Noname manuscript No.
(will be inserted by the editor)

Smoothing graphons for modelling exchangeable relational data

Yaqiong Li · Xuhui Fan* · Ling Chen · Bin Li · Scott A. Sisson

Received: date / Accepted: date

Abstract Modelling exchangeable relational data can be described appropriately in *graphon theory*. Most Bayesian methods for modelling exchangeable relational data can be attributed to this framework by exploiting different forms of graphons. However, the graphons adopted by existing Bayesian methods are either piecewise-constant functions, which are insufficiently flexible for accurate modelling of the relational data, or are complicated continuous functions, which incur heavy computational costs for inference. In this work, we overcome these two shortcomings by smoothing piecewise-constant graphons, which permits continuous intensity values for describing relations, without impractically increasing computational costs. In particular, we focus on the Bayesian Stochastic Block Model (SBM) and demonstrate how to adapt the piecewise-constant SBM graphon to the smoothed version. We first propose the Integrated Smoothing Graphon (ISG) which introduces one smoothing parameter to the SBM graphon to generate continuous relational intensity values. Then, we further develop the Latent Feature Smoothing Graphon (LFSG), which improves the ISG, by introducing auxiliary hidden labels to decompose the calculation of the ISG intensity and enable efficient inference. Experimental results on real-world data sets validate the advantages of applying smoothing strategies to the Stochastic Block Model, demonstrating that smoothing graphons can greatly improve AUC and precision for link prediction without increasing computational complexity.

Y. Li, L. Chen
Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia.
E-mail: yaqiong.li@student.uts.edu.au,ling.chen@uts.edu.au

X. Fan, S. A. Sisson
UNSW Data Science Hub & School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2052, Australia.
E-mail: xuhui.fan@unsw.edu.au,scott.sisson@unsw.edu.au

B. Li
School of Computer Science, Fudan University, Shanghai, China.
E-mail: libin@fudan.edu.cn

Keywords Bayesian methods · Exchangeable Relational Data · Graphon

1 Introduction

Exchangeable relational data [18,27,38], such as tensor data [31,42] and collaborative filtering data [22,24,41], are commonly observed in many real-world applications. In general, exchangeable relational data describe the relationship between two or more nodes (e.g. friendship linkages in social networks; user-item rating matrices in recommendation systems; and protein-to-protein interactions in computational biology), where exchangeability refers to the phenomenon that the joint distribution over all observed relations remains invariant under node permutations. Techniques for modelling exchangeable relational data include node partitioning to form "homogeneous blocks" [19,27,36], graph embedding methods to generate low-dimensional representations [8, 12, 30], and optimization strategies to minimize prediction errors [26, 39].

Graphon theory [23, 28, 29] has recently been proposed as a unified theoretical framework for modelling exchangeable relational data. In graphon theory, each relation from a node i to another node j is represented by an *intensity* value generated by a *graphon function*, which maps from the corresponding coordinates of the node pair in a unit square, $(u_i^{(1)}, u_j^{(2)})$, to an intensity value in a unit interval. Many existing Bayesian methods for modelling exchangeable relational data can be described using graphon theory with various graphon functions. Figure 1 illustrates several typical graphon functions, including the Stochastic Block Model (SBM) [19, 27], the Mondrian Process Relational Model (MP-RM) [36], the Rectangular Tiling Process Relational Model (RTP-RM) [25], and the Gaussian Process Prior Relational Model (GP-RM) [28].

The simplest of these graphon functions is the regular-grid piecewise-constant graphon (Figure 1, left). Generally, it is constructed from two-independent partition processes in a two-dimensional space. The resulting orthogonal crossover between these dimensions produces a regular grid partition in the space. Typical regular-grid partition models include the SBM [27] and its infinite states variant, the Infinite Relational Model (IRM) [19]. The SBM uses a Dirichlet distribution (or Dirichlet process for the IRM) to independently generate a finite (or infinite for the IRM) number of segments in each dimension.

The Mondrian process relational model (MP-RM; Figure 1, centre-left) [34–36] is a representative model which generates k-d tree-structured piecewise-constant graphons. In general, the Mondrian process recursively generates axis-aligned cuts in the unit square and partitions the space in a hierarchical fashion known as a k-d tree. The tree-structure is regulated by attaching an exponentially distributed cost to each axis-aligned cut, so that the tree-generation process terminates when the accumulated cost exceeds a budget value. The Binary Space Partitioning-Tree process relational model (BSP-RM) [14,15] also generates tree-structured partitions. The difference between the BSP-RM and the MP-RM is that the BSP-RM uses two dimensions to form oblique cuts and thus generate convex polyhedron-shaped blocks. These oblique cuts concentrate more on describing the inter-dimensional dependency and can produce more efficient space partitions.

The Rectangular Tiling process relational model (RTP-RM; Figure 1, centre-right) [25] produces a flat partition structure on a two-dimensional array by assigning each entry to an existing block or a new block in sequence, without violating the rectangular restriction of the blocks. By relaxing the restrictions of the hierarchical or regular-grid structure, the RTP-RM aims to provide more flexibility in block generation. However, the process of generating blocks is quite complicated for practical use. As a result, while the hierarchical and regular-grid partition models can be used for continuous space and multi-dimensional arrays (after trivial modifications), the RTP-RM is restricted to (discrete) arrays only.

The Rectangular Bounding process relational model (RBP-RM) [13] uses a bounding strategy to generate rectangular blocks in the space. In contrast to the previously described cutting strategies, the RBP-RM concentrates more on the important regions of the space and avoids over-modelling sparse and noisy regions. In the RBP-RM, the number of possible intensities is equivalent to the number of blocks, which follows a Poisson distribution and is almost certainly finite.

The Gaussian process relational model (GP-RM; Figure 1, right) [23] uses a prior over a random function in the unit square to form a continuous graphon. In this way it can potentially generate desired continuous intensity values via the graphon function. However, the computational cost of the GP-RM is the same as that of the Gaussian process, which scales to the cubic of the number of nodes (n).

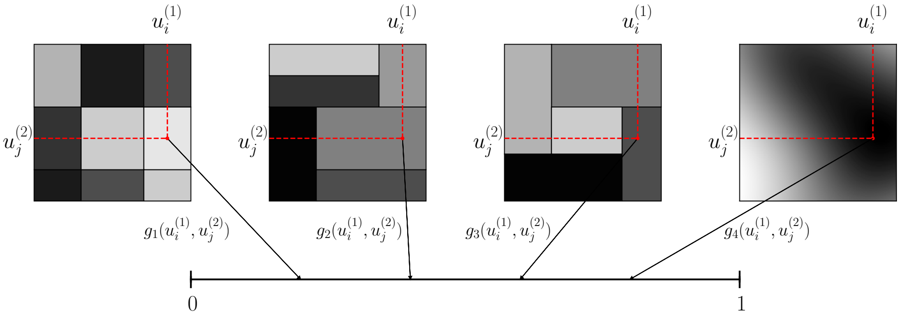
$u_i^{(1)}$
$u_j^{(2)}$
$g_1(u_i^{(1)}, u_j^{(2)})$
$u_i^{(1)}$
$u_j^{(2)}$
$g_2(u_i^{(1)}, u_j^{(2)})$
$u_i^{(1)}$
$u_j^{(2)}$
$g_3(u_i^{(1)}, u_j^{(2)})$
$u_i^{(1)}$
$u_j^{(2)}$
$g_4(u_i^{(1)}, u_j^{(2)})$
0
1

Fig. 1 Visualisation of Bayesian graphon-construction methods for modelling exchangeable relational data. From left to right: the Stochastic Block Model (SBM); the Mondrian Process Relational Model (MP-RM); the Rectangular Tiling Process Relational Model (RTP-RM) and the Gaussian Process Prior Relational Model (GP-RM). For any pair of node coordinates $(u_i^{(1)}, u_j^{(2)})$ the relation intensity is mapped from the unit square to a unit interval using a graphon function (denoted $g_1, \ldots, g_4$), where a darker colour observed in the unit square represents a higher mapped intensity in the unit interval.

These existing models can be broadly classified into two categories. The first category, which includes the SBM, MP-RM, RTP-RM and RBP-RM models, uses node-partitioning strategies to construct the relational model. By partitioning the set of nodes into groups along node co-ordinate margins, blocks can be constructed from these marginal groups that partition the full-dimensional co-ordinate space according to a given construction method (Figure 1). These models then assume that the relation intensity for node pairs is constant within each block. That is, the graphon function that generates intensity values over node co-ordinate space is constructed in a piecewise-constant manner. However, such piecewise-constant graphons can only provide limited modelling flexibility with a fixed and constant number of intensity values (i.e. equivalent to the number of blocks). As a result, they are restricted in their ability to model the ground-truth well. The second category of relational models, which includes the GP-RM, aims to address this limitation as the graphon function can provide continuous intensity values. However, the computational complexity for estimating this graphon function is proportional to the cubic of the number of nodes, which makes it practically non-viable for medium or large sized datasets.

In this paper, we propose to apply a smoothing procedure to piecewise-constant graphons to form *smoothing graphons*, which will naturally permit continuous intensity values for describing relations without impractically increasing computational costs. As the Stochastic Block Model is one of the most popular Bayesian methods for modelling exchangeable relational data, we focus on developing smoothing strategies within the piecewise-constant SBM graphon framework. In particular, we develop two variant smoothing strategies for the SBM: the Integrated Smoothing Graphon (ISG) and the Latent Feature Smoothing Graphon (LFSG).

- ISG: In contrast to existing piecewise-constant graphons, which determine the intensity value based only on the block within which a node pair resides, the ISG alternatively calculates a mixture intensity for each pair of nodes by taking into account the intensities of all other blocks. The resulting mixture graphon function is constructed so that its output values are continuous.
- LFSG: This strategy introduces auxiliary pairwise hidden labels to decompose the calculation of the mixture intensity used in the ISG, in order to enable efficient inference. In addition, the introduction of these labels allows each node to belong to multiple groups in each dimension (e.g. a user might interact with different people by playing different roles in a social network), which provides more modelling flexibility compared with the ISG (and existing piecewise-graphons) where each node is assigned to one group only.

Note that while we develop the ISG and LFSG for SBM-based graphons, our smoothing approach can be applied easily to other piecewise-constant graphons. The main contributions of our work are summarised as follows:

- We identify the key limitation of existing piecewise-constant graphons and develop a smoothing strategy to flexibly generate continuous graphon intensity values, which might better reflect the reality of a process.
- We develop the ISG smoothing strategy for the SBM to demonstrate how piecewise-constant graphons can be converted into smoothing graphons.
- We improve on the ISG by devising the LFSG, which achieves the same objective of generating continuous intensity values but without sacrificing computation efficiency. Compared with the ISG where each node belongs to only one group, the LFSG allows each node to belong to multiple groups (e.g. so that the node plays different roles in different relations), and thereby also providing a probabilistic interpretation of node groups.
- We evaluate the performance of our methods on the task of link prediction by comparing with the SBM and other benchmark methods. The experimental results

clearly show that the smoothing graphons can achieve significant performance improvement over piecewise-constant graphons.

2 Preliminaries

2.1 Graphon theory

The Aldous–Hoover theorem ([2, 17]) provides the theoretical foundation for modelling exchangeable multi-dimensional arrays (i.e. exchangeable relational data) conditioned on a stochastic partition model. A random 2-dimensional array is called *separately exchangeable* if its distribution is invariant under separate permutations of rows and columns.

Theorem 1 *[23, 29]: A random array (R_{ij}) is separately exchangeable if and only if it can be represented as follows: there exists a random measurable function $F : [0,1]^3 \mapsto \{0,1\}$ such that $(R_{ij}) \overset{d}{=} \Big(F(u_i^{(1)}, u_j^{(2)}, \nu_{ij})\Big)$, where $\{u_i^{(1)}\}_i, \{u_j^{(2)}\}_j$ and $\{\nu_{ij}\}_{i,j}$ are two sequences and an array of i.i.d. uniform random variables in $[0,1]$, respectively.*

Many existing Bayesian methods for modelling exchangeable relational data can be represented as in Theorem 1, using specific forms of the mapping function F. For instance, as illustrated in Figure 1, given the uniformly distributed node coordinates $(u_i^{(1)}, u_j^{(2)})$, the SBM corresponds to F being a regular-grid constant graphon, in which the partitions along each dimension are crossed over to form the blocks; the MP-RM characterizes an F being a k-d tree-structured constant graphon, in which the blocks are hierarchically aligned; the RTP-RM assumes an F being an arbitrary rectangle constant graphon, in which the blocks are floor-plan aligned; and the GP-RM lets the F perform a continuous two-dimensional function. While taking different forms, these graphon functions commonly map from pairs of node coordinates in a unit square to intensity values in a unit interval. For the above piecewise-constant graphons, we can write the function F as $F(u_i^{(1)}, u_j^{(2)}|\{\omega_k, \square_k\}) = \sum_k \omega_k \cdot \mathbf{1}((u_i^{(1)}, u_j^{(2)}) \in \square_k)$, where $\square_k$ is the kth block and ω_k refers to the intensity variable of $\square_k$. As shown in Figure 1, the darker colour for the pair of node coordinates indicates the higher intensity in the interval, which corresponds to a larger probability of observing or generating the relationship between the pair of nodes.

2.2 Piecewise-constant graphons and their limitations

Many alternative piecewise-constant graphons can be implemented to model exchangeable relational data R, where R is a binary adjacency matrix which can be either directed (asymmetric) or undirected (symmetric). Here we consider the more complicated situation where R is a $n \times n$ asymmetric matrix with $R_{ji} \neq R_{ij}$ (the extension of our method to the symmetric case is straightforward). For any two nodes in R, if node i is related to node j then $R_{ij} = 1$, otherwise $R_{ij} = 0$.

We take the SBM as an illustrative example. In a two-dimensional SBM, there are two distributions generating the groups, $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)} \sim$ Dirichlet($\boldsymbol{\alpha}_{1\times K}$), where K is the number of groups and $\boldsymbol{\alpha}_{1\times K}$ is the K-vector concentration parameter. Each node $i \in \{1,\ldots,n\}$ is associated with two hidden labels $z_i^{(1)}, z_i^{(2)} \in \{1,\ldots,K\}$, and $\{z_i^{(1)}\}_i \sim$ Categorical($\boldsymbol{\theta}^{(1)}$), $\{z_i^{(2)}\}_i \sim$ Categorical($\boldsymbol{\theta}^{(2)}$) for $i = 1,\ldots,n$. Hence, $z_i^{(1)}$ and $z_i^{(2)}$ denote the particular groups that node i belongs to in two dimensions, respectively. (That is, $z_i^{(1)}$ is the group of node i when i links to other nodes, and $z_i^{(2)}$ is the group of node i when other nodes link to it.) The relation R_{ij} from node i to node j is then generated based on the interaction between their respective groups $z_i^{(1)}$ and $z_j^{(2)}$.

Let $\boldsymbol{B}$ be a $K \times K$ matrix, where each entry $B_{k_1,k_2} \in [0,1]$ denotes the probability of generating a relation from group k_1 in the first dimension to group k_2 in the second dimension. For $k_1, k_2 = 1,\ldots,K, B_{k_1,k_2} \sim$ Beta(α_0, β_0), where α_0, β_0 are hyper-parameters for $\{B_{k_1,k_2}\}_{k_1,k_2}$. That is, we have $P(R_{ij} = 1 | z_i^{(1)}, z_j^{(2)}, \boldsymbol{B}) = B_{z_i^{(1)}, z_j^{(2)}}$.

Now, consider the SBM from the graphon perspective (Figure 1; left). Let $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}$ be group (or segment) distributions of the two dimensions in a unit square respectively. The generation of hidden labels $z_i^{(1)}$ for node i and $z_j^{(2)}$ for node j proceeds as follows: Uniform random variables $u_i^{(1)}$ and $u_j^{(2)}$ are respectively generated in the first and second dimensions. Then, $z_i^{(1)}$ and $z_j^{(2)}$ can be determined by checking in which particular segments of $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$, $u_i^{(1)}$ and $u_j^{(2)}$ are located respectively. Formally, we have:

$$\begin{aligned} &\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)} \sim \text{Dirichlet}(\boldsymbol{\alpha}_{1\times K}), \quad u_i^{(1)}, u_j^{(2)} \sim \text{Unif}[0,1] \\ &z_i^{(1)} = (\boldsymbol{\theta}^{(1)})^{-1}(u_i^{(1)}), \quad z_j^{(2)} = (\boldsymbol{\theta}^{(2)})^{-1}(u_j^{(2)}), \end{aligned} \tag{1}$$

where $(\boldsymbol{\theta}^{(1)})^{-1}(u_i^{(1)})$ and $(\boldsymbol{\theta}^{(2)})^{-1}(u_j^{(2)})$ respectively map $u_i^{(1)}$ and $u_j^{(2)}$ to particular segments of $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$.

A regular-grid partition ($\boxplus$) can be formed in the unit square by combining the segment distributions $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}$ in two dimensions. Each block in this regular-grid partition is presented in a rectangular shape. Let $L_k^{(1)} = \sum_{k'=1}^{k} \theta_{k'}^{(1)}$ and $L_k^{(2)} = \sum_{k'=1}^{k} \theta_{k'}^{(2)}$ be the accumulated sum of the first k elements of $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ respectively (w.l.o.g. $L_0^{(1)} = L_0^{(2)} = 0, L_K^{(1)} = L_K^{(2)} = 1$). Use $\square_{k_1,k_2} = [L_{k_1-1}^{(1)}, L_{k_1}^{(1)}] \times [L_{k_2-1}^{(2)}, L_{k_2}^{(2)}]$ to represent the (k_1,k_2)th block in the unit square of $[0,1]^2$, such that $\bigcup_{k_1,k_2} \square_{k_1,k_2} = [0,1]^2$. Then, an intensity function defined on the pair (u_i, u_j) can be obtained by the piecewise-constant graphon function

$$g\left(u_i^{(1)}, u_j^{(2)}\right) = \sum_{k_1,k_2} \mathbf{1}((u_i^{(1)}, u_j^{(2)}) \in \square_{k_1,k_2}) \cdot B_{k_1,k_2} \tag{2}$$

where $\mathbf{1}(A) = 1$ if A is true and 0 otherwise, and where $B_{k_1,k_2} \in [0,1]$ is the intensity of the (k_1,k_2)th block. We term (2) the SBM-graphon. Thus, the generative process of the SBM-graphon can be described as:

1. For $k_1, k_2 = 1, \ldots, K$, generate $B_{k_1,k_2} \sim \text{Beta}(\alpha_0, \beta_0)$, where α_0, β_0 are hyperparameters;
2. Generate the segment distributions $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}$ via Equation (1) and form the partition ($\boxplus$) according to combinations of $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}$ in the unit square;
3. Uniformly generate the 1^{st} dimension coordinates $\{u_i^{(1)}\}_{i=1}^n$ and the 2^{nd} dimension coordinates $\{u_i^{(2)}\}_{i=1}^n$ for all nodes;
4. For $i, j = 1, \ldots, n$
   (a) Calculate the intensity $g(u_i^{(1)}, u_j^{(2)})$ according to Equation (2) based on the node coordinates $(u_i^{(1)}, u_j^{(2)})$;
   (b) Generate $R_{ij} \sim \text{Bernoulli}(g(u_i^{(1)}, u_j^{(2)}))$.

Alternatively, if considering the latent labels $(z_i^{(1)}, z_j^{(2)})$ for nodes i and j, then from Equation (1) and the deterministic relations between $z_i^{(1)}, z_j^{(2)}$ and $\boldsymbol{\theta}_i^{(1)}, u_i^{(1)}, \boldsymbol{\theta}_i^{(2)}, u_i^{(2)}$, the intensity value $g(u_i^{(1)}, u_j^{(2)})$ in step 4.(a) is the same as $B_{z_i^{(1)}, z_j^{(2)}}$. As a result, step 4 in the above generative process can be equivalently written as

4. For $i, j = 1, \cdots, n$,
   (a) Generate the latent labels $(z_i^{(1)}, z_j^{(2)})$ via Equation (1);
   (b) Generate $R_{ij} \sim \text{Bernoulli}\left(B_{z_i^{(1)}, z_j^{(2)}}\right)$.

The SBM-graphon has several limitations. To begin with, the SBM-graphon function (Equation (2)) is piecewise-constant. That is, the generated intensities for node pairs are discrete and the number of different intensity values is limited to the number of blocks in the partition ($\boxplus$). Consequently, this leads to an over-simplified description when modelling real relational data, which can result in at least two issues. On the one hand, as long as two nodes belong to the same segment in one dimension, their probability of generating relations with another node are the same even if the distance between the two nodes in that dimension is quite large. Conversely, given two nodes that are close in one dimension but belong to two adjacent segments, their probability of generating relations with another node could be dramatically different, depending on the respective block intensities (e.g., B_{k_1,k_2}).

The second limitation of the SBM-graphon is that it determines the intensity value for a pair of nodes by considering only the block ($\square_{k_1,k_2}$) in which (u_i, u_j) resides. However, the nodes relations with other nodes, especially neighbouring nodes in adjacent blocks, might also be expected to have a certain influence on the generation of the target relation, if one considers the relational data collectively. As a result, perhaps it could be beneficial to consider the interactions that naturally exist among all blocks when generating the relation R_{ij}.

The third limitation of the SBM-graphon is that it provides latent information of node clustering as a side-product through the hidden labels $\{z_i^{(1)}, z_i^{(2)}\}_{i=1}^n$. However, the clustering information might not be ideal because each node is assigned to only one cluster in each dimension. That is, when considering the outgoing relations from node i, it is assumed that node i consistently plays one single role in any relation with other nodes. In fact, in practice a node might play different roles by participating in different relations with different nodes. As a result, it would be more useful and flexible to allow a node to belong to multiple clusters in each dimension.

To address the limitations of piecewise-constant graphons (and in particular, the SBM-graphon), we propose a smoothing strategy to enable piecewise-constant graphons to produce continuous intensity values. The proposed smoothing graphons naturally consider interactions between the partitions and allow each node to play multiple roles in different relations.

3 Main Models

3.1 The Integrated Smoothing Graphon (ISG)

In order to improve on the limitations of the piecewise-constant graphon we first develop the Integrated Smoothing Graphon (ISG), based on the SBM-graphon construction. The piecewise-constant nature of the SBM-graphon is created through the use of an indicator function in (2) that selects only the particular block accommodating the target node pair. Accordingly, we replace the indicator function with an alternative that can produce continuous intensity values. Moreover, to capture the interaction between all blocks, we construct the smoothing graphon function to generate the intensity value as a summation over all block intensities, weighted by the importance of each block. Let $F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)})$ be the weight of the block $\square_{k_1,k_2}$ with respect to $(u_i^{(1)}, u_j^{(2)}) \in [0,1]^2$. The mixture intensity $g\left(u_i^{(1)}, u_j^{(2)}\right)$, used to determine the R_{ij}, can then be represented as

$$g\left(u_i^{(1)}, u_j^{(2)}\right) = \sum_{k_1,k_2} F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)}) \cdot B_{k_1,k_2}, \tag{3}$$

where $\sum_{k_1,k_2} F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)}) = 1$.

The ISG generative process can be summarised as:

1)∼3) The block intensities ($\boldsymbol{B}$), graphon partition ($\boxplus$) and two-dimensional coordinates ($\{u_i^{(1)}, u_i^{(2)}\}_{i=1}^n$) are generated as for the SBM-graphon;

4. For $i, j = 1, \cdots, n$,
   (a) Calculate the mixture intensity $g\left(u_i^{(1)}, u_j^{(2)}\right)$ according to (3) for the node coordinates $(u_i^{(1)}, u_j^{(2)})$;
   (b) Generate $R_{ij} \sim \text{Bernoulli}\left(g\left(u_i^{(1)}, u_j^{(2)}\right)\right)$.

As a consequence, while the SBM-graphon determines the relation intensity based only on the single block in which $(u_i^{(1)}, u_j^{(2)})$ resides, the ISG computes a mixture intensity as a weighted (and normalised) sum of all block intensities. That is, instead of assigning a weight of 1 for one particular block and weights of 0 for all other blocks, the ISG weights the importance of each block with respect to the pair of node coordinates $(u_i^{(1)}, u_j^{(2)})$. As long as the weighting function $F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)})$

is continuous, it follows that the mixture intensity (3) is also continuous. The intensity function (3) then becomes a smoothing graphon function.

The ISG allows the mixture intensity to take any value between the minimum and maximum of all block intensities. As a result, the ISG provides more modelling flexibility compared to the SBM-graphon, where only limited discrete intensity values (equivalent to the number of blocks) are available to describe relations.

3.2 Construction of the mixture intensity

To ensure that the graphon function (3) is continuous, we consider an integral-based weighting function of the form

$$F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)}) \propto \int_{L_{k_1-1}^{(1)}}^{L_{k_1}^{(1)}} f(x - u_i^{(1)})dx \cdot \int_{L_{k_2-1}^{(2)}}^{L_{k_2}^{(2)}} f(x - u_j^{(2)})dx \tag{4}$$

where $f(x-u)$ is a univariate derivative function. Beyond the continuity requirement, $f(x-u)$ and $F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)})$ should satisfy the following three conditions:

1. $f(x-u)$ is non-negative;
2. $f(x-u)$ increases with decreasing distance (i.e. $|x-u|$) between x and the corresponding coordinate u. This condition means that the closer the block $\square_{k_1,k_2}$ is to the pair of node coordinates $(u_i^{(1)}, u_j^{(2)})$, the larger weight the block will be assigned. The maximum weight value is achieved when $|x-u_i^{(1)}| = 0$ and $|x-u_j^{(2)}| = 0$;
3. The total weight of all blocks remains invariant regardless of different partitioning of the unit space. That is, $F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)}) = F_{\square'_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)}) + F_{\square''_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)})$, where $\square'_{k_1,k_2}, \square''_{k_1,k_2}$ are sub-boxes of $\square_{k_1,k_2}$ such that $\square_{k_1,k_2} = \square'_{k_1,k_2} \cup \square''_{k_1,k_2}$ and $\square'_{k_1,k_2} \cap \square''_{k_1,k_2} = \emptyset$.

It is also expected that $F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)})$ can be normalised over all the K^2 community pairs as $\sum_{k_1,k_2} F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)}) = 1$.

There are many candidate functions satisfying these conditions, such as Gaussian or Laplace probability density functions. For ease of computation and convenience of integration, we use the scaled Laplace density (with location parameter $\mu = 0$) as the derivative function $f_\lambda(x-u)$. We leave other function choices for future work. In particular, we define $f_\lambda(x-u) = \frac{\lambda}{2}\frac{e^{-\lambda|x-u|}}{G_\lambda(1-u)-G_\lambda(-u)}$, where $G_\lambda(x-u) = \begin{cases} \frac{1}{2}e^{\lambda(x-u)}; & (x-u) < 0 \\ 1-\frac{1}{2}e^{-\lambda(x-u)}; & (x-u) \geq 0 \end{cases}$. We then have $\int_{L_{k_1-1}}^{L_{k_1}} f_\lambda(x-u)dx = \frac{G_\lambda(L_{k_1}-u)-G_\lambda(L_{k_1-1}-u)}{G_\lambda(1-u)-G_\lambda(-u)}$. As a result, for relation R_{ij} and corresponding node coordinates $(u_i^{(1)}, u_j^{(2)})$, the normalised weight $F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)})$ of the (k_1, k_2)th block

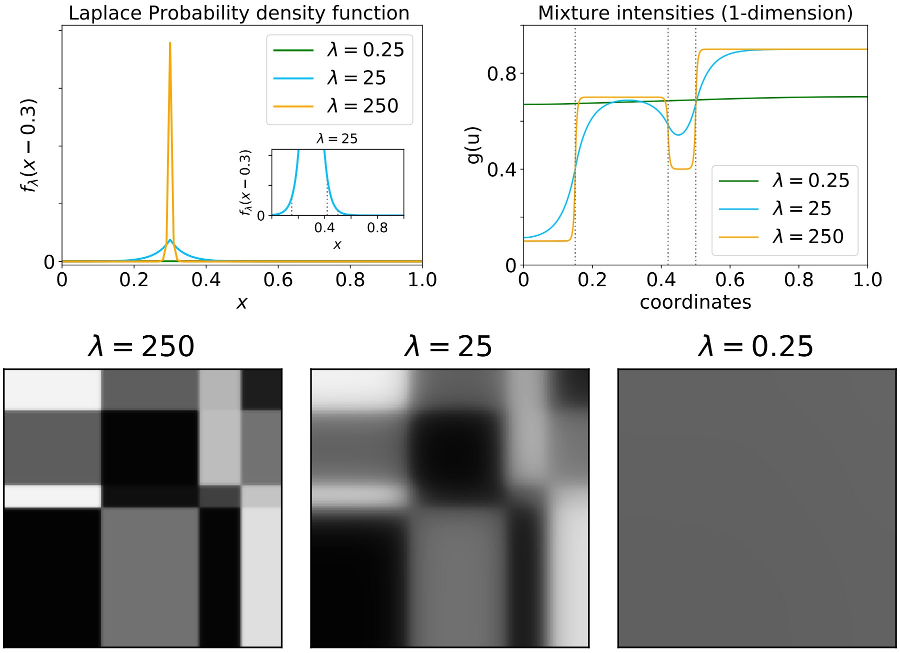

Fig. 2 Top row: the influence of the λ parameter on the Laplace probability density function with coordinate located at $u = 0.3$ (left), and the corresponding mixture intensities for $\{u_i\}_{i=1}^n$ (right). Different colors represent different settings of λ. The gray dotted lines represent segment division in one dimension with $\boldsymbol{\theta} = (0.15, 0.27, 0.08, 0.5)^\top \sim \text{Dirichlet}(1, 1, 1, 1)$. Bottom row: visualizations of the Integrated Smoothing Graphon under the Stochastic Block Model for different values of λ. Darker shading represents higher graphon intensity.

$\square_{k_1,k_2}$ contributing to the mixture intensity of R_{ij} is given by

$$F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)}) = \frac{G_\lambda(L_{k_1}^{(1)} - u_i^{(1)}) - G_\lambda(L_{k_1-1}^{(1)} - u_i^{(1)})}{G_\lambda(1 - u_i^{(1)}) - G_\lambda(-u_i^{(1)})} \times \frac{G_\lambda(L_{k_2}^{(2)} - u_j^{(2)}) - G_\lambda(L_{k_2-1}^{(2)} - u_j^{(2)})}{G_\lambda(1 - u_j^{(2)}) - G_\lambda(-u_j^{(2)})}. \tag{5}$$

Proposition 1 $\sum_{k_1,k_2} F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)}) = 1.$

Proof

$$\begin{aligned}
&\sum_{k_1,k_2} F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)}) \\
=&\left[\sum_{k_1} \frac{G_\lambda(L_{k_1}^{(1)} - u_i^{(1)}) - G_\lambda(L_{k_1-1}^{(1)} - u_i^{(1)})}{G_\lambda(1 - u_i^{(1)}) - G_\lambda(-u_i^{(1)})}\right] \\
&\cdot \left[\sum_{k_2} \frac{G_\lambda(L_{k_2}^{(2)} - u_j^{(2)}) - G_\lambda(L_{k_2-1}^{(2)} - u_j^{(2)})}{G_\lambda(1 - u_j^{(2)}) - G_\lambda(-u_j^{(2)})}\right] = 1.
\end{aligned} \tag{6}$$

Figure 2 (left) illustrates the function curves of $f_\lambda(x-u)$ for $u = 0.3$ and Figure 2 (right) shows the resulting one-dimensional mixture of intensities under varying scale parameter values $\lambda = 0.25, 25$ and 250. It is easily observed that, when λ is smaller, both the curves of the derivative function and the mixture intensity become flatter and smoother. Conversely, for larger λ, the mixture intensity values (generated for the coordinate 0.3) become more discrete. Bottom row of Figure 2 visualizes the mixture intensities obtained by applying the ISG to the SBM under the same three λ values.

Proposition 2 *λ controls the smoothness of the graphon, with $\lambda \to \infty$ recovering the piecewise-constant graphon, and $\lambda \to 0$ resulting in a globally constant graphon.*

Proof Using L'hopital's rule, when $\lambda \to 0$, we have

$$\begin{aligned}
&\lim_{\lambda\to 0}\frac{G_\lambda(L_{k_1}^{(1)}-u_i^{(1)})-G_\lambda(L_{k_1-1}^{(1)}-u_i^{(1)})}{G_\lambda(1-u_i^{(1)})-G_\lambda(-u_i^{(1)})}\\
=&\frac{L_{k_1}^{(1)}-u_i^{(1)}-(L_{k_1-1}^{(1)}-u_i^{(1)})}{1-u_i^{(1)}+u_i^{(1)}}=L_{k_1}^{(1)}-L_{k_1-1}^{(1)}. \qquad (7)
\end{aligned}$$

Thus, we get $F_{\square_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)}) = (L_{k_1}^{(1)} - L_{k_1-1}^{(1)})(L_{k_2}^{(2)} - L_{k_2-1}^{(2)})$, which is unrelated to the coordinate of $(u_i^{(1)}, u_j^{(2)})$. The graphon is a globally constants graphon.

We have three different cases when $\lambda \to \infty$: case (1), $L_{k_1}^{(1)} > L_{k_1-1}^{(1)} > u_i^{(1)}$, we have

$$\begin{aligned}
&\lim_{\lambda\to\infty}\frac{G_\lambda(L_{k_1}^{(1)}-u_i^{(1)})-G_\lambda(L_{k_1-1}^{(1)}-u_i^{(1)})}{G_\lambda(1-u_i^{(1)})-G_\lambda(-u_i^{(1)})}\\
=&\lim_{\lambda\to\infty}\frac{1-\frac{1}{2}e^{-\lambda(L_{k_1}^{(1)}-u_i^{(1)})}-(1-\frac{1}{2}e^{-\lambda(L_{k_1-1}^{(1)}-u_i^{(1)})})}{1-\frac{1}{2}e^{-\lambda(1-u_i^{(1)})}-\frac{1}{2}e^{-\lambda(u_i^{(1)})}}=0;
\end{aligned}$$

case (2), $L_{k_1-1}^{(1)} < L_{k_1-1}^{(1)} < u_i^{(1)}$, we have

$$\begin{aligned}
&\lim_{\lambda\to\infty}\frac{G_\lambda(L_{k_1}^{(1)}-u_i^{(1)})-G_\lambda(L_{k_1-1}^{(1)}-u_i^{(1)})}{G_\lambda(1-u_i^{(1)})-G_\lambda(-u_i^{(1)})}\\
=&\lim_{\lambda\to\infty}\frac{\frac{1}{2}e^{\lambda(L_{k_1}^{(1)}-u_i^{(1)})}-(\frac{1}{2}e^{\lambda(L_{k_1-1}^{(1)}-u_i^{(1)})})}{1-\frac{1}{2}e^{-\lambda(1-u_i^{(1)})}-\frac{1}{2}e^{-\lambda(u_i^{(1)})}}=0;
\end{aligned}$$

case (3), $L_{k_1-1}^{(1)} < u_i^{(1)} < L_{k_1-1}^{(1)}$, we have

$$\begin{aligned}
&\lim_{\lambda\to\infty}\frac{G_\lambda(L_{k_1}^{(1)}-u_i^{(1)})-G_\lambda(L_{k_1-1}^{(1)}-u_i^{(1)})}{G_\lambda(1-u_i^{(1)})-G_\lambda(-u_i^{(1)})}\\
=&\lim_{\lambda\to\infty}\frac{1-\frac{1}{2}e^{-\lambda(L_{k_1}^{(1)}-u_i^{(1)})}-(\frac{1}{2}e^{\lambda(L_{k_1-1}^{(1)}-u_i^{(1)})})}{1-\frac{1}{2}e^{-\lambda(1-u_i^{(1)})}-\frac{1}{2}e^{-\lambda(u_i^{(1)})}}=1.
\end{aligned}$$

That is, $F_{\Box_{k_1,k_2}}(u_i^{(1)}, u_j^{(2)}) = 1$ if and only if the coordinate $(u_i^{(1)}, u_j^{(2)})$ locates in the (k_1, k_2)th block. Thus, this smoothing graphon only becomes piecewise-constant when $\lambda \to \infty$.

Accordingly, we refer to λ as the smoothing parameter.

3.3 Latent Feature Smoothing Graphon (LFSG) with probabilistic assignment

While the ISG addresses the limitations of the SBM-graphon by generating continuous intensity values, its graphon function (3) indicates that all blocks are involved in calculating the mixture intensity for generating individual relations. Accordingly, the additive form for evaluating the mixture intensity makes it difficult to form efficient inference schemes for all random variables. To improve inferential efficiency we introduce auxiliary pairwise latent labels $\{s_{ij}\}_{j=1}^n$ (associated with node i) and $\{r_{ij}\}_{i=1}^n$ (associated with node j) for individual relations $\{R_{ij}\}_{i,j=1}^n$, where $s_{ij}, r_{ij} \in \{1, \ldots, K\}$ are the sender and receiver effects respectively. The $\{s_{ij}\}_{j=1}^n$ and $\{r_{ij}\}_{i=1}^n$ are sampled from the respective node categorical distributions in their corresponding dimensions using normalised weights as probabilities. In particular

$$\begin{aligned} \{s_{ij}\}_{j=1}^n &\sim \text{Categorical}(F_1^{(1)}(u_i^{(1)}), \ldots, F_K^{(1)}(u_i^{(1)})) \\ \{r_{ij}\}_{i=1}^n &\sim \text{Categorical}(F_1^{(2)}(u_j^{(2)}), \ldots, F_K^{(2)}(u_j^{(2)})), \end{aligned} \tag{8}$$

where $F_k(u) = \frac{G_\lambda(L_k-u)-G_\lambda(L_{k-1}-u)}{G_\lambda(1-u)-G_\lambda(-u)}$ is the normalised weight of segment k in the dimension of coordinate u. For each relation from node i to node j (R_{ij}), the hidden label s_{ij} denotes the group that node i belongs to (in the 1st dimension) and r_{ij} denotes the group that node j belongs to (in the 2nd dimension). Through the introduction of the two labels, the final intensity in determining R_{ij} can be obtained similarly to the Mixed Membership Stochastic Block Model (MMSB) [1]:

$$P(R_{ij} = 1 | s_{ij}, r_{ij}, \boldsymbol{B}) = B_{s_{ij}, r_{ij}}. \tag{9}$$

Note that since both $\{s_{ij}\}_{j=1}^n$ and $\{r_{ij}\}_{j=1}^n$ are n-element arrays, each node has the potential to belong to multiple segments, rather than the single segment under the SBM-graphon. When participating in different relations, each outgoing node i (incoming node j) can fall into different segments, which means that each node can play different roles when taking part in different relations. Note that assuming expectations over the hidden labels s_{ij} and r_{ij}, results in the same intensity as for the ISG, so that

$$\mathbb{E}_{s_{ij}, r_{ij}}\left[P(R_{ij} = 1 | s_{ij}, r_{ij}, \boldsymbol{B})\right] = g\left(u_i^{(1)}, u_j^{(2)}\right). \tag{10}$$

We term this approach the Latent Feature Smoothing Graphon (LFSG). Its generative process is described as follows:

1)∼3) The block intensities ($\boldsymbol{B}$), graphon partition ($\boxplus$) and 2-dimensional coordinates ($\{u_i^{(1)}, u_i^{(2)}\}_{i=1}^n$) are generated as for the SBM-graphon;

4. For $i = 1, \cdots, n$, calculate the hidden label distributions in each dimension, $\boldsymbol{F}^{(1)}(u_i^{(1)})$ and $\boldsymbol{F}^{(2)}(u_i^{(2)})$, where $\boldsymbol{F}^{(1)}(u_i^{(1)}) = (F_1^{(1)}(u_i^{(1)}), \ldots, F_K^{(1)}(u_i^{(1)}))$;
5. For $i, j = 1, \cdots, n$,
   (a) Generate the hidden labels $s_{ij} \sim \boldsymbol{F}^{(1)}(u_i^{(1)}), r_{ij} \sim \boldsymbol{F}^{(2)}(u_j^{(2)})$ following (8)
   (b) Generate $R_{ij} \sim \text{Bernoulli}\left(B_{s_{ij}, r_{ij}}\right)$.

Within the LFSG, λ can provide additional insight into the latent structures, in that it indicates the extent to which nodes belong to multiple communities. Larger (smaller) values of λ indicate that nodes are likely to belong to fewer (more) communities.

The number of communities for the LFSG models can be determined using similar strategies as for the SBM and ISG model.

Comparing the LFSG and the MMSB: The MMSB model is another notable Bayesian method for modelling exchangeable relational data. In contrast to other graphon methods, the MMSB model allows each node i to have a group distribution $\boldsymbol{F}_i$, which follows a Dirichlet distribution. To form the relation between any two nodes i, j, a latent label pair consisting of a sender and a receiver (s_{ij}, r_{ij}) is first generated via $s_{ij} \sim \text{Categorical}(\boldsymbol{F}_i)$, and $r_{ij} \sim \text{Categorical}(\boldsymbol{F}_j)$. The relation R_{ij} can then be generated based on the intensity of the block $\boldsymbol{B}$ formed by group s_{ij} and group r_{ij}: $R_{ij} \sim \text{Bernoulli}(B_{s_{ij}, r_{ij}})$. Our proposed LFSG model shares similarities with the MMSB model, since both of them use group distributions to represent individual nodes and the likelihood generation method is the same. However, there are key differences. These are: (1) The priors for the group distributions are different. In the MMSB model, the group distributions of all nodes are generated independently from a Dirichlet distribution, whereas in the LFSG model, the group distributions are highly dependent, since all are determined by the same partition structure and nodes coordinates (see Eq. 5); (2) The MMSB model requires nK parameters to form the group distributions, while the LFSG model requires only $2(n + K)$ parameters.

The LFSG model naturally fits within the graphon framework. The MMSB model can also be made to fit within the graphon framework in two ways. Firstly, by considering the group distributions $\boldsymbol{\pi}_i \in [0, 1]^D$ in the K-dimensional hypercube instead of the unit interval. Secondly, noting that the minimal condition on the function F under general graphon theory is that F is measurable, applying a transformation from $[0, 1]$ to $[0, 1]^K$ means that the MMSB model can also fit within the graphon framework on the unit square.

The graphical models for implementing the ISG within the SBM (referred to as the ISG-SBM), as well as for implementing the LFSG within the SBM (referred to as the LFSG-SBM) are illustrated in Figure 3. The main difference between the two models – the introduction of the pairwise hidden labels s_{ij} and r_{ij} for generating each relation R_{ij} – allows the LFSG-SBM to enjoy the following advantages over the ISG-SBM:

- The aggregated counting information of the hidden labels enables efficient Gibbs sampling of the block intensities $\boldsymbol{B}$. In the ISG (or SBM), the block intensity $\boldsymbol{B}$ is inferred through each node's latent label, while $\boldsymbol{B}$ is inferred through the senders and receivers for each relation in the LFSG (or MMSB). Since the numbers of senders and receivers are larger than the number of latent labels for nodes, i.e.

$N^2 > N$, the inference on $\boldsymbol{B}$ under the LFSG (or MMSB) is likely better than under the ISG (or SBM).

- Calculation involving $F_{\Box}$ is instead reduced to calculation involving $F_k(u)$, avoiding the inclusion of all blocks when calculating the mixture intensity.
- Because each node is allowed to belong to different groups when linking to other nodes, permitting differences in the natures of those links, the group distribution $F(u)$ is then easily interpretable as the group membership distribution for that node. For example, a higher membership degree in group k indicates the node is more important or active in group k.

Model identifiability: In a similar manner to the MMSB model, the LFSG model also mitigates issues of identifiability by relating the number of blocks to the low-rank property of the edge probability matrix. In this respect, the LFSG model can be regarded as a "restricted" version of the MMSB model (see above), in that while the group distributions in the LFSG are highly dependent, those in the MMSB model are independently generated. In our simulations we did not encounter any parameter non-identifiability. However, we note that the number of parameters in the LFSG model is smaller than that of the MMSB model (i.e. $2(n + K) < nK$ for $K \geq 3$ and a moderate value of n), and so the LFSG model may perform better than the MMSB model in overcoming issues of non-identifiability.

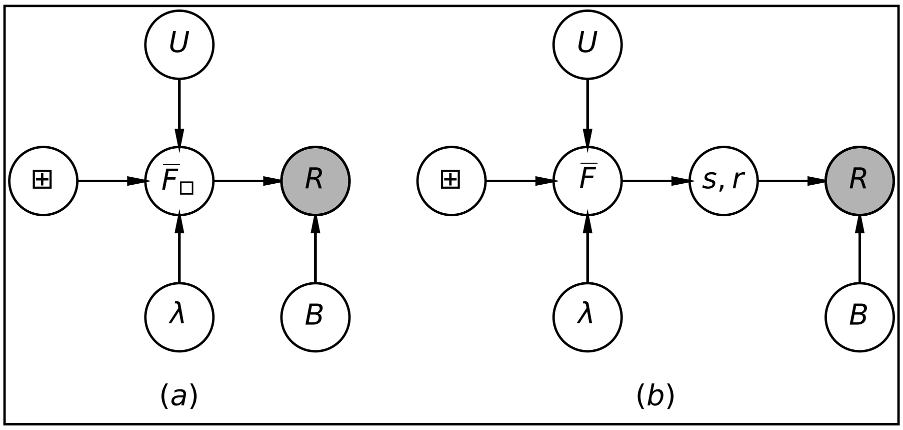

Fig. 3 The graphical model for (a) the ISG-SBM and (b) the LFSG-SBM. (a) The weights of the blocks $F_{\Box}$ are first calculated by using the partition $\boxplus$, node coordinates U and smoothing parameter λ. Then the $F_{\Box}$ and block intensities $\boldsymbol{B}$ are integrated together to generate the exchangeable relations R. (b) The weight F for each node is individually generated using the partition $\boxplus$, node coordinates U and the smoothing parameter λ, based on the auxiliary hidden labels s, r for the node pair relationship. Then s, r are used to generate the exchangeable relational, R, together with the block intensities $\boldsymbol{B}$.

3.4 Extensions to other piecewise-constant graphons

The major difference between the construction of the existing piecewise-constant graphons is the generation process of partitions ($\boxplus$; Figure 1). As a result, our smooth-

Algorithm 1 MCMC for the ISG

```
Input: Exchangeable relational data $R \in \{0,1\}^{n\times n}$, hyperparameters $\alpha_0, \beta_0, \boldsymbol{\alpha}_{1\times K}$, iteration time $T$
Output: $\{u_i^{(1)}, u_i^{(2)}\}_{i=1}^n, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \boldsymbol{B}, \lambda$
  for $t = 1, \cdots, T$ do
    for $i = 1, \ldots, n$ do
      Sample $u_i^{(1)}, u_i^{(2)}$; // according to (12)
    end for
    Sample $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}$; // according to (13)
    for $k_1, k_2 = 1, \ldots, K$ do
      Sample $B_{k_1,k_2}$; // according to (14)
    end for
    Sample $\lambda$; // according to (16)
  end for
```

ing approach, while described for the SBM-graphon, can be straight-forwardly applied to other piecewise-constant graphons. For example, to apply the ISG to other piecewise-constant graphons, we can similarly calculate a mixture intensity as a weighted sum of the intensities of all existing blocks. When the partitioned blocks are rectangular-shaped (as for e.g. the MP-RM, RTP-RM and RBP-RM), the intensity for each can be computed by independently integrating the derivative function over two dimensions. If the partitioned blocks are shaped as convex-polygons (as for e.g. the Binary Space Partitioning-Relational Model (BSP-RM) [14]), the intensity can be generated via integrating the derivative function over the polygon.

Nonparametric methods for Stochastic Block Models: In addition to the above Bayesian models, there are a number of nonparametric approaches for implementing stochastic block models. These approaches differ in terms of statistical accuracy and computational complexity, and include likelihood-based methods [3, 7, 9, 11, 44], moment-based methods [4], convex optimization methods [10], and spectral clustering methods [6, 16, 33, 37]. These approaches typically aim to produce consistent parameter point estimators, rather than full posterior distributions on the model parameters as considered here.

4 Inference

We present a Markov Chain Monte Carlo (MCMC) algorithm for posterior model inference, with detailed steps for the ISG and the LFSG models as illustrated in Algorithms 1 and 2 respectively. In general, the joint distribution over the hidden labels $\{s_{ij}, r_{ij}\}_{i,j=1}^n$, pairwise node coordinates $\{u_i^{(1)}, u_i^{(2)}\}_{i=1}^n$, group distributions

Algorithm 2 MCMC for the LFSG

```
Input: Exchangeable relational data $R \in \{0,1\}^{n\times n}$, hyperparameters $\alpha_0, \beta_0, \boldsymbol{\alpha}_{1\times K}$, iteration time $T$
Output: $\{u_i^{(1)}, u_i^{(2)}\}_{i=1}^n, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \boldsymbol{B}, \{s_{ij}, r_{ij}\}_{i,j=1}^n, \lambda$
  for $t = 1, \cdots, T$ do
    for $i = 1, \dots, n$ do
      Sample $u_i^{(1)}, u_i^{(2)}$; // according to (12)
    end for
    Sample $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}$; // according to (13)
    for $k_1, k_2 = 1, \dots, K$ do
      Sample $B_{k_1,k_2}$; // according to (14)
    end for
    for $i, j = 1, \dots, n$ do
      Sample $s_{ij}, r_{ij}$; // according to (15)
    end for
    Sample $\lambda$; // according to (16)
  end for
```

$\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}$, the block intensities $\boldsymbol{B}$ and the smoothing parameter λ is:

$$\begin{aligned}
&P(\{s_{ij}, r_{ij}, R_{ij}\}_{i,j=1}^n, \{u_i^{(1)}, u_i^{(2)}\}_{i=1}^n, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \boldsymbol{B}, \lambda | \alpha_0, \beta_0) \\
\propto &\prod_{i,k} \left[F_k^{(1)}(u_i^{(1)} | \boldsymbol{\theta}^{(1)}, \lambda)^{m_{ik}^{(1)}} F_k^{(2)}(u_i^{(2)} | \boldsymbol{\theta}^{(2)}, \lambda)^{m_{ik}^{(2)}} \right] \\
&\cdot \prod_{k_1,k_2} \left[B_{k_1,k_2}^{N_{k_1,k_2}^{(1)} + \alpha_0 - 1} (1 - B_{k_1,k_2})^{N_{k_1,k_2}^{(0)} + \beta_0 - 1} \right] \\
&\cdot \prod_k \left[(\theta_k^{(1)})^{\alpha_k - 1} (\theta_k^{(2)})^{\alpha_k - 1} \right] \cdot P(\lambda),
\end{aligned} \tag{11}$$

where $m_{ik}^{(1)} = \sum_{j=1}^n \mathbf{1}(s_{ij} = k), m_{ik}^{(2)} = \sum_{j=1}^n \mathbf{1}(r_{ji} = k), N_{k_1,k_2}^{(1)} = \sum_{(i,j):s_{ij}=k_1, r_{ij}=k_2} \mathbf{1}(R_{ij} = 1), N_{k_1,k_2}^{(0)} = \sum_{(i,j):s_{ij}=k_1, r_{ij}=k_2} \mathbf{1}(R_{ij} = 0)$. In this joint distribution, we have set the following prior distributions for the variables: $s_{ij} \sim \text{Categorical}(\boldsymbol{F}(u_i^{(1)} | \boldsymbol{\theta}^{(1)}, \lambda))$, $B_{k_1,k_2} \sim \text{Beta}(\alpha_0, \beta_0)$, $\boldsymbol{\theta}^{(1)} \sim \text{Dirichlet}(\boldsymbol{\alpha}_{1\times K})$. We let $\lambda \sim \text{Gamma}(0.1, 0.1)$ follow a vague Gamma distribution, where $\text{Gamma}(a, b)$ is a Gamma distribution with mean a/b and variance a/b^2

The details for updating each parameter in the ISG and LFSG MCMC algorithms are listed below.

Updating $\{u_i^{(1)}, u_i^{(2)}\}_{i=1}^n$*:* Independent Metropolis-Hastings steps can be used to update the variables $u_i^{(1)}, u_i^{(2)}$. We propose a new sample for $u_i^{(1)}$ from $u^* \sim \text{Beta}[\alpha_u, \beta_u]$, and accept this proposal with probability $\min(1, \alpha_{u_i^{(1)}})$ where

$$\alpha_{u_i^{(1)}} = \frac{\text{Be}(u_i^{(1)} | \alpha_u, \beta_u)}{\text{Be}(u^* | \alpha_u, \beta_u)} \prod_k \frac{F_k^{(1)}(u^* | \boldsymbol{\theta}^{(1)}, \lambda)^{m_{ik}^{(1)}}}{F_k^{(1)}(u_i^{(1)} | \boldsymbol{\theta}^{(1)}, \lambda)^{m_{ik}^{(1)}}}, \tag{12}$$

where $\text{Be}(u | \alpha, \beta)$ denotes the Beta density with parameters α and β evaluated at u. The update for $u_i^{(2)}$ proceeds likewise. Note that each of the $2n$ parameters $\{u_i^{(1)}, u_i^{(2)}\}_{i=1}^n$

can be updated in parallel. In our simulations we found that $\alpha_u = \beta_u = 1$ gave good sampler performance.

Updating $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}$*:* A random-walk Metropolis-Hastings step can be used to update $\boldsymbol{\theta}^{(1)}$, and $\boldsymbol{\theta}^{(2)}$. For $\boldsymbol{\theta}^{(1)}$ or $\boldsymbol{\theta}^{(2)}$ we draw a proposed sample $\boldsymbol{\theta}^* \sim \text{Dirichlet}(\boldsymbol{\alpha}_{1\times K})$ from a Dirichlet distribution with concentration parameters $\boldsymbol{\alpha}_{1\times K}$. We accept the proposal $\boldsymbol{\theta}^*$ for w.l.o.g. $\boldsymbol{\theta}^{(1)}$ with probability $\min(1, \alpha_{\boldsymbol{\theta}^{(1)}})$, where

$$\alpha_{\boldsymbol{\theta}^{(1)}} = \frac{\text{Diri}(\boldsymbol{\theta}^*|\boldsymbol{\alpha})}{\text{Diri}(\boldsymbol{\theta}^{(1)}|\boldsymbol{\alpha})} \cdot \prod_{i,k} \frac{F_k^{(1)}(u_i^{(1)}|\boldsymbol{\theta}^*, \lambda)^{m_{ik}^{(1)}}}{F_k^{(1)}(u_i^{(1)}|\boldsymbol{\theta}^{(1)}, \lambda)^{m_{ik}^{(1)}}}, \tag{13}$$

where $\text{Diri}(\boldsymbol{\theta}|\boldsymbol{\alpha})$ denotes the Dirichlet density with concentration parameter $\boldsymbol{\alpha}$ evaluated at $\boldsymbol{\theta}$. A similar update can be implemented for $\boldsymbol{\theta}^{(2)}$. Both $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ can be updated in parallel.

Updating $\boldsymbol{B}$*:* The conjugacy between the prior and the conditional likelihood for $\boldsymbol{B}$ means that we can update $\boldsymbol{B}$ via a Gibbs sampling step. Specifically, each entry B_{k_1,k_2} can be updated in parallel via

$$B_{k_1,k_2} \sim \text{Beta}(\alpha_0 + N_{k_1,k_2}^{(1)}, \beta_0 + N_{k_1,k_2}^{(0)}), \forall k_1, k_2. \tag{14}$$

Updating $\{s_{ij}, r_{ij}\}_{i,j=1}^n$*:* The posterior distribution of s_{ij} is a categorical distribution, where the probability of $s_{ij} = k$ is

$$P(s_{ij} = k|\theta_k^{(i)}, R_{ij}, B_{k,r_{ij}}) \propto F_k^{(1)}(u_i^{(1)}|\boldsymbol{\theta}^{(1)}, \lambda) \times B_{k,r_{ij}}^{R_{ij}}(1 - B_{k,r_{ij}})^{1-R_{ij}}, \tag{15}$$

and from which s_{ij} can be straightforwardly updated (r_{ij} can be updated in a similar way). Each of the $2n$ parameters can be updated in parallel.

Updating λ*:* A Metropolis-Hastings step can be used to update λ. We use the random walk Metropolis-Hastings algorithm to propose a new value of λ^* and accept it with probability $\min(1, \alpha_\lambda)$, where

$$\alpha_\lambda = \left[\prod_{i,k} \frac{F_k^{(1)}(u_i^{(1)}|\boldsymbol{\theta}^{(1)}, \lambda^*)^{m_{ik}^{(1)}} F_k^{(2)}(u_i^{(2)}|\boldsymbol{\theta}^{(2)}, \lambda^*)^{m_{ik}^{(2)}}}{F_k^{(1)}(u_i^{(1)}|\boldsymbol{\theta}^{(1)}, \lambda)^{m_{ik}^{(1)}} F_k^{(2)}(u_i^{(2)}|\boldsymbol{\theta}^{(2)}, \lambda)^{m_{ik}^{(2)}}}\right] \cdot \frac{[\lambda^*]^{-0.9} e^{-0.1\lambda^*}}{[\lambda]^{-0.9} e^{-0.1\lambda}}. \tag{16}$$

Table 1 Per-iteration model complexity comparison (n is the number of nodes, K is the number of communities and L is the number of positive links.)

Model	Intensity computation	Label sampling
SBM	$\mathcal{O}(K^2L)$	$\mathcal{O}(nK)$
ISG	$\mathcal{O}(K^2n^2)$	–
LFSG	$\mathcal{O}(K^2L)$	$\mathcal{O}(n^2K)$
MMSB	$\mathcal{O}(K^2L)$	$\mathcal{O}(n^2K)$
GP-RM	$\mathcal{O}(n^3)$	–

4.1 Computational complexities

Table 1 compares the per-iteration computational complexities of sampling from the ISG and the LFSG models against representative existing models, including the SBM, the MMSB and the GP-RM. The ISG algorithm requires a computational complexity of $\mathcal{O}(K)$ to sample each coordinate $u_i^{(1)}$ ($u_i^{(2)}$) (Eq.12), resulting in a total of $\mathcal{O}(NK)$ for all the coordinates; it needs a complexity of $\mathcal{O}(NK)$ to sample $\boldsymbol{\theta}^{(1)}$ ($\boldsymbol{\theta}^{(2)}$) (Eq.13); it needs a complexity of $\mathcal{O}(n^2K^2)$ to sample all the block intensities $\boldsymbol{B}$ (Eq.14) and a complexity of $\mathcal{O}(NK)$ to sample λ (Eq.16). In particular, it requires a computational complexity of $\mathcal{O}(n^2K^2)$ to calculate the intensity for generating the relations $\{R_{ij}\}_{i,j=1}^n$, since the calculation of the mixture intensity for each relation involves a pair of coordinates (giving a total of n^2) and all of the block intensities (which is K^2).

The computational complexity for the LFSG is similar to that of the ISG, as it needs a complexity of $\mathcal{O}(NK)$ to sample all coordinates, $\mathcal{O}(NK)$ to sample all partitions and $\mathcal{O}(NK)$ to sample λ. The LFSG has a complexity of $\mathcal{O}(n^2K)$ to sample all the latent labels (Eq.15) and K^2L to sample all the block intensities.

However, the uncoupling strategy applied in the LFSG lowers this cost dramatically to $\mathcal{O}(K^2L)$, where L is the number of positive links (i.e. $R_{ij} = 1$) observed in the data (Table 2 enumerates L for each data set analysed below). Note that the mixture intensity computation cost of the LFSG is the same as that of both the SBM and the MMSB. As a result, the continuous intensities of the LFSG compared to the discrete intensities of the SBM are achieved without sacrificing computation complexity. In contrast, the computational cost of computing the mixture intensity for the GP-RM is $\mathcal{O}(n^3)$ [32], which is the highest among these methods, even though it can also provide continuous intensities. Regarding the complexity of sampling the labels, both the LFSG and the MMSB provide multiple labels for each node and incur the same cost of $\mathcal{O}(n^2K)$. However, while the SBM requires a smaller cost of $\mathcal{O}(nK)$ for label sampling, it only allows a single label for each node.

5 Experiments

We now evaluate the performance of the ISG-SBM and the LFSG-SBM on real-world data sets, comparing them with four state-of-the-art methods: the SBM, the MP-RM, the MMSB and GP-RM. We implement posterior simulation for the SBM and the MMSB using Gibbs sampling and a conditional Sequential Monte Carlo algorithm

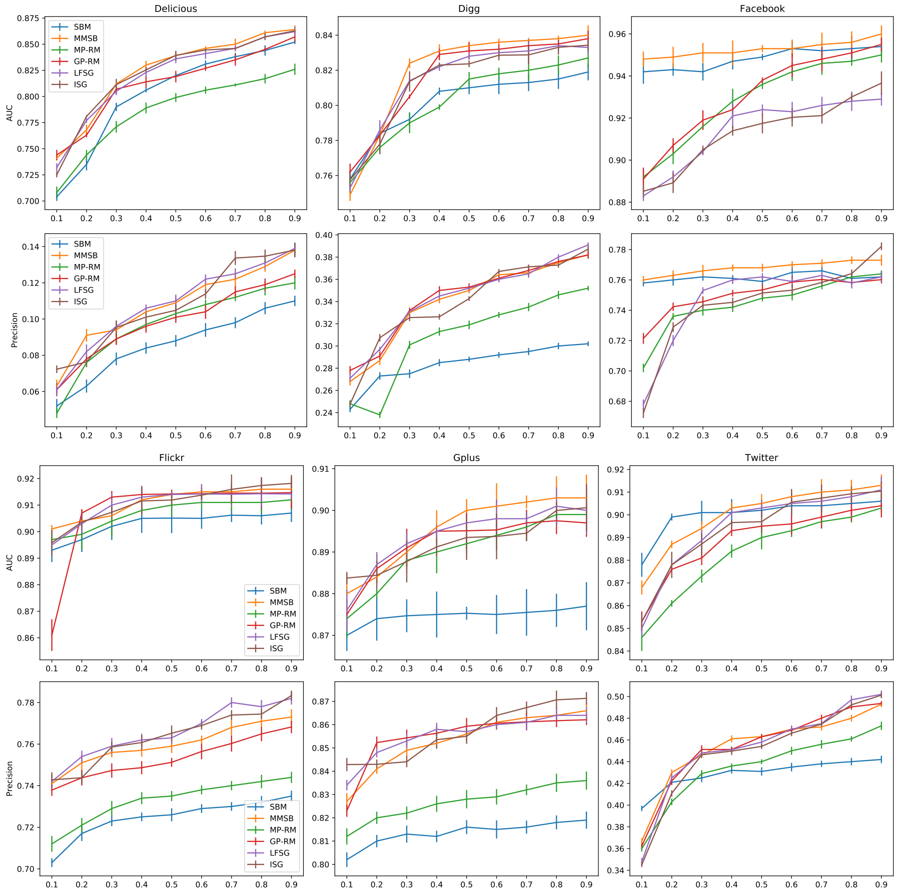

Fig. 4 Average area under the curve receiver operating characteristic (AUC) and the precision recall (Precision) under the Stochastic Block Model (SBM, blue line), Mixed-membership Stochastic Block Model (MMSB, orange line), Mondrian Process-Relational Model (MP-RM, green line), Gaussian Process-Relational Model (GP-RM, red line), Latent Feature Smoothing Graphon on the SBM (LFSG, purple line) and Integrated Smoothing Graphon on the SBM (ISG, brown line) for each of the Delicious, Digg, Facebook, Flickr, Gplus and Twitter datasets, under different proportions of training data (x-axis).

Table 2 Dataset summary information ($S(\%)$ is the sparsity of the positive links.)

Dataset	L	$S(\%)$	Dataset	L	$S(\%)$
Delicious	$10,775$	4.31	Gplus	$76,575$	30.63
Digg	$25,943$	10.38	Facebook	$54,476$	21.79
Flickr	$49,524$	19.81	Twitter	$24,378$	9.75

[5, 15, 20] for the MP-RM. We used 10 000 iterations for each sampling algorithm, retaining the final 5 000 iterations as post-burn-in draws from the posterior. Inspection of AUC and precision-value trace-plots indicated that 5 000 iterations were more than enough to ensure sampler convergence.

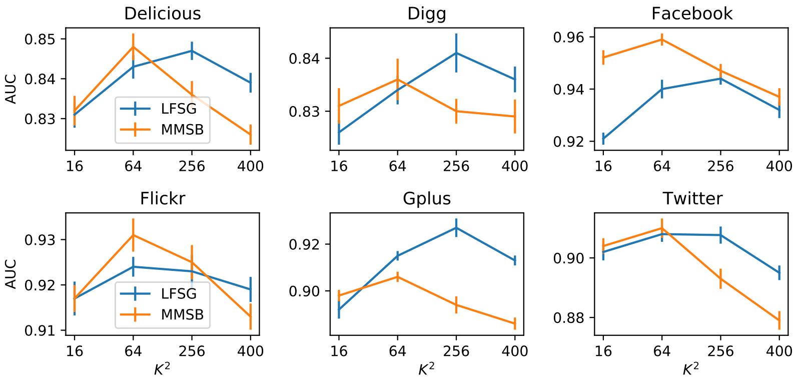

Fig. 5 AUC performance of the LFSG-SBM and the MMSB under different numbers (K^2) of blocks (in each dimension) for the Delicious, Digg, Facebook, Flickr, Gplus and Twitter datasets.

Table 3 Performance of neighborhood smoothing method of [43] with 90%/10% training/testing data, for each real world dataset.

Dataset	Delicious	Digg	Facebook	Flickr	Gplus	Twitter
AUC	0.853	0.833	0.941	0.912	0.893	0.907
Precision	0.129	0.373	0.752	0.769	0.863	0.473

5.1 Data sets

We examine six real-world exchangeable relational data sets: Delicious [40], Digg [40], Flickr [40], Gplus [21], Facebook [21], and Twitter [21]. To construct the exchangeable relational data matrix we extract the top 1 000 active nodes based on node interaction frequencies, and then randomly sample 500 nodes from these top 1 000 nodes to form the 500×500 interaction binary matrix. Table 2 summarizes the number of positive links (L) and the corresponding sparsity ($S\%$), which is defined as the ratio of the number of positive links to the total number of links, for each dataset.

5.2 Experimental setting

The hyper-parameters for each method are set as follows: for the SBM, LFSG-SBM, ISG-SBM, MMSB and MP-RM, the hyper-parameters α_0, β_0 used in generating the block intensities are set as $\alpha_0 = S\%, \beta_0 = 1 - S\%$, where $S\%$ refers to the sparsity shown in Table 2, such that the block intensity has an expectation equivalent to the sparsity of the exchangeable relational data; for the SBM, LFSG-SBM, ISG-SBM and MMSB, we set the group distribution of $\boldsymbol{\theta}$ as Dirichlet$(\mathbf{1}_{1\times 4})$. Hence, the number of groups in each dimension in these models is set as 4, with a total of 16 blocks generated in the unit square; for the MP-RM, the budget parameter is set to 3, which suggests that approximately $(3+1)\times(3+1)$ blocks would be generated. We use the generative processes of the corresponding models to initialize their random variables, as independent and random initialisation of these random variables would make the MCMC algorithm take a longer sequence of iterations to converge.

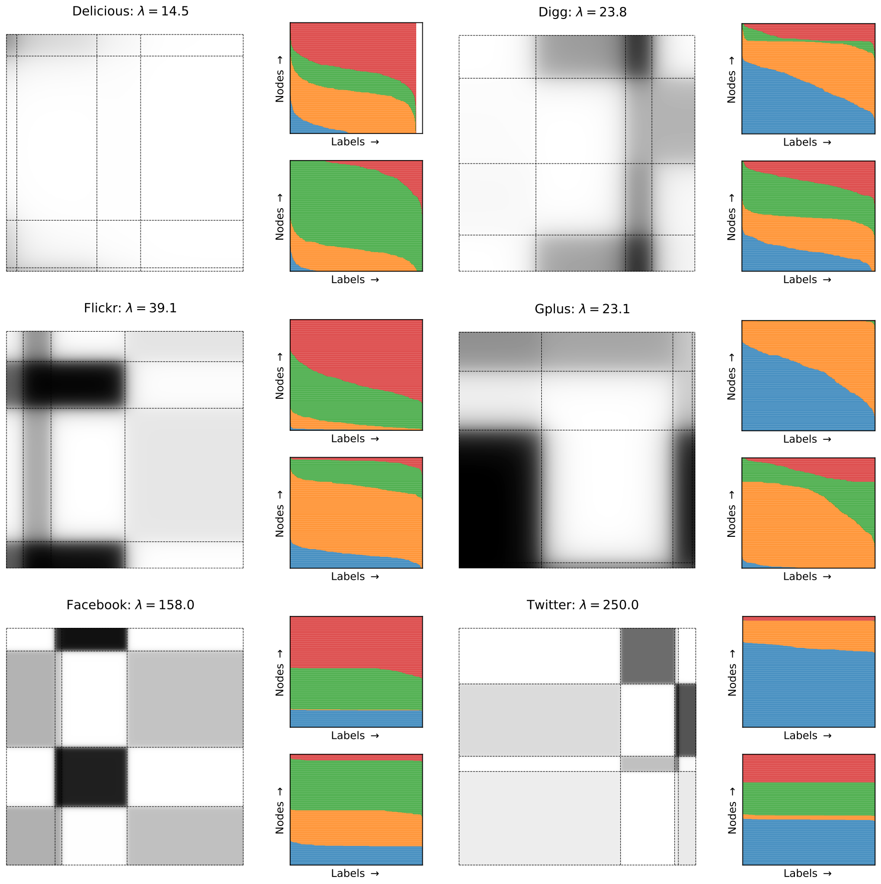

Fig. 6 Visualisation of mixture intensity from one posterior draw, the posterior mean of smoothing parameter λ and pairwise hidden labels on the Delicious, Digg, Flickr, Gplus, Facebook and Twitter datasets when implementing the Latent Feature Smoothing Graphon within the Stochastic Block Model. There are three figures for each dataset. Left: the grey level in the unit square illustrates the predicted mixture intensity for each relation (darker = higher intensity), the dotted lines indicate the related partition ($\boxplus = \boldsymbol{\theta}^{(1)} \times \boldsymbol{\theta}^{(2)}$). Right: the different colours represent the different values of the latent labels s_{ij} (right top) and r_{ij} (right bottom), with the y-axis indicating different nodes (sorted by the ratio of the labels) and the x-axis showing the proportion of different labels for each node.

5.3 Link prediction performance

The performance of each model in the task of link prediction is shown in Figure 4, which reports both the average area under the curve of the receiver operating characteristic (AUC) and the precision-recall (Precision). The AUC denotes the probability that the model will rank a randomly chosen positive link higher than a randomly chosen zero-valued link. The precision is the average ratio of correctly predicted positive links to the total number of predicted positive links. Higher values of AUC and precision indicate better model performance. For each dataset, we vary the ratio of training

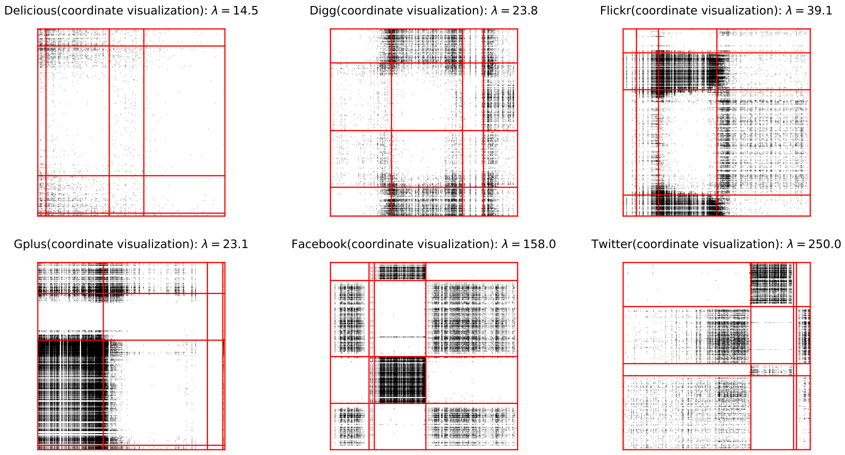

Fig. 7 Partition structure visualisations on the Delicious, Digg, Flickr, Gplus, Facebook and Twitter datasets when implementing the Latent Feature Smoothing Graphon within the Stochastic Block Model. Black dots refers to the observed linkages. We re-arrange the row and column indexes based on the nodes' coordinates and these visualisations reflect the observed relational data matrix after the index re-arrangements.

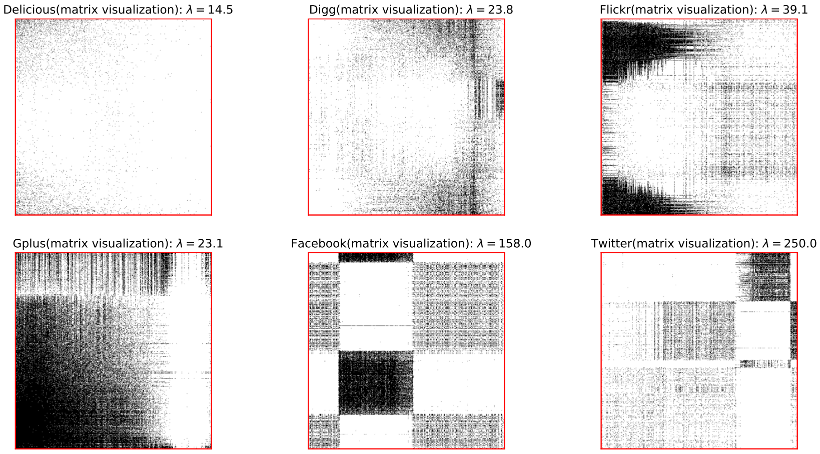

Fig. 8 The coordinates of all the nodes visualisations on the Delicious, Digg, Flickr, Gplus, Facebook and Twitter datasets when implementing the Latent Feature Smoothing Graphon within the Stochastic Block Model. Black dots refers to the observed linkages. We re-arrange the row and column indexes based on the nodes' coordinates and these visualisations reflect the observed relational data matrix after the index re-arrangements.

data from 10% to 90% and use the remainder for testing. The training/test data split is created in rows. In particular, we take the same ratio of training data from each row of the relational matrix, so that each node shares the same amount of training data. It is noted that the heterogeneity of the nodes' degrees might make this choice sample more 1-valued links from higher degree nodes and might thus produce a systematic bias in the results.

From Figure 4, both the AUC and precision of all models improves as the amount of training data increases. The trend generally becomes steady when the proportion is larger than 0.3, indicating the amount of data required to fit a model with a $\sim$16-block complexity.

Except for the Facebook data, we can see that the AUC and precision of both the ISG-SBM and the LFSG-SBM are better than for the piecewise-constant graphon models (i.e. the SBM and MP-RM) and the continuous graphon model (i.e. GP-RM). The proposed smoothing graphons typically achieve similar performance to the MMSB, demonstrating that the smoothing graphon strategy is useful for improving model performance. For the Facebook dataset, the SBM seems to perform better than the smoothing graphon-based models. This is examined in greater detail in the next section.

Figure 5 shows the relationships between the AUC performance and the number of blocks for the LFSG-SBM and the MMSB. In general, there are two stages for the behaviour of AUC values. Initially, the AUC values increase as the number of blocks becomes larger, possibly due to the enlarged model representation capability; then, the AUC values start to decline, even when the number of blocks continues to increase, possibly due to overfitting. We can see that the AUC values of the LFSG-SBM are usually better than those of the MMSB when the number of blocks is larger. That is, the LFSG-SBM is performs better than the MMSB on with respect to overfitting.

We additionally compare model performance with the neighborhood smoothing nonparametric method of [43], which uses a neighborhood smoothing to avoid the concept of blocks in the model. We implement this model using the R package `grahon`, using $90\%/10\%$ of the data as training/testing data. The resulting link prediction performance is shown in Table 3. When compared with the results in Figure 4, it is easy to see that the LFSG and ISG models have stronger performance.

5.4 Graphon and hidden label visualisation

In addition to the quantitative analysis, we visualise the generated graphons and hidden labels under the LFSG-SBM on all six data sets in Figure 6 and Figure 7. It is noted all these visualizations are based on single draws from the posterior distribution. In particular, these figures are created by re-ordering the nodes based on their estimated latent coordinates which depend on a particular posterior draw. For each dataset, we visualise the resulting mixture intensities for one posterior sample, with the learned posterior mean of the smoothing parameter λ, based on using 90% training data. We observe that the displayed graphon intensities exhibit smooth transitions between blocks for each dataset, highlighting that continuous, rather than discrete, mixture intensity values are generated under the smoothing graphon. The transition speed of the intensity between blocks is influenced by the smoothing parameter λ – a larger value of λ leads to a less smooth graphon, and a smaller value of λ to a smoother graphon – similar to that observed in Figure 2.

In Figure 6, for each dataset, we also display the posterior proportions of the pairwise hidden labels s_{ij} (top right) and r_{ij} (bottom right) for each node. Here the x-axis indicates different nodes (sorted by the label probabilities) and the y-axis displays the posterior mean of label probabilities (each label represented by a different colour). For each node i on the x-axis, more colours observed on the y-axis indicates a greater diversity of groups associated with that node, which in turn represents a higher potential for that node to belong to different groups when interacting with other nodes. In other words, the larger the tendency away from vertical line transitions between groups in these plots, the larger the number of nodes belonging to multiple groups.

Compared with the value of the smoothing parameter λ learned on the other four data sets, the values of λ estimated from the Facebook and Twitter datasets are larger. Further, the visualisations of the hidden labels for these two data sets are partitioned by almost straight horizontal lines, which suggests that only one label is a realistic possibility for most of the nodes. This could explain why both the AUC and precision values of the ISG-SBM and the LFSG-SBM are less competitive compared with those of the SBM on these two datasets (Figure 4). Here, the SBM assigns each node to exactly one group only, which aligns well with the ground-truth for these two datasets.

Another explanation for the performance on the Facebook and Twitter datasets is that we can recover the SBM if and only if $\lambda = \infty$. For any finite value of the smoothing parameter λ, it is impossible to have any posterior mass on the SBM. To this end, we might reparameterise by mapping λ to $(0, \infty) \to (0, 1)$ (e.g. via $\lambda \to 1 - e^{-\lambda}$) such that we are able to place substantial posterior mass close to 1. Since this mapping would permit model fitting arbitrarily close to the SMB, we would then expect the ISG-SBM and LFSG-SBM models to perform similar to or better than the SBM, even for the Facebook and Twitter datasets.

Figure 8 illustrates one sample partition drawn from their posterior distribution. As the black dots represent observed linkages, we can clearly see the pattern where they merge to form dense blocks. Furthermore, the dense regions in the partitions on the Facebook and Twitter datasets show clearer rectangular box shapes than those for the other datasets. This phenomenon is consistent with Figure 6, in which the smoothing parameter λ is larger for these two datasets. For other datasets, the dense regions are not confined to regular shapes, and accordingly may be better modelled by our method.

6 Conclusion

In this paper, we have introduced a smoothing strategy to modify conventional piecewise-constant graphons in order to increase their continuity. Through the introduction of a single smoothing parameter λ, we first developed the Integrated Smoothing Graphon (ISG) that addressed the key limitation of existing piecewise-constant graphons which only generate a limited number of discrete intensity values. To improve the computational efficiency of the ISG and to allow for the possibility of each node's belonging to multiple groups, we further developed the Latent Feature Smoothing

Graphon (LFSG) by the introduction of auxiliary hidden labels. Our experimental results verify the effectiveness of this smoothing strategy in terms of greatly improved AUC and precision scores in the task of link prediction. The visualisations of the generated graphons and the posterior hidden label summaries further provide an intuitive understanding of the nature of the smoothing mechanism for the given dataset.

Declarations

Funding - Yaqiong Li is a recipient of UTS Research Excellence Scholarship. Xuhui Fan and Scott A. Sisson are supported by the Australian Research Council (ARC) through the Australian Centre of Excellence in Mathematical and Statistical Frontiers (ACEMS, CE140100049), and Scott A. Sisson through the ARC Future Fellow Scheme (FT170100079). Ling Chen is supported by ARC Discovery Project DP180100966. Bin Li is supported in part by STCSM Project (20511100400), Shanghai Municipal Science and Technology Major Projects (2018SHZDZX01,2021SHZDZX0103), and the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

Conflicts of interest/Competing interests - The authors declare that they have no conflicts of interest/competing interests.

Ethics approval - Not applicable.

Consent to participate - Not applicable.

Consent for publication - Not applicable.

Availability of data and material - The data used in the experiments is available upon request to the corresponding author.

Code availability - The code written to get the experimental results is available upon request to the corresponding author.

Authors' contributions - Xuhui Fan initiated the main idea and model. Yaqiong Li formalized the detail inference methods, wrote the code, executed the experiments and developed the first draft. Ling Chen, Bin Li and Scott Sisson contributed to the writing and improvement of the paper.

References

1. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. In: NIPS, pp. 33–40 (2009)
2. Aldous, D.J.: Representations for partially exchangeable arrays of random variables. Journal of Multivariate Analysis **11**(4), 581–598 (1981)
3. Amini, A.A., Chen, A., Bickel, P.J., Levina, E.: Pseudo-likelihood methods for community detection in large sparse networks. The Annals of Statistics **41**(4), 2097–2122 (2013)
4. Anandkumar, A., Ge, R., Hsu, D., Kakade, S.: A tensor spectral approach to learning mixed membership community models. In: Proceedings of the 26th Annual Conference on Learning Theory, pp. 867–881 (2013)
5. Andrieu, C., Doucet, A., Holenstein, R.: Particle markov chain monte carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **72**(3), 269–342 (2010)
6. Balakrishnan, S., Xu, M., Krishnamurthy, A., Singh, A.: Noise thresholds for spectral clustering. In: Advances in Neural Information Processing Systems (2011)
7. Bickel, P.J., Chen, A.: A nonparametric view of network models and newman girvan and other modularities. Proceedings of the National Academy of Sciences **106**(50), 21068–21073 (2009)

8. Bouzas, D., Arvanitopoulos, N., Tefas, A.: Graph embedded nonparametric mutual information for supervised dimensionality reduction. IEEE Transactions on Neural Networks and Learning Systems **26**(5), 951–963 (2015)
9. Celisse, A., Daudin, J.J., Pierre, L.: Consistency of maximum-likelihood and variational estimators in the stochastic block model. Electronic Journal of Statistics **6**, 1847 – 1899 (2012)
10. Chen, Y., Sanghavi, S., Xu, H.: Clustering sparse graphs. In: Advances in Neural Information Processing Systems (2012)
11. CHOI, D.S., WOLFE, P.J., AIROLDI, E.M.: Stochastic blockmodels with a growing number of classes. Biometrika **99**(2), 273–284 (2012)
12. Dutta, A., Sahbi, H.: Stochastic graphlet embedding. IEEE Transactions on Neural Networks and Learning Systems **30**(8), 2369–2382 (2019)
13. Fan, X., Li, B., Sisson, S.: Rectangular bounding process. In: NeurIPS, pp. 7631–7641 (2018)
14. Fan, X., Li, B., Sisson, S.A.: The binary space partitioning-tree process. In: AISTATS, vol. 84, pp. 1859–1867 (2018)
15. Fan, X., Li, B., Sisson, S.A.: The binary space partitioning forests. In: AISTATS, vol. 89, pp. 3022–3031 (2019)
16. Fishkind, D., Sussman, D., Tang, M., Vogelstein, J., Priebe, C.: Consistent Adjacency-Spectral Partitioning for the Stochastic Block Model When the Model Parameters Are Unknown. SIAM Journal on Matrix Analysis and Applications **34**(1), 23 – 39 (2013)
17. Hoover, D.N.: Relations on probability spaces and arrays of random variables. Preprint, Institute for Advanced Study, School of Mathematics, Princeton, NJ (1979)
18. Ishiguro, K., Iwata, T., Ueda, N., Tenenbaum, J.B.: Dynamic infinite relational model for time-varying relational data analysis. In: NIPS, pp. 919–927 (2010)
19. Kemp, C., Tenenbaum, J.B., Griffiths, T.L., Yamada, T., Ueda, N.: Learning systems of concepts with an infinite relational model. In: AAAI, vol. 3, pp. 381–388 (2006)
20. Lakshminarayanan, B., Roy, D.M., Teh, Y.W.: Particle Gibbs for Bayesian additive regression trees. In: AISTATS, pp. 553–561 (2015)
21. Leskovec, J., Mcauley, J.J.: Learning to discover social circles in ego networks. In: Advances in neural information processing systems, pp. 539–547 (2012)
22. Li, B., Yang, Q., Xue, X.: Transfer learning for collaborative filtering via a rating-matrix generative model. In: ICML, pp. 617–624 (2009)
23. Lloyd, J., Orbanz, P., Ghahramani, Z., Roy, D.M.: Random function priors for exchangeable arrays with applications to graphs and relational data. In: NIPS, pp. 1007–1015 (2012)
24. Luo, X., Zhou, M., Li, S., You, Z., Xia, Y., Zhu, Q.: A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method. IEEE Transactions on Neural Networks and Learning Systems **27**(3), 579–592 (2016)
25. Nakano, M., Ishiguro, K., Kimura, A., Yamada, T., Ueda, N.: Rectangular tiling process. In: ICML, pp. 361–369 (2014)
26. Nikolaidis, K., Rodriguez-Martinez, E., Goulermas, J.Y., Wu, Q.H.: Spectral graph optimization for instance reduction. IEEE Transactions on Neural Networks and Learning Systems **23**(7), 1169–1175 (2012)
27. Nowicki, K., Snijders, T.A.: Estimation and prediction for stochastic block structures. Journal of the American Statistical Association **96**(455), 1077–1087 (2001)
28. Orbanz, P.: Construction of nonparametric Bayesian models from parametric Bayes equations. In: NIPS, pp. 1392–1400 (2009)
29. Orbanz, P., Roy, D.M.: Bayesian models of graphs, arrays and other exchangeable random structures. IEEE transactions on pattern analysis and machine intelligence **37**(2), 437–461 (2014)
30. Pang, Y., Ji, Z., Jing, P., Li, X.: Ranking graph embedding for learning to rerank. IEEE Transactions on Neural Networks and Learning Systems **24**(8), 1292–1303 (2013)
31. Pensky, M., et al.: Dynamic network models and graphon estimation. The Annals of Statistics **47**(4), 2378–2403 (2019)
32. Rasmussen, C.E.: Gaussian processes in machine learning. In: Summer School on Machine Learning, pp. 63–71. Springer (2003)
33. Rohe, K., Chatterjee, S., Yu, B.: Spectral clustering and the high-dimensional stochastic blockmodel. The Annals of Statistics **39**(4), 1878 – 1915 (2011)
34. Roy, D.M.: Computability, inference and modeling in probabilistic programming. Ph.D. thesis, MIT (2011)
35. Roy, D.M., Kemp, C., Mansinghka, V., Tenenbaum, J.B.: Learning annotated hierarchies from relational data. In: NIPS, pp. 1185–1192 (2007)

36. Roy, D.M., Teh, Y.W.: The Mondrian process. In: NIPS, pp. 1377–1384 (2009)
37. Sarkar, P., Bickel, P.J.: Role of normalization in spectral clustering for stochastic blockmodels. The Annals of Statistics **43**(3), 962 – 990 (2015)
38. Schmidt, M.N., Mørup, M.: Nonparametric Bayesian modeling of complex networks: An introduction. IEEE Signal Processing Magazine **30**(3), 110–128 (2013)
39. Wang, Q., Qin, Z., Nie, F., Li, X.: Spectral embedded adaptive neighbors clustering. IEEE Transactions on Neural Networks and Learning Systems **30**(4), 1265–1271 (2019)
40. Zafarani, R., Liu, H.: Social computing data repository at ASU (2009)
41. Zhang, Q., Lu, J., Wu, D., Zhang, G.: A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities. IEEE Transactions on Neural Networks and Learning Systems **30**(7), 1998–2012 (2019)
42. Zhang, X.: A nonconvex relaxation approach to low-rank tensor completion. IEEE Transactions on Neural Networks and Learning Systems **30**(6), 1659–1671 (2019)
43. Zhang, Y., Levina, E., Zhu, J.: Estimating network edge probabilities by neighbourhood smoothing. Biometrika **104**(4), 771–783 (2017)
44. Zhao, Y., Levina, E., Zhu, J.: Consistency of community detection in networks under degree-corrected stochastic block models. The Annals of Statistics **40**(4), 2266 – 2292 (2012)

Minor Revision Statement for "Smoothing Graphons for Modelling Exchangeable Relational Data"

MACH-D-20-00306R1

A Letter to the Action Editor,

Dear Action Editor Prof. Cussens,

We appreciate the time you and your reviewers have taken to arrange a second review of this paper. The valuable comments from the reviewer have helped significantly improve the quality of our work.

In this modified version, we have made thorough and substantial improvements, addressing the review comments carefully, as reflected in Section B below. Our revisions have concentrated on the following features:

- Adding references about the nonparametric algorithms for the stochastic block models (on Page 15)
- Discussing the identifiability issue in our smooth graphon model (on Page 14)
- Adding one nonparametric method comparison to the existing Bayesian approaches (Table 3)
- General tightening of the presentation and text.

More detailed explanations follow in the next section. For reading convenience, we have highlighted our main revision points in red in the manuscript. Please let us know if any further amendments are required.

Sincerely,

Yaqiong Li, Xuhui Fan, Ling Chen, Bin Li, and Scott A. Sisson

B Answers to comments from Reviewer 1

B.1 *Is the model identifiable? I can imagine that once you consider adding the smoothing on top of the block constant model, there may be more than one way to partition the blocks and fit the intensities such that the final distribution over the data is exactly the same under the two models? Given a smoothed model, it becomes a bit unclear whether the blocks still have the same meaning as before, as the averaged out model of edge probabilities could now be smooth across boundaries of blocks. In some ways that is the whole point of the smoothing process, and yet it generates new questions with respect to the algorithm and the interpretation of the model. One practical concern is that this unidentifiability could cause the posterior distribution to be very strangely shaped such that it would be hard to optimize over (as many different models would result in the same data distribution). In addition, there may not be a unique stationary distribution anymore, as the algorithm could converges to any mixture of two models producing the same data distribution. So I wonder how this would impact the behavior of the algorithm/output. In terms of the mixed membership SBM, this unidentifiability is one of the challenges for inference, and one suggested fix has been to impose a condition on the mixture probabilities. The number of blocks in the mixed membership SBM also has a meaningful interpretation as relating to the low rank property of the edge probability matrix.*

Answer: Thank you for this question. In a similar manner to the mixed-membership SBM, the LFSG model also mitigates issues of identifiability by relating the number of blocks to the low-rank property of the edge probability matrix. In this respect, the LFSG model can be regarded as a "restricted" version of the mixed-membership SBM (page 13), in that while the group distributions in the LFSG are highly dependent, those in the mixed-membership SBM are independently generated. In our simulations we did not encounter any non-identifiability. However, we note that the number of parameters in the LFSG is smaller than that of the mixed-membership SBM (i.e. 2(n+K)<nK for K>=3 and a moderate value of n), and so the LFSG model may perform better than the mixed-membership SBM in overcoming issues of non-identifiability.

We have included this discussion on model identifiability at the end of Section 3.3.

The blocks in the smooth graphon model have the same meaning as before. The smooth graphon model allows each point to have a probabilistic membership of each block -- this should not affect the meaning or definition of the blocks themselves.

B.2 *There are some nonparametric algorithms that have simple computational complexity and yet can extend beyond block models. In particular "Estimating network edge probabilities by neighborhood smoothing" by Yuan Zhang, Elizaveta Levina, Ji Zhu is one such example that assumes Lipschitzness of the underlying graphon. It would be helpful to compare and contrast your approach with their algorithm and to include this algorithm in the empirical comparisons as well. Another nonparametric property that has been considered is monotonicity, and there is also a computationally simple estimator as proposed in "A Consistent Histogram Estimator for Exchangeable Graph Models" by Stanley Chan and Edo Airoldi.*

Answer: Thank you for bringing these references and models to our attention. We have now implemented the Zhang et al. (2017) method in our simulations via the **est.ndbsmooth** function in the R package ***graphon*****,** and compared its empirical performance on our datasets. The detailed results are reported in Table 3 (page 19) and discussed at the end of Section 5.3. The developed smoothing graphon models outperform the Zhang et al. (2017) method, in this case.

We have also included a discussion of non-parametric methods on page 15, and emphasis our focus on Bayesian model implementations.

B.3 *I somewhat disagree with the statement that MMSBM cannot be modeled by graphon theory. I imagine that you are imagining graphons being limited to the domain of a unit square and smooth functions F? There are extensions of graphons to domains beyond the unit square, e.g. consider the latent variable u_i \in [0,1]^d being in the d-dimensional hypercube instead of the unit interval. Then the MMSBM would fit within the graphon framework easily. Furthermore, for general graphon theory, the minimal condition on the function F is simply that it is measurable, such that by applying a transformation that would map from [0,1] to [0,1]^d, the MMSBM would even be representable with the unit square domain.*

Answer: Thank you for sharing this interesting idea. We agree, have amended this statement, and have incorporated this explanation in the manuscript (page 13).

B.4 *Can you discuss the efficiency of the MCMC approach on this model? Did you observe it to take long to converge? Did you compare the Bayesian approach with spectral based algorithms that don't depend on a prior (at least for the SBM and MMSBM there are spectral algorithms in the literature). This discussion would be helpful in the computational complexity section, as the stated complexity in Table 1 should be multiplied by the time it takes to converge, as opposed to a spectral algorithm that would compute directly.*

Answer: We found that the MCMC sampler was fairly quick to converge. As stated in lines 47-49 on page 18, we used 10,000 iterations for each sampling algorithm, retaining the last 5,000 iterations as post-burn-in draws from the posterior. Inspection of AUC and precision-value trace-plots indicated that 5,000 iterations were enough to ensure convergence.

As our focus here was on Bayesian implementations, we did not consider other (e.g. non-Bayesian) algorithms. This also helped to ensure that the complexity calculations compared like with like on a per-iteration basis. (We now clarify that the complexities are per-iteration in Section 4.1.)

B.5 *After estimating the posterior distribution, how did you go from the posterior to predictions that were used to calculate the precision/recall curve? Was the curve resulting from setting different thresholds mapping form the probability of an edge in the model to the prediction of an edge? I imagine that you computed probability of an edge wrt the posterior distribution over the model? So that you didn't need to compute the maximum a posteriori model parameters which might have involved a complicated optimization.*

Answer: This is correct – we computed the probability of an edge under the model with respect to the posterior distribution. (We are unsure of the value of MAP estimators in this case given that a full posterior is available.)

That is, we used draws from the posterior to compute the expected AUC and precision values using the functions **metrics.roc_auc_score** and **metrics.average_precision_score** in Python's **scikit-learn** package (detailed values are obtained through setting different thresholds mapping from the probability of an edge to the prediction of an edge).

MLJ Contribution Information Sheet

Yaqiong Li, Xuhui Fan, Ling Chen, Bin Li, Scott A. Sisson

- **What is the main claim of the paper? Why is this an important contribution to the machine learning literature?**

  In machine learning, modelling exchangeable relational data can be described by graphon theory. Most Bayesian methods for modelling exchangeable relational data can be attributed to this framework by exploiting different forms of graphons. However, the graphons adopted by existing Bayesian methods are either piecewise-constant functions, which are insufficiently flexible for accurate modelling of the relational data, or are complicated continuous functions, which incur heavy computational costs for inference. In this work, we overcome these two shortcomings by smoothing piecewise-constant graphons, which permit continuous intensity values for describing relations, but without impractically increasing computational costs.

- **What is the evidence you provide to support your claim? Be precise.**

  In this work, we focus on the Bayesian Stochastic Block Model (SBM) and demonstrate how to adapt the piecewise-constant SBM graphon to the smoothed version. We initially propose the Integrated Smoothing Graphon (ISG) which introduces one smoothing parameter to the SBM graphon to generate continuous relational intensity values. We then develop the Latent Feature Smoothing Graphon (LFSG), which improves on the ISG by introducing auxiliary hidden labels to decompose the calculation of the ISG intensity and enable efficient inference. Experimental results on real-world data sets validate the advantages of applying smoothing strategies to the Stochastic Block Model, demonstrating that smoothing graphons can greatly improve AUC and precision for link prediction without increasing computational complexity.

- **What papers by other authors make the most closely related contributions, and how is your paper related to them?**

  This work is closely related to the following 5 papers.

  1. Airoldi,E.M.,Blei,D.M.,Fienberg,S.E.,Xing,E.P.:Mixedmembershipstochasticblock models.In: NIPS, pp. 33–40 (2009)

2. Fan,X.,Li,B.,Sisson,S.:Rectangularboundingprocess.In:NeurIPS,pp.7631–7641(2018)
3. Nowicki, K., Snijders, T.A.: Estimation and prediction for stochastic block structures. Journal of the American Statistical Association 96(455), 1077–1087 (2001)
4. Roy,D.M.,Teh,Y.W.:TheMondrianprocess.In:NIPS,pp.1377–1384(2009)
5. Lloyd, J., Orbanz, P., Ghahramani, Z., Roy, D.M.: Random function priors for exchangeable arrays with applications to graphs and relational data. In: NIPS, pp. 1007–1015 (2012)

When papers 1-4 proposes piecewise-constant graphons for modeling relational data and paper 5 uses Gaussian processes to model the Graphon functions, they are either accused of inflexibility modeling or high computational cost. Our work is a naturally combination between these two works and overcomes the above shortcomings.

- **Have you published parts of your paper before, for instance in a conference? If so, give details of your previous paper(s) and a precise statement detailing how your paper provides a significant contribution beyond the previous paper(s).**

  No, we have not published parts of this paper before.