9). More precisely, it can be explained as follows:

Remark 3.4. ***(Third observation)*** *The relationship between the following statements is considered:* $(S1)$ *The permutation w is contained in the zigzag word as a subsequence.* $(S2)$ *The rectangular partition R corresponding to the permutation w is contained as a restriction in the zigzag rectangulation. At this time,* $(S1) \Leftarrow (S2)$ *is valid, but* $(S1) \Rightarrow (S2)$ *is not.*

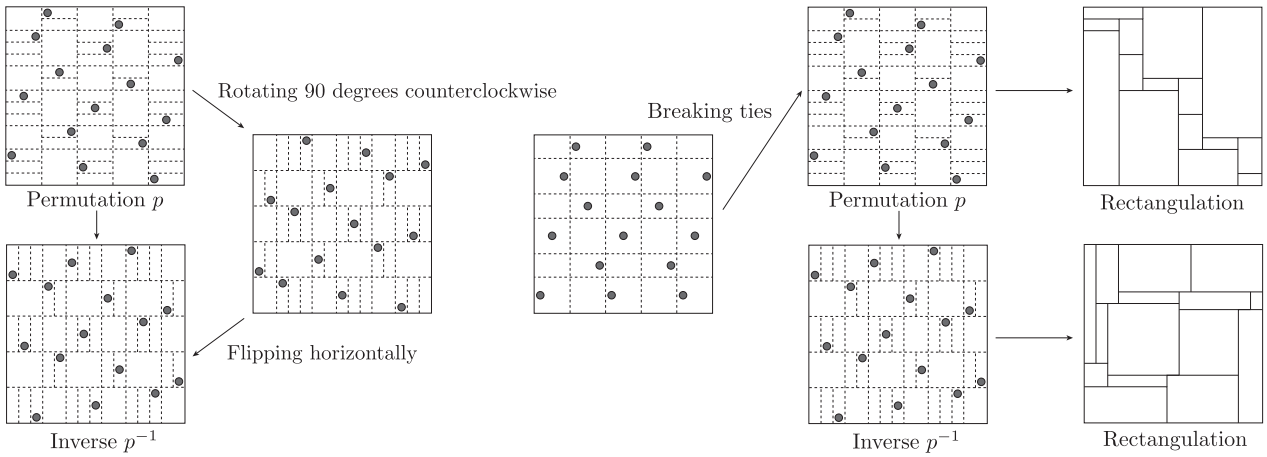


Figure 10: **Left**: Geometric interpretation of inverse of zigzag word. We apply the breaking ties operation to the zigzag word to obtain the corresponding permutation p, and then take the inverse p^{-1}. **Right**: Rectangulations corresponding to the zigzag word and its inverse obtained by applying the mapping from permutations to generic rectangulations described in Section 2.3.

This observation clarifies the difference between the superpermutation and the superrectangulation. Additionally, it suggests another possibility as a means of generating a rectangulation (i.e., another candidate of a superrectangulation) from the zigzag word. Once again, we will carefully inspect Figure 9. We focus on the cases (Figure 9 (a), (b), (d), (e)) where the subsequences in the zigzag word correspond to the rectangulations in the zigzag rectangulation. In these cases, we see that the subsequences in the zigzag word are concentrated in narrow vertical regions in the geometric representation of the zigzag word. Conversely, vertically spread subsequences in the zigzag word are less likely to correspond to rectangulations in the zigzag rectangulation. Therefore, a possible way to rescue the vertically spread subsequences of the zigzag word is to rotate the geometric representation of the zigzag words by 90 degrees, so that they can be pushed into narrow vertical regions, as shown in Figure 10. We will call the division that can be made from the inverse of the zigzag word a *zigzag$^{\mathrm{i}}$ rectangulation*.

Based on the above considerations, we expect that zigzag partitioning and zigzag$^{\mathrm{i}}$ rectangulation may include all rectangular partitions with n blocks. However, at the same time, we are finding out that the proof does not seem to be easy. Therefore, we wish to share this with the community as an open problem:

Conjecture 3.5. *The zigzag rectangulation and the zigzag$^{\mathrm{i}}$ rectangulation contains all rectangular partitions with n blocks for all $n \in \mathbb{N}$.*

So far, we have discussed the superectangulation in terms of combinatorics, satisfying strict requirements. On the other hand, we would like to share some more machine learning flavored ideas at the end. How about adding randomness to the superectangulation? This idea immediately leads to another interesting subject:

Hypothesis 3.6. *A random rectangulation with an extremely large number of blocks, generated from a uniform distribution of generic rectangulations, contains every rectangular partition with some small number of blocks with high probability.*

This is exactly the hypothesis that comes from the LTH analogy as discussed in Section 1. The reasons for our strong endorsements of this conjecture and hypothesis will be explained in a little more depth in the supplementary material. We believe that these topics can be a new research direction in Bayesian modeling. Now, in the next section, we will actually demonstrate data analysis using the zigzag/zigzag$^{\mathrm{i}}$ rectangulation (abbreviated as Zigzag) and the random rectangulation (Random) for the SB relational model.

4 SUPER BAYESIAN ANALYSIS

Super Bayesian relational model - The overall picture of the SB relational model is as described in Section 1 and Figure 1. We suppose that the input observation relational data $\boldsymbol{Z} = (Z_{i,j})_{N \times M}$ consists of categorical elements, i.e., $Z_{i,j} \in \{1, 2, \dots, D\}$ $(D \in \mathbb{N})$. The generative probabilistic model of the SB rectangular partitioning is as described in Figure 1 (b). Each block (indexed by $k = 1, 2, \dots$) has a latent Dirichlet random variable $\vartheta_k \sim \mathrm{Dirichlet}(\boldsymbol{\alpha}_0)$ $(k = 1, 2, \dots)$, where $\boldsymbol{\alpha}_0$ is a D-dimensional non-negative hyper parameter. For the superrectangulations, as discussed in Section 3, the number of blocks corresponding to the representational power needs to be set in advance, so we shall generate it from a Poisson distribution with

Table 2: Perplexity comparison for real-world relational data analysis (mean±std)

	Nonparametric Bayes			Super Bayes	
	MP	BBP	PCRP	Zigzag	Random
Wiki	1.2838 ± 0.0094	1.2712 ± 0.0056	1.2583 ± 0.0041	$\mathbf{1.2565} \pm 0.0017$	1.2648 ± 0.0082
Facebook	1.1944 ± 0.0217	1.1818 ± 0.0197	1.1545 ± 0.0187	$\mathbf{1.1493} \pm 0.0095$	1.1682 ± 0.0234
Twitter	1.2316 ± 0.0209	1.2146 ± 0.0058	$\mathbf{1.2057} \pm 0.0092$	1.2077 ± 0.0071	1.2102 ± 0.0087
Epinions	1.4098 ± 0.0064	1.4006 ± 0.0044	1.3955 ± 0.0061	$\mathbf{1.3951} \pm 0.0054$	1.3979 ± 0.0056

a sufficiently large mean as a simple strategy. As for the zigzag/zigzag$^{\mathrm{i}}$ rectangulation, we chose one or the other with uniform probability for each trial. The details of the model are specified in Appendix A.1.

Datasets - We employ the standard benchmark datasets for evaluation Leskovec et al. (2010): **Wiki** (dat, a), **Facebook** (dat, b), **Twitter** (dat, c), and **Epinions** (dat, d). We selected the top 1000 active nodes based on their interactions with others; subsequently we randomly sampled 500×500 matrix to construct the relational data, as in (Fan et al., 2020; Nakano et al., 2020). We held out 20% cells of the input data for testing, and each model was trained using the remaining 80% of the cells. We evaluated the models using perplexity as a criterion: $\mathrm{perp}(\hat{Z}) = \exp(-(\log\, p(\hat{Z}))/E)$, where E is the number of non-missing cells in the partitioned matrix $\hat{Z}$. We compare the SB method with the existing BNP models for rectangular partitioning, the Mondrian process (MP) (Roy and Teh, 2009), the block-breaking process (BBP) (Nakano et al., 2020), and the permuton-induced Chinese restaurant process (PCRP) (Nakano et al., 2021). The details of experimental settings and inference algorithms are provided in Appendix A.2.

Experimental results - We ran 10 trials of analysis for each method on each data set. Table 3 summarizes the test perplexity comparison results. In terms of average prediction performance, it can be confirmed that the SB methods show equal or slightly better performance than the BNP methods. Furthermore, as the standard deviation of the prediction performance shows, the SB methods have less variation in the analysis results in multiple trials, and it can be confirmed that it can reduce the influence of local optima in Bayesian inference. This can be attributed to the fact that the BNP model iteratively updates two elements, the row and column coordinates and the rectangular partition, while the SB model completely eliminates the update of the rectangular partition. Furthermore, interestingly, when comparing the modified zigzag rectangulation and random rectangulation for the SB methods, they show comparable prediction performance. This may experimentally suggest that, like LTH in NNs, a sufficiently large random rectangulation serves as a pseudo-superrectangulation.

5 CONCLUSION

This paper has proposed the super Bayesian strategy for learning with highly redundant universal objects in Bayesian methods, inspired by the lottery ticket hypothesis in deep neural networks. As a concrete example of the super Bayesian data analysis, this paper focused on relational data analysis, and in the process has proposed an interesting research topic: superrectangulation, which is a rectangular partition that contains every small rectangular partition. Aiming at constructing the superrectangulation, we have taken two approaches, one from a combinatorial perspective and the other from a statistical machine learning perspective. We expect that super Bayesian strategy has the potential to become a new framework that can be twinned with Bayesian nonparametric methods with models on infinite-dimensional parameter spaces.

References

`http://snap.stanford.edu/data/wiki-Vote.html`.

`http://snap.stanford.edu/data/ego-Facebook.html`.

`http://snap.stanford.edu/data/ego-Twitter.html`.

`http://snap.stanford.edu/data/soc-Epinions1.html`.

Ackerman, E., Barequet, G., and Pinter, R. Y. (2006). A bijection between permutations and floorplans, and its applications. *Discrete Applied Mathematics*, 154(12):1674–1684.

Airoldi, E. M., Costa, T. B., and Chan, S. H. (2013). Stochastic block model approximation of a graphon: Theory andconsistent estimation. In *Advances in Neural Information Processing Systems*.

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

Borga, J. and Maazoun, M. (2020). Scaling and local limits of baxter permutations through coalescent-walk processes. In *International Conference on*

Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms, pages 7:1–7:18.

Brix, C., Bahar, P., and Ney, H. (2020). Successfully applying the stabilized lottery ticket hypothesis to the transformer architecture. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3909–3915.

Caldas, J. and Kaski, S. (2008). Bayesian biclustering with the plaid model. In *2008 IEEE Workshop on Machine Learning for Signal Processing*, pages 291–296.

Chen, T., Frankle, J., Chang, S., Liu, S., Zhang, Y., Carbin, M., and Wang, Z. (2021a). The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 16306–16316.

Chen, T., Frankle, J., Chang, S., Liu, S., Zhang, Y., Wang, Z., and Carbin, M. (2020). The lottery ticket hypothesis for pre-trained BERT networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*.

Chen, T., Sui, Y., Chen, X., Zhang, A., and Wang, Z. (2021b). A unified lottery ticket hypothesis for graph neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, pages 1695–1706.

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

Diffenderfer, J. and Kailkhura, B. (2021). Multi-prize lottery ticket hypothesis: Finding accurate binary neural networks by pruning A randomly weighted network. In *International Conference on Learning Representations*.

Engen, M. and Vatter, V. (2021). Containing all permutations. *Am. Math. Mon.*, 128(1):4–24.

Fan, X., Li, B., and Sisson, S. (2018a). Rectangular bounding process. In *Advances in Neural Information Processing Systems*, pages 7631–7641.

Fan, X., Li, B., and Sisson, S. A. (2018b). The binary space partitioning-tree process. In *International Conference on Artificial Intelligence and Statistics*, pages 1859–1867.

Fan, X., Li, B., and Sisson, S. A. (2020). Online binary space partitioning forests. In *The 23rd International Conference on Artificial Intelligence and Statistics*, pages 527–537.

Fan, X., Li, B., Wang, Y., Wang, Y., and Chen, F. (2016). The Ostomachion Process. In *AAAI Conference on Artificial Intelligence*, pages 1547–1553.

Ferguson, T. (1973). Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 2(1):209–230.

Frankle, J. and Carbin, M. (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Frankle, J., Dziugaite, G. K., Roy, D. M., and Carbin, M. (2020). Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269.

Gao, A. L. L., Kitaev, S., Steiner, W., and Zhang, P. B. (2019). On a greedy algorithm to construct universal cycles for permutations. *Int. J. Found. Comput. Sci.*, 30(1):61–72.

Ge, S., Wang, S., Teh, Y. W., Wang, L., and Elliott, L. (2019). Random tessellation forests. In *Advances in Neural Information Processing Systems 32*, pages 9575–9585.

Girish, S., Maiya, S. R., Gupta, K., Chen, H., Davis, L. S., and Shrivastava, A. (2021). The lottery ticket hypothesis for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–771.

Hjort, N., Holmes, C., Mueller, P., and Walker, S. (2010). *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, Cambridge, UK.

Hong, X., Huang, G., Cai, Y., Gu, J., Dong, S., Cheng, C., and Gu, J. (2000). Corner block list: an effective and efficient topological representation of non-slicing floorplan. In *IEEE/ACM International Conference on Computer-Aided Design*.

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

Ishiguro, K., Sato, I., Nakano, M., Kimura, A., and Ueda, N. (2016). Infinite plaid models for infinite bi-clustering. In *AAAI Conference on Artificial Intelligence*, pages 1701–1708.

Johnson, J. R. (2009). Universal cycles for permutations. *Discret. Math.*, 309(17):5264–5270.

Kallenberg, O. (1989). On the representation theorem for exchangeable arrays. *Journal of Multivariate Analysis*, 30(1):137–154.

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

Law, S. and Reading, N. (2012). The hopf algebra of diagonal rectangulations. *J. Comb. Theory, Ser. A*, 119(3):788–824.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). Predicting positive and negative links in online social networks. In *International Conference on World Wide Web*, pages 641–650.

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*.

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

Maazoun, M. (2019). On the brownian separable permuton. *Combinatorics, Probability and Computing*, 29(2):241–266.

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

Malach, E., Yehudai, G., Shalev-Shwartz, S., and Shamir, O. (2020). Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning*, pages 6682–6691.

Merino, A. and Mütze, T. (2021). Combinatorial generation via permutation languages. iii. rectangulations. *arXiv:2103.09333*.

Miller, A. (2009). Asymptotic bounds for permutations containing many different patterns. *J. Comb. Theory, Ser. A*, 116(1):92–108.

Miller, K., Jordan, M. I., and Griffiths, T. L. (2009). Nonparametric latent feature models for link prediction. pages 1276–1284.

Nakano, M., Fujiwara, Y., Kimura, A., Yamada, T., and Ueda, N. (2021). Permuton-induced Chinese restaurant process. In *Advances in Neural Information Processing Systems*.

Nakano, M., Ishiguro, K., Kimura, A., Yamada, T., and Ueda, N. (2014). Rectangular tiling process. In *International Conference on Machine Learning*, pages 361–369.

Nakano, M., Kimura, A., Yamada, T., and Ueda, N. (2020). Baxter permutation process. In *Advances in Neural Information Processing Systems*.

Orbanz, P. (2011). Projective limit random probabilities on Polish spaces. *Electronic Journal of Statistics*, 5:1354 – 1373.

Orbanz, P. and Roy, D. M. (2013). Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:437–461.

Orbanz, P. and Teh, Y. W. (2010). Bayesian nonparametric models. In Sammut, C. and Webb, G. I., editors, *Encyclopedia of Machine Learning*, pages 81–89. Springer.

Radomirovic, S. (2012). A construction of short sequences containing all permutations of a set as subsequences. *Electron. J. Comb.*, 19(4):P31.

Raj, A. and Mishra, S. (2020). Lottery ticket hypothesis: Placing the k-orrect bets. In *Machine Learning, Optimization, and Data Science*, volume 12566 of *Lecture Notes in Computer Science*, pages 228–239. Springer.

Reading, N. (2012). Generic rectangulations. *European Journal of Combinatorics*, 33(4):610–623.

Rodriguez, A. and Ghosh, K. (2009). Nested partition models. Technical report, JackBaskin School of Engineering.

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

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

Shan, H. and Banerjee, A. (2008). Bayesian coclustering. In *IEEE International Conference on Data Mining*, pages 530–539.

Teh, Y. W. (2010). Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer.

Teh, Y. W. and Jordan, M. I. (2010). Hierarchical Bayesian nonparametric models with applications. In Hjort, N., Holmes, C., Müller, P., and Walker, S., editors, *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press.

Zafarani, R. and Liu., H. (2009). Social computing data repository at ASU.

Supplementary Material: Nonparametric Relational Models with Superrectangulation

A Detailed description of super Bayesian relational model

We will describe the details of the super Bayesian relational model introduced in **Section 1 (Introduction)** and used in the experiments in **Section 4 (Super Bayesian analysis)**.

A.1 Generative model of super Bayesian relational model

The basic strategy for constructing a super Bayesian relational model is to use the Aldous-Hoover-Kallenberg (AHK) representation theorem (Aldous, 1981; Hoover, 1979; Kallenberg, 1989), as in the standard Bayesian nonparametric (BNP) relational models. However, what is different from the BNP models is that the random rectangular partition R on $[0,1] \times [0,1]$, which is an intermediate random function in the AHK representation, is replaced by a deterministic superrectangulation. We suppose that the input observation relational data $\boldsymbol{Z} = (Z_{i,j})_{N \times M}$ consists of categorical elements, i.e., $Z_{i,j} \in \{1, 2, \dots, D\}$ $(D \in \mathbb{N})$. The generative probabilistic model of the super Bayesian relational model can be expressed as follows. First, we set a superrectangulation as the rectangular partition R of $[0,1] \times [0,1]$. We suppose that the rectangulation R consists of blocks indexed by $k = 1, 2, \dots$. Next, we draw two atomic random measures on $[0,1]$, $G^{(\mathrm{row})} \sim \mathrm{DP}(\alpha_1, \mathrm{Uniform}([0,1]))$ and $G^{(\mathrm{column})} \sim \mathrm{DP}(\alpha_2, \mathrm{Uniform}([0,1]))$, where α_1 and α_2 are tunable positive real variables, and $\mathrm{DP}(\alpha, G_0)$ is the Dirichlet process (Ferguson, 1973) with the concentration parameter $\alpha > 0$ and the base measure G_0. Then, we draw the virtual coordinate on $[0,1]$ of the ith row from $U_i^{(\mathrm{row})} \sim G^{(\mathrm{row})}$ $(i = 1, 2, \dots, N)$, and draw the virtual coordinate on $[0,1]$ of the jth column from $U_j^{(\mathrm{column})} \sim G^{(\mathrm{column})}$. The cluster to which the ith row and jth column elements of the input matrix belong is determined by the rectangular block to which $(U_j^{(\mathrm{column})}, U_i^{(\mathrm{row})})$ belong. We denote the block index to which $(U_j^{(\mathrm{column})}, U_i^{(\mathrm{row})})$ belongs by $k(U_j^{(\mathrm{column})}, U_i^{(\mathrm{row})})$. Each block (indexed by $k = 1, 2, \dots$) has a latent Dirichlet random variable $\vartheta_k \sim \mathrm{Dirichlet}(\boldsymbol{\alpha}_0)$ $(k = 1, 2, \dots)$, where $\boldsymbol{\alpha}_0$ is a D-dimensional non-negative hyper parameter. Finally, each element $Z_{i,j}$ is drawn from $\mathrm{Categorical}\left(\vartheta_{k(U_j^{(\mathrm{column})}, U_i^{(\mathrm{row})})}\right)$. In short, the generative probabilistic model can be summarized as follows:

$$
\begin{aligned}
R &\leftarrow \text{Superrectangulation} & \vartheta_k &\underset{\text{i.i.d.}}{\sim} \mathrm{Dirichlet}(\boldsymbol{\alpha}_0) \quad (k = 1, 2, \dots) \\
G^{(\mathrm{row})} &\sim \mathrm{DP}(\alpha_1, \mathrm{Uniform}([0,1])) & G^{(\mathrm{column})} &\sim \mathrm{DP}(\alpha_2, \mathrm{Uniform}([0,1])) \\
U_i^{(\mathrm{row})} &\underset{\text{i.i.d.}}{\sim} G^{(\mathrm{row})} \quad (i = 1, 2, \dots, N) & U_j^{(\mathrm{column})} &\underset{\text{i.i.d.}}{\sim} G^{(\mathrm{column})} \quad (j = 1, 2, \dots, M) \\
Z_{i,j} &\underset{\text{i.i.d.}}{\sim} \mathrm{Categorical}\left(\vartheta_{k(U_j^{(\mathrm{column})}, U_i^{(\mathrm{row})})}\right) & &\quad (i = 1, 2, \dots, N, \quad j = 1, 2, \dots, M)
\end{aligned}
$$

Again, it is important to note that the rectangular partition R of $[0,1] \times [0,1]$ is not a random variable to be estimated, but is definitively fixed.

A.2 Bayesian inference for super Bayesian relational model

For super Bayesian relational models, it is possible to derive Bayesian inference algorithms that are very easy to implement. This is a very important property considering the history of BNP relational models. The infinite relational model (IRM) (Kemp et al., 2006), which is also the origin of the relational model, was indeed able to derive a simple Bayesian inference algorithm based on the Gibbs sampling method due to the simplicity of the model (i.e., the product of the Chinese restaurant processes). However, the expressive power of IRM for rectangular partitioning is very low, and since then, models with higher expressive power have been devised, including the Mondrian process (Roy and Teh, 2009), the rectangular tiling process (Nakano et al., 2014), the block-breaking process (Nakano et al., 2020), and the permuton-induced Chinese restaurant process (Nakano

et al., 2021). However, Bayesian inference algorithms for these extended models have become very complicated to implement. On the other hand, Bayesian inference for super Bayesian relational models can go back to its roots and naively lead to Gibbs sampling in a form similar to IRM.

The goal of Bayesian inference is to estimate the posterior probabilities of parameters $\boldsymbol{U}^{(\mathrm{row})} := \left(U_i^{(\mathrm{row})}\right)_{i\in\{1,\dots,N\}}$ and $\boldsymbol{U}^{(\mathrm{column})} := \left(U_j^{(\mathrm{column})}\right)_{j\in\{1,\dots,M\}}$ given the observed data $\boldsymbol{Z}$. By marginalizing out random atomic measures $G^{(\mathrm{row})}$ and $G^{(\mathrm{column})}$, the joint probability density can be obtained as follows.

$$p\left(\boldsymbol{Z},\boldsymbol{U}^{(\mathrm{row})},\boldsymbol{U}^{(\mathrm{column})} \mid R,\boldsymbol{\alpha}_0,\alpha_1,\alpha_2\right) = p_{\mathrm{CRP}}\left(\boldsymbol{U}^{(\mathrm{row})} \mid \alpha_1\right)\cdot p_{\mathrm{CRP}}\left(\boldsymbol{U}^{(\mathrm{column})} \mid \alpha_2\right) \times p_{\mathrm{obs.}}\left(\boldsymbol{Z} \mid R,\boldsymbol{U}^{(\mathrm{row})},\boldsymbol{U}^{(\mathrm{column})},\boldsymbol{\alpha}_0\right), \tag{3}$$

where each of these terms will be discussed in detail immediately after this. For the first term $p_{\mathrm{CRP}}\left(\boldsymbol{U}^{(\mathrm{row})} \mid \alpha_1\right)$ is the posterior probabilities induced from the standard CRPs, that is, for $u_r \underset{\text{i.i.d.}}{\sim} \mathrm{Uniform}([0,1])$ $(r=1,2,\dots)$,

$$\mathbb{P}\left[U_i^{(\mathrm{row})} = u_r \;\middle|\; \boldsymbol{U}_{-i}^{(\mathrm{row})},\alpha_1\right] = \begin{cases} \dfrac{\mathcal{N}_r}{N+\alpha_1} & (\mathcal{N}_r>0) \\ \dfrac{\alpha_1}{N+\alpha_1} & (\text{otherwise}) \end{cases}, \tag{4}$$

where $\boldsymbol{U}_{-i}^{(\mathrm{row})} := (U_1^{(\mathrm{row})},\dots,U_{i-1}^{(\mathrm{row})},U_{i+1}^{(\mathrm{row})},\dots,U_N^{(\mathrm{row})})$, and $\mathcal{N}_r$ denotes the number of $\left\{i' \;\middle|\; U_{i'}^{(\mathrm{row})} = u_r, i'\neq i\right\}$. Similarly, for the second term $p_{\mathrm{CRP}}\left(\boldsymbol{U}^{(\mathrm{column})} \mid \alpha_2\right)$ is the posterior probabilities induced from the standard CRPs, that is, for $u_c \underset{\text{i.i.d.}}{\sim} \mathrm{Uniform}([0,1])$ $(c=1,2,\dots)$,

$$\mathbb{P}\left[U_j^{(\mathrm{column})} = u_c \;\middle|\; \boldsymbol{U}_{-j}^{(\mathrm{column})},\alpha_2\right] = \begin{cases} \dfrac{\mathcal{M}_c}{N+\alpha_2} & (\mathcal{M}_c>0) \\ \dfrac{\alpha_2}{M+\alpha_2} & (\text{otherwise}) \end{cases}, \tag{5}$$

where $\boldsymbol{U}_{-j}^{(\mathrm{row})} := (U_1^{(\mathrm{column})},\dots,U_{j-1}^{(\mathrm{column})},U_{j+1}^{(\mathrm{column})},\dots,U_N^{(\mathrm{column})})$, and $\mathcal{M}_c$ denotes the number of $\left\{j' \;\middle|\; U_{j'}^{(\mathrm{column})} = u_c, j'\neq j\right\}$. Finally, the third term is

$$p_{\mathrm{obs.}}\left(\boldsymbol{Z} \mid R,\boldsymbol{U}^{(\mathrm{row})},\boldsymbol{U}^{(\mathrm{column})},\boldsymbol{\alpha}_0\right) \propto \prod_{k=1}^{\infty}\left(\frac{\Gamma(D\alpha_0)}{\Gamma(D\alpha_0+\sum_{d=1}^{D}\mathcal{L}_{k,d})}\prod_{d=1}^{D}\frac{\Gamma(\alpha_0+\mathcal{L}_{k,d})}{\Gamma(\alpha_0)}\right), \tag{6}$$

where $\mathcal{L}_{k,d}$ denotes the number of elements in both the k-th block and the d-th category of the categorical distribution.

The Gibbs sampling algorithm for the super Bayesian relational model described in Section A.1 can be described as follows. We will iteratively repeat the following two update rules:

- For each $i=1,2,\dots,N$, we iteratively draw a new sample of $U_i^{(\mathrm{row})}$ from the conditional probability distribution on $U_i^{(\mathrm{row})}$ obtained from Equation (3).
- For each $j=1,2,\dots,M$, we iteratively draw a new sample of $U_j^{(\mathrm{column})}$ from the conditional probability distribution on $U_j^{(\mathrm{column})}$ obtained from Equation (3).

It is important to emphasize that the conditional probability distribution for each $U_i^{(\mathrm{row})}$ and $U_j^{(\mathrm{column})}$ can be computed exactly (without approximation), since the rectangular partition R of $[0,1]\times[0,1]$ is fixed.

B Proofs and additional notes omitted in Section 3 (Superrectangulation)

In this section, we give proofs for all the propositions in **Section 3 (Superrectangulation)** of the main text, and supplement them with some remarks, conjectures and hypotheses. The goal throughout this section is

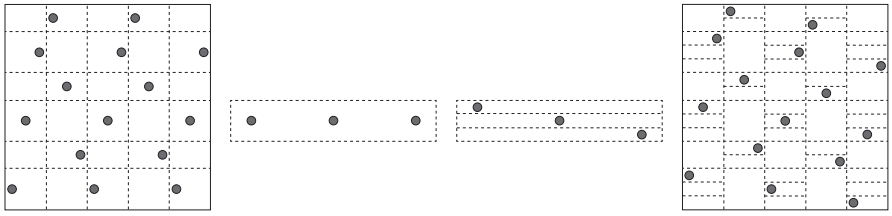


Figure 11: Illustration of breaking ties operation. This operation can be used to convert words into permutations. **Left**: Word. **Second from left**: Tie. **Third from left**: Breaking ties. **Right**: Corresponding permutation.

to construct a universal object that we call *superrectangulation*, which *contains* within itself all rectangular partitions with n blocks. In order to make what turns out and what remains unresolved more clear, we would like to distinguish between two cases of the superpermutation:

- **Superrectangulation**$^{\mathrm{d}}$ - If a rectangular partition of $[0,1] \times [0,1]$ can generate every *diagonal* rectangulation of the number of blocks n by restriction (i.e., the restriction operation described in **Section 3 (Superrectangulation)** of the main text, we call it a superrectangulation$^{\mathrm{d}}$.
- **Superrectangulation**$^{\mathrm{g}}$ - If a rectangular partition of $[0,1] \times [0,1]$ can generate every *generic* rectangulation of the number of blocks n by restriction, we call it a superrectangulation$^{\mathrm{g}}$.

When there is no need to explicitly distinguish between the superrectangulation$^{\mathrm{d}}$ and the superrectangulation$^{\mathrm{g}}$, we will simply write them as the superrectangulation. It follows immediately from the definition that if a partition is a superpartition$^{\mathrm{g}}$, then it is also a superpartition$^{\mathrm{d}}$. This is because the set of diagonal rectangulations is a subset of the set of generic rectangulations, so if a partition contains all generic rectangulations, then it also contains all diagonal rectangulations. For simplicity of notation, we would like to write the maps that convert permutations to rectangular partitions as follows. These symbols are in accordance with Reading (2012) that introduced these transformations.

- **Map ρ from permutations to diagonal rectangulations** - Figure 5 shows the details of this transformation. For every permutation p, we can obtain a unique diagonal rectangulation $\rho(p)$.
- **Map γ from permutations to generic rectangulations** - **Figure 6** of the main text shows the details of this transformation. For every permutation p, we can obtain a unique diagonal rectangulation $\gamma(p)$.

The first naive observation is that the superrectangulation$^{\mathrm{g}}$ does indeed exist.

Proposition B.1. ***(Proposition 3.2 in the main text)*** *There is a superrectangulation^{g} that has $n \cdot n!$ blocks.*

Proof. (Proposition B.1.) As a constructive proof, we indeed construct a superrectangulation$^{\mathrm{g}}$ with $n \cdot n!$ blocks. There are $n!$ permutations of length n, and for all of them, we construct a generic rectangulation of block number n using the mapping (described in **Section 2.3** in the main text). The $n!$ rectangular partitions created in this way cover all the general rectangular partitions (**Proposition 2.5** in the main text). Finally, we generate one rectangular partition on $[0,1] \times [0,1]$ with $n!$ blocks, and for each block, we fill in one of the $n!$. This rectangular partition on $[0,1] \times [0,1]$ has $n \cdot n!$ blocks and satisfies the requirement of the superrectangulation since it can generate arbitrary generic rectangulations with n blocks by restriction. We have completed the proof. □

The first observation above is very naive, since it is based on a method that exhausts all rectangular partitions through permutations, but we can reduce the size of the superrectangulation a bit by making the expression of the permutation for a superpermutation. This result is non-trivial because it is not obvious whether the rectangular partition corresponding to a superpermutation is a superrectangulation or not. In order to construct this superrectangulation, we need to convert the superpermutation (word) to a permutation, and then convert the permutation to a rectangulation. Therefore, we will first introduce the *breaking ties* operation to convert words into permutations.

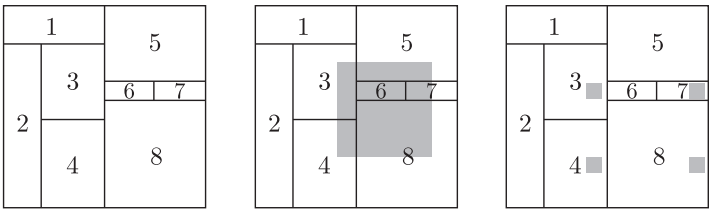


Figure 12: Difficulty of extracting generic rectangulations by restriction from generic rectangulations

- **Breaking ties operation** - Figure 11 shows an intuitive illustration of this operation. We suppose that a word w whose alphabet is natural numbers $\mathbb{N}$ has I ties $w(i_1), w(i_2), \ldots, w(i_{m_i})$ $(i = 1, 2, \ldots, I)$, that is, $w(i_1) = w(i_2) = \cdots = w(i_{m_i})$ $(i \in \{1, \ldots, I\})$. Then, we obtain a word w' whose alphabet is a rational number $\mathbb{Q}$ obtained as follows. First, we set up a copy of the word w as the word w'. Next, we make some modifications to the word w'. For each tie $w'(i_1), w'(i_2), \ldots, w'(i_{m_i})$, we modify w' as

$$w'(i_1) \leftarrow w'(i_1) + \frac{m_i - 1}{m_i}, \quad w'(i_2) \leftarrow w'(i_2) + \frac{m_i - 2}{m_i}, \quad \ldots, w'(i_{m_i}) \leftarrow w'(i_{m_i}) + \frac{m_i - m_i}{m_i}. \tag{7}$$

  Now that we have obtained w' with rational numbers as its alphabet, the last step is to convert it to a permutation. For the word w', there is a unique permutation p that is an order-isomorphic to w'. We will denote such a transformation by the map $\beta : w \mapsto p$ and call it the *breaking ties* operation.

In addition, we also use the *inverse* operation of a permutation:

- **Inverse of permutations** - A permutation p can be viewed as a map from $\mathbb{N}$ to $\mathbb{N}$. The inverse map p^{-1} is then called the inverse of the permutation p. For example, the inverse of the permutation $p = 35241$ is $p^{-1} = 53142$. Using the two-line notation for permutations, the inverse p^{-1} of a permutation p can be obtained more intuitively by reading the permutation p "upside down":

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 2 & 4 & 1 \end{pmatrix}, \qquad p^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 1 & 4 & 2 \end{pmatrix} \tag{8}$$

  In addition, the inverse of a permutation corresponds in geometric representation to the operation of rotating 90 degrees counterclockwise and then flipping horizontally, as shown in Figure 10 (middle).

The superrectangulations described in Proposition B.1 are somewhat difficult to handle in practical applications of relational data analysis, since they are too large to be practical. Therefore, we need a strategy to reduce the size of the superrectangulation more dramatically. It is worth recalling that the size of the superpermutation could be dramatically reduced by using the zigzag word. Therefore, the strategy we can come up with is to convert the zigzag word into a rectangular partition. Will the rectangular partition obtained by this transformation from the zigzag word be a superrectangulation? The issues of interest to us can be summarized as follows:

Problem B.2. *Let w be the zigzag word containing all permutations of length n as subsequences, as defined in* ***Section 2.1*** *of the main text.*

(i) Whether or not the the diagonal rectangulation $\rho(\beta(w)^{-1})$ is a superrectangulation^d?

(ii) Whether or not the the diagonal rectangulation $\rho(\beta(w))$ is a superrectangulation^d?

(iii) Whether or not the concatenation of the generic rectangulation $\gamma(\beta(w))$ and the generic rectangulation $\gamma(\beta(w)^{-1})$ constitutes a superrectangulation^g?

We will discuss the reasons for considering both $\beta(w)$ and $\beta(w)^{-1}$ in more detail later. Before considering these issues, we can make some simple observations about rectangular partitioning corresponding to zigzag words.

Proposition B.3. *Let w be the zigzag word containing all permutations of length n as subsequences. Then, the generic rectangulation $\gamma(\beta(w))$ is equivalent to the diagonal rectangulation $\rho(\beta(w))$. Similarly, the generic rectangulation $\gamma(\beta(w)^{-1})$ is equivalent to the diagonal rectangulation $\rho(\beta(w)^{-1})$.*

Proof. (Proposition B.3.) For each wall in the generic rectangulation $\gamma(\beta(w)^{-1})$, by the definition of the zigzag word w and the configuration of γ, if the wall extend vertically, then vertexes on it correspond to the descending part of the zigzag word w. Similarly, if the wall extend horizontally, then vertexes on it correspond to the ascending part of the zigzag word w. Therefore, by the configuration of ρ, we can check that $\gamma(\beta(w)^{-1}) = \rho(\beta(w)^{-1})$. In the same way, we can check that $\gamma(\beta(w)) = \rho(\beta(w))$. □

As a result, Problem B.4 can be reformulated as follows:

Problem B.4. *Let w be the zigzag word containing all permutations of length n as subsequences, as defined in Section 2.1 of the main text.*

(i) Whether or not the the diagonal rectangulation $\gamma(\beta(w)^{-1}) = \rho(\beta(w)^{-1})$ is a superrectangulation^d?

(ii) Whether or not the the diagonal rectangulation $\gamma(\beta(w)) = \rho(\beta(w))$ is a superrectangulation^d?

(iii) Whether or not the concatenation of the diagonal rectangulation $\gamma(\beta(w)) = \rho(\beta(w))$ and the diagonal rectangulation $\gamma(\beta(w)^{-1}) = \rho(\beta(w)^{-1})$ constitutes a superrectangulation^g?

In the early stages of this study, we were initially optimistic that since the zigzag word contains every short permutation p as subsequences, $\gamma(\beta(w)^{-1})$, $\gamma(\beta(w))$ and their concatenation may also contain their corresponding rectangular partitions to p. However, unfortunately, this is not true. The reason for this is that the subsequence extraction from the zigzag word does not necessarily correspond to the restriction operation in the zigzag rectangulation. More precisely, it can be explained as follows:

Proposition B.5. *The relationship between the following statements is considered:*

(S1) The permutation p is contained in the zigzag word w as a subsequence.

(S2) The rectangular partition $\gamma(p)$ corresponding to the permutation p is contained as a restriction in $\gamma(\beta(w))$.

At this time, $(S1) \Leftarrow (S2)$ is valid, but $(S1) \Rightarrow (S2)$ is not.

Proof. (Proposition B.5.) First, we show the $(S1) \Leftarrow (S2)$ part. For any given pair of blocks in the rectangulation $\gamma(\beta(w))$, the positioning of the top-left corner (i.e., which is on top and which is on the left) is not changed by the restriction operation. Thus, if the partition generated by restriction is to be partition $\gamma(p)$, then we can choose the word which is order-isomorphic to the permutation p, by extracting the indexes of the blocks referred to during restriction from the zigzag word w.

Second, we show the $(S1) \not\Rightarrow (S2)$ part. This can be done by actually discovering counterexamples. Figure 9 shows the counterexamples for the case of $\gamma(\beta(w))$. □

Finally, we will discuss the reasons why we would want to consider both $\gamma(\beta(w))$ and $\gamma(\beta(w)^{-1})$. As implied by Figure 9, for a subsequence in the zigzag word to be extractable by restriction as the zigzag rectangulation consisting of blocks with those indices, the subsequence must be concentrated in a small area in either the horizontal or vertical direction. For example, as shown in Figure 9 (d), if a subsequence consists of elements scattered in various positions in a zigzag word, the corresponding blocks are also scattered in various positions in the zigzag rectangulation, and cannot be extracted as a rectangular partition by the restriction operation. From this observation, it becomes important to consider both $\gamma(\beta(w))$ and $\gamma(\beta(w)^{-1})$. In the case of $\gamma(\beta(w))$, if the subsequence is concentrated in a narrow horizontal region, there is a high probability that it can be extracted as a rectangular partition. On the other hand, for $\gamma(\beta(w)^{-1})$, if the subsequence is concentrated in a narrow area in the vertical direction, it is highly likely to be extracted as a rectangular partition. As a result, the concatenation of $\gamma(\beta(w))$ and $\gamma(\beta(w)^{-1})$ is expected to be able to handle both cases, making it possible to correspond subsequences to the restriction operations on the zigzag rectangulation. As a further clue to another point of view, we find that the run required for a subsequence to appear in a zigzag word is almost always not very large. Based on these clues, we expect that the concatenation of partitions $\gamma(\beta(w))$ and $\gamma(\beta(w)^{-1})$ may be a superrectangulation, but we would like to leave the answer to this question as an open question.

C Notes on experimental setup

Dataset - We used four social network datasets Zafarani and Liu. (2009): (1) **Wiki** (top-left) (dat, a), consisting of 7115 nodes and 103689 edges. (2) **Facebook** (top-right) (dat, b), consisting of 4039 nodes and 88234. (3) **Twitter** (bottom-left) (dat, c), consisting of 81306 nodes and 1768149 edges. (4) **Epinion** (bottom-right) (dat, d), consisting of 75879 nodes and 508837 edges. For each data, we selected the top 1000 active nodes based on their interactions with others; subsequently we randomly sampled 500×500 matrix to construct the relational data, as in (Fan et al., 2020; Nakano et al., 2020). For model comparison, we held out 20% cells of the input data for testing, and each model was trained by the MCMC using the remaining 80% of the cells.

Relational models - We compare the super Bayesian relational model with the BNP relational models based on rectangular partitioning, such as the Mondrian process (MP) (Roy and Teh, 2009), the block-breaking process (BBP) (Nakano et al., 2020), and the permuton-induced Chinese restaurant process (PCRP) (Nakano et al., 2021). For **MP** (Roy and Teh, 2009), the intermediate random function of the AHK representation is drawn from the MP, the budget parameter of which is set to 3, as in Fan et al. (2020); Nakano et al. (2020). For **BBP** (Nakano et al., 2020), we used the default settings provided by the original code (Nakano et al., 2020). For **PCRP** (Nakano et al., 2021), we employed the non-informative Gamma prior on the concentration parameter for CRP and used the uniform permuton. For our SB models, the number of blocks had to be set in advance. Since it was difficult for us to determine the exact optimal parameter settings due to our computing environment, we generated them from Poisson(50) based on a simple preliminary study in this experiment. We recognize that how to determine the optimal size of the superrectangulation is an important problem to be solved in the near future. Through our preliminary investigations, we expect that the size of the superrectangulation may contribute little to the performance of the model, but we believe that exhaustive experiments are needed in the future to provide stronger support for this observation.