

---

# A Consistent Histogram Estimator for Exchangeable Graph Models

---

Stanley H. Chan

School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA

SCHAN@SEAS.HARVARD.EDU

Edoardo M. Airolidi

Department of Statistics, Harvard University, Cambridge, MA 02138, USA

AIROLDI@FAS.HARVARD.EDU

## Abstract

Exchangeable graph models (ExGM) subsume a number of popular network models. The mathematical object that characterizes an ExGM is termed a *graphon*. Finding scalable estimators of graphons, provably consistent, remains an open issue. In this paper, we propose a histogram estimator of a graphon that is provably consistent and numerically efficient. The proposed estimator is based on a sorting-and-smoothing (SAS) algorithm, which first sorts the empirical degree of a graph, then smooths the sorted graph using total variation minimization. The consistency of the SAS algorithm is proved by leveraging sparsity concepts from compressed sensing.

## 1. Introduction

Developing statistical models for network data has been a growing research area in statistics and machine learning over the past decade (Goldenberg et al., 2009; Kolaczyk, 2009; Airolidi et al., 2011). Among many models, the *parametric* families have been the major focus in the literature because of their simplicity and analytic tractability. Popular examples of these parametric models include the exponential random graph model (Wasserman, 2005; Hunter & Handcock, 2006), the stochastic block-model (Nowicki & Snijders, 2001), the mixed membership model (Airolidi et al., 2008), the latent space model (Hoff et al., 2002), the graphlet (Azari & Airolidi, 2012) and many others. However, as the complexities of the networks increase, it becomes increasingly more challenging to fit the data using a particular parametric model.

### 1.1. Non-parametric representation of a graph

In this paper, we consider a non-parametric perspective of modeling network data using the exchangeable graph models (ExGM). The notion of exchangeability is due to de Finetti, later generalized by Aldous (1981), Hoover (1979) and Kallenberg (2005). A connection between popular parametric models and exchangeable graph models has been recently made (Hoff, 2008; Bickel & Chen, 2009).

The *non-parametric* (limit) object that characterizes an ExGM is often termed a *graphon*. As we will define formally in Section 2, a graphon is a 2-dimensional continuous function on  $[0, 1]^2 \rightarrow [0, 1]$  that generates random graphs. Since a graphon is a model for network data, any model based inference, prediction and hypothesis testing can be performed using a graphon (Lloyd et al., 2012). For instance, when comparing networks that span different sample sizes, graphons provide a natural solution: If two samples of a network are generated from the same ExGM, they should have the same graphon, and hence, comparing two networks can be done by comparing two graphons.

In this paper, we propose an efficient graphon estimator based on 2D histograms. The challenge of the problem is two-fold. First, since graphons are unique up to measure-preserving transformations, it is important to identify the conditions under which graphons can be uniquely recovered (e.g., see Yang et al., 2014). Second, it is desirable for a graphon estimator to be provably consistent.

### 1.2. Related work

Previous methods of graphon estimation algorithms can be classified into two categories as follows.

The first category is to perform graphon estimation conditioned on the node arrangement. When the node arrangement is conditioned on, we can bypass the difficult problem of identifying a canonical representation of the graphon. For example, the universal singular value thresholding (Chatterjee) and the matrix completion (Keshavan et al., 2010) seek low-rank structures of the

adjacency matrix, whereas the stochastic blockmodel approximation (Airoldi et al., 2013; Chan et al., 2013) groups similar nodes to form community structures. However, since the estimations are conditioned on the node arrangement, the resulting graphons are not canonical.

Different from the first category, the second category of methods estimate *canonical* graphons. In (Bickel et al., 2011), the authors proposed a method of moments which is theoretically consistent for that purpose. However, the method requires knowledge of *all* wheels of the network, and hence is computationally infeasible. Choi et al. (Choi et al., 2012; Choi & Wolfe) attempted the problem by a clustering approach, but they stopped at the clustering step without actually estimating the graphon. In (Lloyd et al., 2012), Lloyd et al. considered a Bayesian approach to estimate a graphon. However, the MCMC sampling process of the algorithm is computationally intensive. Moreover, there is no consistency guarantee of the estimator. More recently, other groups have begun exploring alternative approaches (Wolfe & Olhede; Tang et al., 2013; Latouche & Robin, 2013; Olhede & Wolfe). Yet, none of these methods are *both* consistent and computationally efficient.

### 1.3. Contributions

In this paper, we propose a histogram approach to estimate graphons. Our method, called the Sorting-And-Smoothing (SAS) algorithm, consists of two steps. In the first step, we sort the empirical degrees and rearrange the nodes of the graph for a canonical ordering. In the second step, we compute the histogram of the sorted graph and smooth the histogram by a total variation minimization. Details of the SAS algorithm are presented in Section 3.

The estimator returned by the SAS algorithm is consistent. The consistency proof leverages the sparsity concepts from compressed sensing. In particular, we show, in Theorem 3, that if the true graphon satisfies some Lipschitz conditions and has sparse gradients, then the mean squared error (MSE) of the estimator is  $\mathcal{O}((\log n)/n)$ , where  $n$  is the size of the network. Discussion of the consistency is presented in Section 4.

We test the SAS algorithm on both simulation data and real data (Section 5). The experiment of using the simulation data indicates that the SAS algorithm is superior to, both in terms of estimation quality and speed, several existing methods. Applying the SAS algorithm to real data, we estimate graphons of two large-scale social networks and reveal some structures. These results provide an alternative way of analyzing large-scale network data.

For clarity of the presentation, details of the proofs can be found in a follow up report (Chan & Airoldi, 2014).

## 2. Graphons and identifiability

The purpose of this section is to introduce the concepts of a graphon and discuss the conditions under which a graphon can be uniquely identified.

### 2.1. Definition of a graphon

We let  $G$  be the adjacency matrix of a graph with the  $(i, j)$ th entry denoted by  $G_{ij} \in \{0, 1\}$ . For an infinitely sized graph  $G$ , we say that  $G$  is exchangeable if it satisfies the following definition.

**Definition 1.** An infinite random array  $G = (G_{ij})_{i,j \in \mathbb{N}}$  is exchangeable if

$$(G_{ij}) \stackrel{d}{=} (G_{\sigma(i)\sigma(j)}), \quad (1)$$

for any permutation  $\sigma$ .

Definition 1 is also known as the *joint* exchangeability, because the permutation is applied to both rows and columns simultaneously (Orbanz & Roy).

We refer to all random graph models that satisfy exchangeability as exchangeable graph models (ExGM). A useful characterization of an ExGM is given by the Aldous-Hoover theorem.

**Theorem 1** (Aldous-Hoover). An infinite random array  $(G_{ij})_{i,j \in \mathbb{N}}$  is exchangeable if and only if there is a random measurable function  $F : [0, 1]^3 \rightarrow \{0, 1\}$  such that

$$(G_{ij}) \stackrel{d}{=} (F(U_i, U_j, U_{ij})), \quad (2)$$

where  $(U_i)_{i \in \mathbb{N}}$  and  $(U_{ij})_{i,j \in \mathbb{N}}$  are sequences of i.i.d. Uniform $[0, 1]$  random variables.

The function  $F$  in Theorem 1 defines a *graphon*:

**Definition 2** (Graphon). A graphon  $w$  is a symmetric measurable function  $w : [0, 1]^2 \rightarrow [0, 1]$  such that

$$F(U_i, U_j, U_{ij}) = \begin{cases} 1, & \text{if } U_{ij} < w(U_i, U_j) \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where  $(U_i)_{i \in \mathbb{N}}$  and  $(U_{ij})_{i,j \in \mathbb{N}}$  are sequences of i.i.d. Uniform $[0, 1]$  random variables.

Equivalently, (3) can be expressed as the following two-stage sampling scheme:

$$\begin{aligned} U_i & \stackrel{iid}{\sim} \text{Uniform}[0, 1], \\ G_{ij} \mid U_i, U_j & \sim \text{Bernoulli}(w(U_i, U_j)). \end{aligned} \quad (4)$$

Therefore, a finite sized network generated from a graphon can be regarded as a finite sample drawn according to (4).

## 2.2. Identifiability of a graphon

To understand the identifiability issue of a graphon, it is important to discuss measure preserving transformations.

**Definition 3** (Measure Preserving Transformation). A transformation  $\varphi : [0, 1] \rightarrow [0, 1]$  is *measure-preserving w.r.t. a measure  $\mu$*  if it is measurable, and for all  $A \in [0, 1]$ ,

$$\mu(\varphi^{-1}(A)) = \mu(A). \quad (5)$$

For example, if  $\varphi$  is a measure preserving transformation and  $U \sim \text{Uniform}[0, 1]$ , then  $\varphi(U)$  is also distributed uniformly on  $[0, 1]$ . Similarly, if  $\varphi$  is a measure preserving transformation, then the graphon

$$w'(u, v) \stackrel{\text{def}}{=} w(\varphi(u), \varphi(v))$$

defines the same ExGM as  $w$  because there exists a transformation such that  $w$  and  $w'$  are identical.

The identifiability issue of a graphon arises because the converse of Definition 3 is not true in general: If  $w$  and  $w'$  define the same ExGM, there may not exist a measure preserving transformation  $\varphi'$  such that  $w(u, v) = w'(\varphi'(u), \varphi'(v))$  (Diaconis & Janson, 2008). For example, the functions  $w(u, v) = uv$  and  $w'(u, v) = (2u \bmod 1)(2v \bmod 1)$  define the same ExGM, but there is no  $\varphi'$  such that  $w(u, v) = w'(\varphi'(u), \varphi'(v))$ .

A formal statement of the above observation is given by the following theorem, which says that we need to find a pair of measure-preserving transformations  $\varphi$  and  $\varphi'$  in order to show that  $w$  is unique.

**Theorem 2** ((Diaconis & Janson, 2008), Thm. 7.1). Let  $w$  and  $w'$  be two graphons. Then  $\delta_{\square}(w, w') = 0$  if and only if there exist measure-preserving transformations  $\varphi$  and  $\varphi' : [0, 1] \rightarrow [0, 1]$  such that

$$w(\varphi(u), \varphi(v)) = w'(\varphi'(u), \varphi'(v)), \quad (6)$$

where the distance  $\delta_{\square}(w, w')$  is the cut-norm defined by (Lovász & Szegedy, 2006).

A consequence of Theorem 2 is the notion of twin-free:

**Definition 4** (Twin-free (Borgs et al., 2010)). A graphon  $w$  is called *twin-free* if for any  $u_1$  and  $u_2 \in [0, 1]$ ,  $w(u_1, v) \neq w(u_2, v)$  for almost all  $v \in [0, 1]$ .

Essentially, the twin-free condition excludes the cases where two graphons can be made identical by row and column permutations. For example, the pair shown in Figure 1 are twin, and hence they are not identifiable.

The twin-free condition is necessary but not sufficient for identifying a unique graphon when we marginalize a graphon (Orbanz & Roy):

$$g(u) \stackrel{\text{def}}{=} \int_0^1 w(u, v) dv.$$

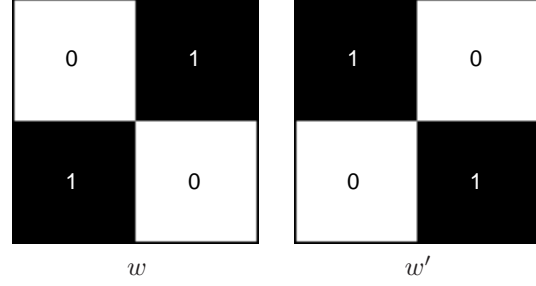


Figure 1. Example of a pair of twin graphons:  $w$  and  $w'$  are not identifiable if we randomly permute their columns and rows.

For example, if we consider  $w$  and  $w'$  in Figure 1, and a graphon  $w''(u, v) = 1/2$ , then  $w''$  is twin-free but  $g(u) = g'(u) = g''(u)$ , where  $g, g'$  and  $g''$  are marginalizations of  $w, w'$  and  $w''$ , respectively.

The necessary and sufficient condition for a graphon to be identifiable is to require strict monotonicity of degrees (Bickel & Chen, 2009; Yang et al., 2014).

**Condition 1** (Strict Monotonicity of Degree). A graphon  $w$  has a unique representation if and only if there exists  $w^{can}$  such that

$$g^{can}(u) \stackrel{\text{def}}{=} \int_0^1 w^{can}(u, v) dv$$

is strictly increasing (or decreasing). The graphon  $w^{can}$  is called the canonical representation of  $w$ .

It is evident that the strict monotonicity condition implies twin-free, but not vice versa. In addition, if we let  $U \sim \text{Uniform}[0, 1]$ , then strict monotonicity implies that  $g^{can}(U)$  is absolutely continuous.

In the rest of the paper we assume that all graphons of interests satisfy the strict monotonicity condition. For notational simplicity, we drop the superscript  $(\cdot)^{can}$ , and denote  $w$  as the canonical representation.

## 3. The sorting-and-smoothing algorithm

### 3.1. Overview

The intuition of the proposed SAS algorithm is based on the following idea: As the size of a graph grows, the (sorted) empirical degree should converge to the ideal (canonical) degree distribution. Therefore, if we can sort the empirical degree of a given graph, then by applying suitable smoothing algorithms we can find an estimate of the canonical graphon.

Following this intuition, we propose a two-stage algorithm. In the first stage, we sort the rows and columns of  $G$  to obtain a sorted graph  $\hat{A}$  according to the empirical degree.

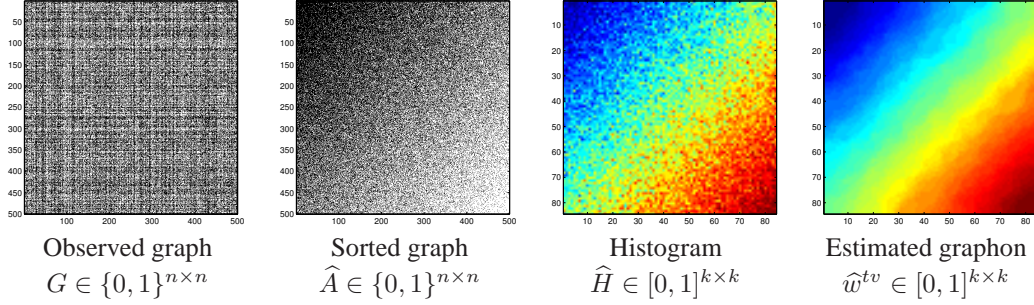


Figure 2. Illustration of the SAS algorithm. Given an observed graph  $G$ , we first sort  $G$  using the empirical degrees to get  $\hat{A}$ . Then, a local histogram  $\hat{H}$  is computed and a total variation minimization is used to determine an estimate  $\hat{w}^{tv}$ .

In the second stage, we compute a histogram  $\hat{H}$  of  $\hat{A}$ , and apply a total variation minimization to find an estimate  $\hat{w}^{tv}$ . An illustration of the SAS algorithm is shown in Figure 2.

### 3.2. Stage 1: Sorting

The purpose of the sorting step is to rearrange the observed graph  $G$  so that the rearranged empirical degrees are monotonically increasing. To this end, we compute the empirical degree

$$d_i \stackrel{\text{def}}{=} \sum_{j=1}^n G_{ij}, \quad (7)$$

and define a permutation  $\hat{\sigma}$  such that  $d_{\hat{\sigma}(1)} < \dots < d_{\hat{\sigma}(n)}$ . Then, we define a rearranged graph

$$\hat{A}_{ij} = G_{\hat{\sigma}(i)\hat{\sigma}(j)}, \quad (8)$$

where an example is shown in Figure 2.

It is important to note that since the permutation  $\hat{\sigma}$  is defined by the empirical degrees, it could be different from the true permutation that defines the canonical graphon according to the node arrangement. To differentiate the empirical permutation  $\hat{\sigma}$  and the true permutation, we define  $\sigma$  as the oracle permutation that sorts the node labels ( $U_i$ ) such that  $U_{\sigma(1)} < \dots < U_{\sigma(n)}$ . Correspondingly, we define the oracle ordered graph as

$$A_{ij} = G_{\sigma(i)\sigma(j)}. \quad (9)$$

### 3.3. Stage 2: Smoothing

#### Network Histogram Estimation

Once the graph is rearranged to have monotonically increasing degrees, the graphon estimation problem becomes finding a smooth surface that best fits  $(A_{ij})$ . To this end, we consider a simplified version of the stochastic block-model approximation (Airolidi et al., 2013) which approximates the continuous graphon using a piecewise constant function. More precisely, the stochastic blockmodel ap-

proximation defines

$$\hat{H}_{ij} = \frac{1}{h^2} \sum_{i_1=1}^h \sum_{j_1=1}^h \hat{A}_{ih+i_1, jh+j_1}, \quad (10)$$

and correspondingly

$$H_{ij} = \frac{1}{h^2} \sum_{i_1=1}^h \sum_{j_1=1}^h A_{ih+i_1, jh+j_1}, \quad (11)$$

for some parameter  $h > 0$  denoting the size of each block.

Equations (10) and (11) indicate that the stochastic block-model approximations  $(\hat{H}_{ij})$  and  $(H_{ij})$  are the histograms of  $(\hat{A}_{ij})$  and  $(A_{ij})$ , respectively. Since all function values in the same block are identical, the effective degrees of freedom in  $(\hat{H}_{ij})$  and  $(H_{ij})$  are  $k \times k$  instead of  $n \times n$ , where  $k = \lfloor n/h \rfloor$  is the number of blocks.

#### Total Variation Minimization

While the network histogram estimation step is consistent, the decay rate of the error can be further improved by introducing a total variation minimization step.

The total variation minimization step is based on a sparsity assumption of the true graphon  $w$ . Analogous to natural images, we assume that graphons are *sparse* in the gradients. Discretizing the continuous graphon  $w$  into a  $k \times k$  grid, the assumption suggests that  $w$  needs to have a small total variation

$$\|w\|_{TV} = \sum_{i=1}^k \sum_{j=1}^k \sqrt{\left(\frac{\partial w}{\partial x}\right)_{ij}^2 + \left(\frac{\partial w}{\partial y}\right)_{ij}^2}, \quad (12)$$

where  $\frac{\partial w}{\partial x}$  and  $\frac{\partial w}{\partial y}$  denote the horizontal and vertical finite difference of  $w$ , respectively.

Using the total variation concept, the refinement step can be posed as the following minimization problem:

$$\hat{w}^{tv} = \underset{\hat{r}}{\operatorname{argmin}} \|\hat{r}\|_{TV}, \quad \text{subject to } \|\hat{r} - \hat{H}\|_2 \leq \varepsilon, \quad (13)$$



where  $\|\cdot\|_2$  is the matrix Frobenius norm, and  $\varepsilon > 0$  is a parameter that controls the fidelity between the total variation solution  $\hat{r}$  and the histogram  $\hat{H}$ . To solve the minimization problem (13), we use the alternating direction method of multipliers (ADMM) (Chan et al., 2011).

We remark that the size of  $\hat{w}^{tv}$  is  $k \times k$ . To ensure that the final estimate have the same size as the true graphon, we define the final estimate as

$$\hat{w}^{est} = \hat{w}^{tv} \otimes \mathbf{1}_{h \times h}, \quad (14)$$

where  $\mathbf{1}_{h \times h}$  denotes an all 1 matrix of size  $h \times h$ , and  $\otimes$  denotes the Kronecker product operator. Therefore, the final estimate  $\hat{w}^{est}$  has a size  $n \times n$ .

### 3.4. Complexity

The complexity of the SAS algorithm can be analyzed by considering each step individually. In computing the empirical degree distribution,  $\mathcal{O}(n)$  additions are used. The sorting procedure, in general, requires  $\mathcal{O}(n \log n)$  comparisons. Therefore, the complexity for sorting is about  $\mathcal{O}(n \log n)$  multiplications plus  $\mathcal{O}(n)$  additions. Next, for the histogram computation, computing each value of the bin requires  $\mathcal{O}(h^2)$  additions, and there are  $k^2 = (n/h)^2$  bins. Thus a total of  $\mathcal{O}(n^2)$  additions are needed. Finally, the total variation minimization is solved on a  $k \times k$  array. Thus, the complexity of the ADMM step is  $\mathcal{O}(k^2 \log k^2)$ . (See (Chan et al., 2011) for discussions.) Combing these results we can show that the overall complexity of the SAS algorithm is  $\mathcal{O}(n \log n + k^2 \log k^2)$  multiplications plus  $\mathcal{O}(n^2)$  additions.

## 4. Consistency

In this section we discuss the statistical consistency of the proposed SAS algorithm.

Analyzing the consistency of the SAS algorithm is equivalent to determining an upper bound of the error

$$\begin{aligned} \text{MSE} &\stackrel{\text{def}}{=} \frac{1}{n^2} \mathbb{E} [\|\hat{w}^{est} - w\|_2^2] \\ &= \frac{1}{n^2} \left( \mathbb{E} [h^2 \|\hat{w}^{tv} - H^w\|_2^2] + \mathbb{E} [\|H^w \otimes \mathbf{1}_{h \times h} - w\|_2^2] \right. \\ &\quad \left. + 2 \mathbb{E} [(\hat{w}^{tv} - H^w)^T (H^w \otimes \mathbf{1}_{h \times h} - w)] \right), \end{aligned} \quad (15)$$

where  $H^w$  is the histogram approximation of  $w$ :

$$H_{ij}^w = \frac{1}{h^2} \sum_{i_1=1}^h \sum_{j_1=1}^h w_{ih+i_1, jh+j_1}. \quad (16)$$

Before we proceed, we note that the second expectation in (15) is a classical result of approximating a continuous function by step functions. The bound is given in the following Lemma.

**Lemma 1** (Piecewise Constant Function Approximation). *Let  $w \in [0, 1]^{n \times n}$  be the true graphon and let  $H^w \in [0, 1]^{k \times k}$  be the histogram approximation defined in (16). Then,*

$$\|H^w \otimes \mathbf{1}_{h \times h} - w\|_2^2 \leq \frac{C'}{k^2}, \quad (17)$$

where  $C'$  is a constant independent of  $n$ .

Therefore, it remains to find an upper bound of  $\|\hat{w}^{tv} - H^w\|_2^2$ . (The last expectation in (15) can be bounded using Cauchy's inequality.) In the following subsections, we discuss how each step of the SAS algorithm contributes to this upper bound.

### 4.1. Consistency of empirical degree sorting

To establish the consistency of the empirical degree sorting, we must first establish the relationship between the oracle permutation ( $\sigma(i)$ ) and the oracle degree ( $d_{\sigma(i)}$ ).

**Lemma 2.** *Let  $\sigma(i)$  be the oracle permutation such that  $U_{\sigma(1)} < U_{\sigma(2)} < \dots < U_{\sigma(n)}$ . Let  $g(u) = \int_0^1 w(u, v) dv$ , and assume that there exists constants  $L_1 > 0$  and  $L_2 > 0$  such that*

$$L_2 |x - y| \leq |g(x) - g(y)| \leq L_1 |x - y|, \quad (18)$$

for any  $0 \leq x \leq 1$  and  $0 \leq y \leq 1$ . Then, the following result holds.

If  $\left| \frac{\sigma(i)}{n} - \frac{\sigma(j)}{n} \right| < \frac{1}{6L_1} \sqrt{\frac{\log n}{n}}$ , then

$$|d_{\sigma(i)} - d_{\sigma(j)}| < \sqrt{\frac{\log n}{n}}, \quad (19)$$

with probability at least  $1 - 8e^{-\frac{1}{18L_1^2} \log n}$ .

Conversely, if (19) holds with probability at least  $1 - 8e^{-\frac{1}{18L_1^2} \log n}$ , then

$$\left| \frac{\sigma(i)}{n} - \frac{\sigma(j)}{n} \right| < \sqrt{\frac{\log n}{n}} \left[ \frac{1}{3L_1} + \frac{1}{3L_1 L_2} + \frac{1}{L_2} \right], \quad (20)$$

with probability at least  $1 - 40e^{-\frac{1}{18L_1^2} \log n}$ .

The interpretation of Lemma 2 is as follows. First, (18) is the two-sided Lipschitz condition, with Lipschitz constants  $L_1$  and  $L_2$ . The Lipschitz condition enforces the degree distribution  $g(u)$  to be well-behaved so that there is no abrupt transition for both  $g$  and  $g^{-1}$ . Second, the forward statement suggests that if the oracle ordered indices have bounded differences, then correspondingly the empirical degrees should also have bounded differences. Conversely, (20) suggests that if we can bound the difference in empirical degrees, then the difference in the true positions should also be bounded.

As an immediate consequence of Lemma 2, we observe that for any fixed  $i$ , if we choose  $j$  such that  $\sigma(j) = \hat{\sigma}(i)$ , then the converse of Lemma 2 implies the following.

**Corollary 1.** *If  $|d_{\sigma(i)} - d_{\hat{\sigma}(i)}| < \sqrt{\frac{\log n}{n}}$  holds with probability at least  $1 - 8e^{-\frac{1}{18L_1^2} \log n}$ , then*

$$\left| \frac{\sigma(i)}{n} - \frac{\hat{\sigma}(i)}{n} \right| < C \sqrt{\frac{\log n}{n}}$$

*holds with probability at least  $1 - 40e^{-\frac{1}{18L_1^2} \log n}$ , where  $C = \frac{1}{3L_1} + \frac{1}{3L_1L_2} + \frac{1}{L_2}$  is a constant independent of  $n$ .*

Therefore, if the error  $|d_{\sigma(i)} - d_{\hat{\sigma}(i)}|$  is small, then the error between  $\sigma(i)$  and  $\hat{\sigma}(i)$  will also be small.

#### 4.2. Consistency of the histogram estimator

During the histogram estimation step, the error associated with the empirical degree sorting is translated to the error between the empirical histogram  $\hat{H}$  and the ideal histogram  $H$ . This is reflected in the following lemma.

**Lemma 3** (Bounds on  $\|\hat{H} - H\|_2$ ). *Let  $w$  be the ground truth graphon and assume that  $w$  is Lipschitz with constant  $L > 0$ . If  $H$  and  $\hat{H}$  are defined according to (11) and (10), respectively, then*

$$\mathbb{E}[\|\hat{H} - H\|_2^2] \leq \frac{k^4}{n^2} \left( 2 + 4C^2L^2 \frac{\log n}{n} \right) + k^2 \left( 4C^2L^2 \frac{\log n}{n} \right), \quad (21)$$

where  $C$  is a constant independent of  $n$ .

We also establish the relationship between  $H$  and the step approximation  $H^w$ .

**Lemma 4** (Bounds on  $\|H - H^w\|_2$ ). *Let  $H^w$  be the step function approximation of the graphon  $w$  and let  $H$  be the histogram defined as (11). Then,*

$$\mathbb{E}[\|H - H^w\|_2^2] \leq \frac{k^4}{n^2}. \quad (22)$$

#### 4.3. Consistency of total variation smoothing

To analyze the total variation minimization step, we first observe that

$$\hat{H} = H^w + \underbrace{\hat{H} - H}_{\eta} + \underbrace{H - H^w}_{\rho}. \quad (23)$$

Therefore, if we consider  $H^w$  as the desired function to be estimated, and consider  $\eta$  and  $\rho$  as perturbations added to  $H^w$ , then  $\hat{H}$  can be regarded as a noisy observation of  $H^w$ . Consequently, by applying total variation minimization to

(23), we find a solution  $\hat{w}^{tv}$  that best fits (23) and has the minimum total variation.

To characterize the solution of the total variation minimization problem, we first define the  $s$ -sparsity of the gradient of a function  $H^w$ .

**Definition 5.** *A function  $H^w \in [0, 1]^{k \times k}$  is  $s$ -sparse in gradient if its gradient  $\nabla H^w$  has at most  $s$  non-zero entries.*

With this definition, we apply the following result in compressed sensing.

**Lemma 5** ((Needell & Ward) Theorem A). *If  $\hat{H} = H^w + \eta + \rho$  with  $\varepsilon^2 = \mathbb{E}[\|\eta + \rho\|_2^2]$ , then the solution  $\hat{w}^{tv}$  of*

$$\hat{w}^{tv} = \underset{\hat{r}}{\operatorname{argmin}} \|\hat{r}\|_{TV} \quad \text{subject to} \quad \|\hat{r} - \hat{H}\|_2 \leq \varepsilon,$$

*satisfies the condition*

$$\|\hat{w}^{tv} - H^w\|_2 \leq \frac{\|\nabla H^w - (\nabla H^w)_s\|_1}{\sqrt{s}} + \varepsilon,$$

where  $(\cdot)_s$  denotes the function reconstructed from the  $s$  most significant non-zero entries of the argument.

Lemma 5 indicates that the error  $\|\hat{w}^{tv} - H^w\|_2$  is controlled by the perturbation  $\varepsilon$  and the sparse approximation error  $\|\nabla H^w - (\nabla H^w)_s\|_1$ . Since  $\varepsilon^2 = \mathbb{E}[\|\eta + \rho\|_2^2]$ , and  $\eta$  and  $\rho$  are defined according to (23),  $\varepsilon$  is upper bounded by (21) and (22). For the sparse approximation error term, in general  $\|\nabla H^w - (\nabla H^w)_s\|_1 \neq 0$  because  $H^w$  is not necessarily  $s$ -sparse in gradient. However, in practice, many real world networks are sparse (*i.e.* number of edges are much fewer than number of nodes). Therefore, for practical consideration it is often reasonable to assume that  $H^w$  is  $s$ -sparse in gradient and so  $\|\nabla H^w - (\nabla H^w)_s\|_1 = 0$ .

#### 4.4. Overall consistency

In summary, the overall consistency is given by the following theorem.

**Theorem 3** (Consistency of SAS algorithm). *Let  $w$  be the true graphon with the following properties: (i)  $w$  is Lipschitz with constant  $L > 0$ ; (ii)  $g(u) = \int_0^1 w(u, v) dv$  is Lipschitz as defined in Lemma 2; (iii)  $H^w$  is  $s$ -sparse in gradient. Then, the MSE of the SAS estimator satisfies*

$$\text{MSE} \leq \mathcal{O} \left( \frac{\log n}{n} \right), \quad (24)$$

and hence  $\text{MSE} \rightarrow 0$  as  $n \rightarrow \infty$  and  $k/n \rightarrow 0$ , where  $k$  is the number of blocks defined in (11).

### 5. Experimental results

After establishing the theoretical results, we now present simulation results of the proposed SAS algorithm.

### 5.1. Simulations

The first experiment considers a number of graphons listed in Table 1. The choices of these graphons are made to include both low rank and high rank graphons, where the rank is measured numerically using a  $1000 \times 1000$  discretization of the continuous graphons. Among the 10 graphons listed in Table 1, we note that graphon no. 1  $w(u, v) = uv$  is a special case of the eigenmodel (Hoff, 2008), graphon no. 5  $w(u, v) = 1/(1 + \exp\{-10(u^2 + v^2)\})$  is a variation of the logistic model presented in (Chatterjee), and graphon no. 6  $w(u, v) = |u - v|$  is the latent distance model (Hoff et al., 2002). Other graphons are chosen to demonstrate the robustness of the SAS algorithm.

ID	$w(u, v)$	$\text{rank}(w)$
1	$uv$	1
2	$\exp\{-(u^{0.7} + v^{0.7})\}$	1
3	$\frac{1}{4}[u^2 + v^2 + u^{1/2} + v^{1/2}]$	2
4	$\frac{1}{2}(u + v)$	2
5	$\frac{1}{1 + \exp\{-10(u^2 + v^2)\}}$	10
6	$ u - v $	1000
7	$\frac{1}{1 + \exp\{-(\max(u, v)^2 + \min(u, v)^4)\}}$	1000
8	$\exp\{-\max(u, v)^{3/4}\}$	1000
9	$\exp\{-\frac{1}{2}(\min(u, v) + u^{1/2} + v^{1/2})\}$	1000
10	$\log(1 + 0.5 \max(u, v))$	1000

Table 1. List of graphons for testing. The rank of  $w$  is estimated from a  $1000 \times 1000$  discretization of the graphon.

We compare the SAS algorithm with the universal singular value thresholding (USVT) algorithm (Chatterjee) and the stochastic blockmodel approximation algorithm (Airoldi et al., 2013). These two algorithms are the existing methods that have provable consistency and are numerically efficient. However, since both of these two methods do not have a sorting step, we apply the sorting step of the SAS algorithm prior to running the two algorithms. For the choice of binwidth  $h$ , we set  $h = \log n$  for the SAS algorithm, and an oracle  $h$  that minimizes the MSE for the SBA algorithm (*i.e.*, using the ground truth).

The results of the experiment are shown in Table 2, where we report the mean squared error (MSE) of the estimated graphons using the SAS algorithm, the USVT algorithm and the SBA algorithm. To reduce the random fluctuations caused by independent realizations of the random graphs, we average the MSE over 50 independent trials. Two cases of graph sizes are considered:  $n = 200$  and  $n = 1000$ . The results show that the SAS algorithm in general outperforms the USVT algorithm and the SBA algorithm. Averaged over the 10 testing graphons, we see that the SAS algorithm achieves the lowest MSE among all three methods.

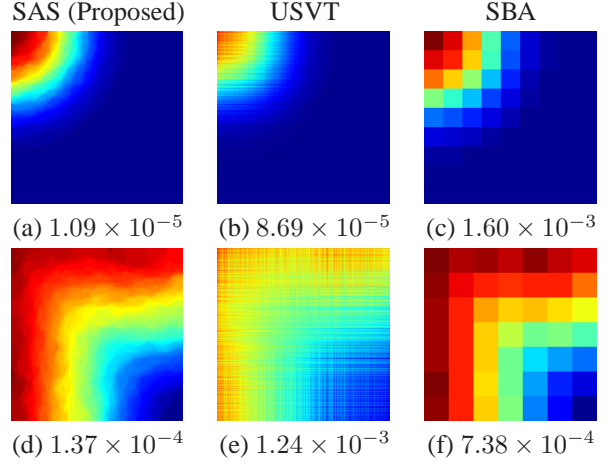


Figure 3. Comparisons between the SAS algorithm, the USVT algorithm (Chatterjee), and the SBA algorithm (Airoldi et al., 2013). Numbers indicate the mean squared error. (a)-(c): Graphon 5; (d)-(f): Graphon 10. SAS algorithm uses  $h = \log n$ . SBA algorithm uses an oracle  $h$  that minimizes the MSE. In this example, we set  $n = 1000$ .

Figure 3 displays two examples of the estimated graphons. As shown in the figure, we see that while the USVT algorithm returns a reasonable estimate for graphon no.5 (which has a low rank), it returns a relatively worse estimate for graphon no. 10 (which has a high rank). Looking at the SBA algorithm, it is evident that using the oracle binwidth  $h$ , the average MSE is lower than that of USVT. However, the SBA algorithm tends to return a graphon with few communities. This is not favorable if the network has non-block structures. In contrast, the SAS algorithm returns results with lower MSE, and retains important features of the true graphons.

In Figure 4 we show the runtime comparison between the SAS algorithm and the USVT algorithm. Both algorithms are implemented on an Intel 3.5GHz machine with 16GB RAM, Windows 7 / MATLAB R7.12.0 platform. The runtime plot indicates that the SAS algorithm has a significantly lower complexity than the USVT algorithm.

### 5.2. Real data analysis

As an application of the proposed SAS algorithm, we consider the problem of estimating graphons from real-world networks. For this purpose, we consider the collaboration network of arXiv astro physics (ca-AstroPh) and the who-trusts-whom network of Epinions.com (soc-Epinions1) from Stanford Large Network Dataset Collection<sup>1</sup>. The ca-AstroPh network is a symmetric binary graph consisting of  $1.8 \times 10^4$  nodes and  $3.9 \times 10^5$  edges, whereas the soc-Epinions-1 network is an unsymmetrical binary

<sup>1</sup><http://www.cise.ufl.edu/research/sparse/matrices/SNAP/>

ID	$n = 200$			$n = 1000$		
	SAS (Proposed)	USVT (Chatterjee)	SBA (Airolidi et al., 2013)	SAS (Proposed)	USVT (Chatterjee)	SBA (Airolidi et al., 2013)
1	$6.59\text{e-}04 \pm 5.18\text{e-}05$	$1.90\text{e-}03 \pm 1.88\text{e-}04$	$2.77\text{e-}03 \pm 1.60\text{e-}04$	$8.56\text{e-}05 \pm 3.42\text{e-}06$	$3.86\text{e-}04 \pm 1.70\text{e-}05$	$9.00\text{e-}04 \pm 1.70\text{e-}05$
2	$4.92\text{e-}04 \pm 6.81\text{e-}05$	$2.18\text{e-}03 \pm 1.95\text{e-}04$	$2.36\text{e-}03 \pm 1.97\text{e-}04$	$7.12\text{e-}05 \pm 5.92\text{e-}06$	$4.46\text{e-}04 \pm 1.84\text{e-}05$	$1.39\text{e-}03 \pm 3.99\text{e-}05$
3	$6.95\text{e-}04 \pm 7.52\text{e-}05$	$3.12\text{e-}03 \pm 2.32\text{e-}04$	$5.08\text{e-}03 \pm 2.26\text{e-}04$	$9.60\text{e-}05 \pm 5.78\text{e-}06$	$9.69\text{e-}04 \pm 2.67\text{e-}05$	$8.66\text{e-}04 \pm 1.90\text{e-}05$
4	$6.48\text{e-}04 \pm 5.30\text{e-}05$	$3.51\text{e-}03 \pm 1.93\text{e-}04$	$2.77\text{e-}03 \pm 1.49\text{e-}04$	$7.82\text{e-}05 \pm 5.17\text{e-}06$	$8.83\text{e-}04 \pm 2.47\text{e-}05$	$1.43\text{e-}03 \pm 2.63\text{e-}05$
5	$9.74\text{e-}05 \pm 2.76\text{e-}05$	$3.15\text{e-}03 \pm 8.76\text{e-}04$	$3.13\text{e-}03 \pm 3.31\text{e-}04$	$1.09\text{e-}05 \pm 1.66\text{e-}06$	$8.69\text{e-}05 \pm 7.03\text{e-}06$	$1.60\text{e-}03 \pm 3.45\text{e-}05$
6	$4.29\text{e-}02 \pm 9.27\text{e-}05$	$8.91\text{e-}02 \pm 1.23\text{e-}03$	$4.37\text{e-}02 \pm 1.20\text{e-}04$	$4.19\text{e-}02 \pm 9.58\text{e-}06$	$8.42\text{e-}02 \pm 1.70\text{e-}04$	$4.22\text{e-}02 \pm 1.42\text{e-}05$
7	$4.81\text{e-}04 \pm 7.50\text{e-}05$	$2.40\text{e-}03 \pm 1.77\text{e-}04$	$2.71\text{e-}03 \pm 2.09\text{e-}04$	$8.48\text{e-}05 \pm 7.47\text{e-}06$	$6.76\text{e-}04 \pm 1.81\text{e-}05$	$1.21\text{e-}03 \pm 3.65\text{e-}05$
8	$9.38\text{e-}04 \pm 1.21\text{e-}04$	$6.27\text{e-}03 \pm 1.58\text{e-}03$	$1.52\text{e-}03 \pm 1.52\text{e-}04$	$1.73\text{e-}04 \pm 1.30\text{e-}05$	$1.66\text{e-}03 \pm 4.56\text{e-}05$	$6.81\text{e-}04 \pm 2.14\text{e-}05$
9	$6.50\text{e-}04 \pm 7.73\text{e-}05$	$2.87\text{e-}03 \pm 2.32\text{e-}04$	$3.96\text{e-}03 \pm 3.25\text{e-}04$	$1.02\text{e-}04 \pm 5.15\text{e-}06$	$1.26\text{e-}03 \pm 3.01\text{e-}05$	$1.15\text{e-}03 \pm 3.44\text{e-}05$
10	$7.67\text{e-}04 \pm 1.01\text{e-}04$	$4.74\text{e-}03 \pm 6.25\text{e-}04$	$1.13\text{e-}03 \pm 1.23\text{e-}04$	$1.37\text{e-}04 \pm 1.02\text{e-}05$	$1.24\text{e-}03 \pm 3.30\text{e-}05$	$7.38\text{e-}04 \pm 1.67\text{e-}05$
Average	$4.83\text{e-}03 \pm 7.43\text{e-}05$	$1.19\text{e-}02 \pm 4.65\text{e-}04$	$6.91\text{e-}03 \pm 1.99\text{e-}04$	$4.27\text{e-}03 \pm 6.74\text{e-}06$	$9.18\text{e-}03 \pm 3.91\text{e-}05$	$5.22\text{e-}03 \pm 2.06\text{e-}05$

Table 2. Mean squared error (average  $\pm$  std. dev.) comparisons between the SAS algorithm, the USVT algorithm (Chatterjee), and the SBA algorithm (Airolidi et al., 2013). MSE is averaged over 50 independent trials.

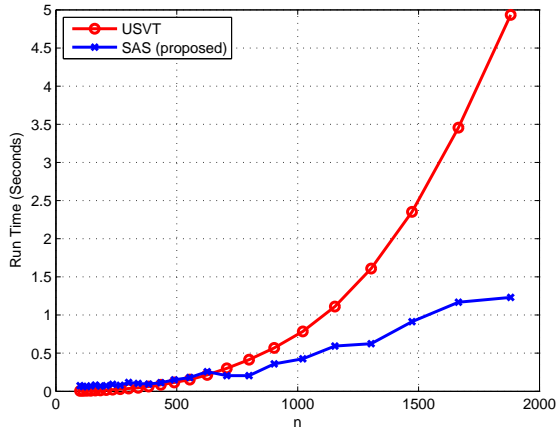


Figure 4. Run time comparison between USVT (Chatterjee) and the SAS algorithm (averaged over 10 graphons listed in Table 1).

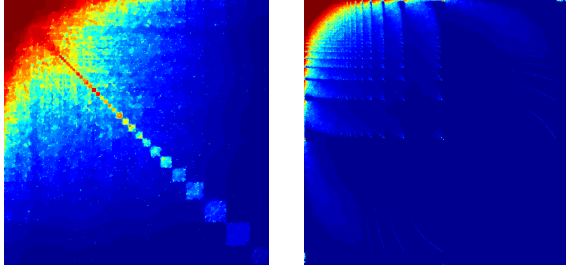


Figure 5. Estimated graphons for real networks. Left: Collaboration network of arXiv astro physics (ca-AstroPh)  $n = 1.8 \times 10^4$ . Right: who-trusts-whom network of Epinions.com (soc-Epinions1)  $n = 7.5 \times 10^4$ .

graph consisting of  $7.5 \times 10^4$  nodes and  $5.1 \times 10^5$  edges. For both networks, we randomly permute the rows and columns to simulate the raw data scenario where nodes are initially unordered.

Figure 5 shows the results of the SAS algorithm. For the

ca-AstroPh network, the graphon shows close collaborations among a group of people concentrated around the top left corner of the graphon. It also shows a number of small communities along the diagonal. For the soc-Epinions1 network, the graphon indicates that there are some influential nodes which consistently interact among themselves. These can be seen from the repeated patterns of the graphon.

We remark that for the ca-AstroPh network ( $n = 1.8 \times 10^4$ ) and the soc-Epinions-1 network ( $n = 7.5 \times 10^4$ ), the estimations are completed in 20 seconds and 170 seconds, respectively, on a PC using an unoptimized MATLAB code. This provides a strong indication of the scalability of the SAS algorithm to larger networks.

## 6. Concluding remarks

The Sorting-And-Smoothing (SAS) algorithm is a consistent and efficient graphon estimation algorithm. The SAS algorithm consists of two steps. In the first step, the observed graph is rearranged so that the degrees are monotonically increasing. In the second step, a histogram estimation and a total variation minimization is applied to estimate a smooth surface that best fits the observed data. The SAS algorithm is evaluated on both simulation data and real network data. Our simulation results indicate that the SAS algorithm outperforms the universal singular value thresholding algorithm and the stochastic blockmodel approximation algorithm. On large-scale real networks, the SAS algorithm returns consistent graphon estimates.

**Code.** Available at: <https://github.com/airoldilab/SAS>

**Acknowledgments.** The authors thank J. J. Yang and C. Q. Han for useful discussions. SHC is partially supported by a Croucher Foundation Postdoctoral Research Fellowship. EMA is partially supported by NSF CAREER award IIS-1149662, AROMURI award W911NF-11-1-0036, and an Alfred P. Sloan Research Fellowship.



## References

- Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. Mixed-membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.
- Airoldi, E. M., Bai, X., and Carley, K. M. Network sampling and classification: An investigation of network model representations. *Decision Support Systems*, 51:506–518, Jun. 2011.
- Airoldi, E. M., Costa, T. B., and Chan, S. H. Stochastic block-model approximation of a graphon: Theory and consistent estimation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 26, pp. 692–700, 2013. ArXiv:1311.1731.
- Aldous, D. J. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4): 581–598, Dec. 1981.
- Azari, H. and Airoldi, E. M. Graphlet decomposition of a weighted network. *Journal of Machine Learning Research, W&CP*, 22:54–63, 2012.
- Bickel, P. J. and Chen, A. A nonparametric view of network models and Newman-Girvan and other modularities. *Proc. Natl. Acad. Sci. USA*, 106(50):21068–21073, Dec. 2009.
- Bickel, P. J., Chen, A., and Levina, E. The method of moments and degree distributions for network models. *The Annals of Statistics*, 39(5):2280–2301, 2011.
- Borgs, C., Chayes, J., and Lovász, L. Moments of two-variable functions and the uniqueness of graph limits. *Geom. Funct. Anal.*, 19:1597–1619, Mar. 2010.
- Chan, S. H. and Airoldi, E. M. A consistent histogram estimator for exchangeable graph models. arXiv:1402.1888, Feb. 2014.
- Chan, S. H., Khoshabeh, R., Gibson, K. B., Gill, P. E., and Nguyen, T. Q. An augmented Lagrangian method for total variation video restoration. *IEEE Trans. Image Process.*, 20(11):3097–3111, Nov. 2011.
- Chan, S. H., Costa, T. B., and Airoldi, E. M. Estimation of exchangeable random graph models by stochastic blockmodel approximation. In *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 293–296, 2013.
- Chatterjee, S. Matrix estimation by universal singular value thresholding. ArXiv:1212.1247. 2012.
- Choi, D. S., Wolfe, P. J., and Airoldi, E. M. Stochastic block-models with a growing number of classes. *Biometrika*, 99(2): 273–284, Jun. 2012.
- Choi, D.S. and Wolfe, P.J. Co-clustering separately exchangeable network data. ArXiv:1212.4093. 2012.
- Diaconis, P. and Janson, S. Graph limits and exchangeable random graphs. *Rendiconti di Matematica e delle sue Applicazioni, Series VII*, pp. 33–61, 2008.
- Goldenberg, A., Zheng, A. X., Fienberg, S. E., and Airoldi, E. M. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, Feb. 2009.
- Hoff, P. D. Modeling homophily and stochastic equivalence in symmetric relational data. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pp. 657–664, 2008.
- Hoff, P. D., Raftery, A. E., and Handcock, M. S. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, Dec. 2002.
- Hoover, D. Relations on probability spaces and arrays of random variables. Institute for Advanced Study, Princeton, NJ, 1979.
- Hunter, D. R. and Handcock, M. S. Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*, 15(3):565–583, 2006.
- Kallenberg, O. *Probabilistic Symmetries and Invariance Principles*. Springer, 2005.
- Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from a few entries. *IEEE Trans. Information Theory*, 56:2980–2998, Jun. 2010.
- Kolaczyk, E. *Statistical Analysis of Network Data: Methods and Models*. Springer, 2009.
- Latouche, P. and Robin, S. Bayesian model averaging of stochastic block models to estimate the graphon function and motif frequencies in a w-graph model. ArXiv:1310.6150, Oct. 2013. Unpublished manuscript.
- Lloyd, J. R., Orbanz, P., Ghahramani, Z., and Roy, D. M. Random function priors for exchangeable arrays with applications to graphs and relational data. In *Advances in Neural Information Processing Systems (NIPS)*, volume 25, pp. 1007–1015, 2012.
- Lovász, L. and Szegedy, B. Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B*, 96:933–957, 2006.
- Needell, D. and Ward, R. Stable image reconstruction using total variation minimization. ArXiv:1202.6429. 2013.
- Nowicki, K. and Snijders, T. Estimation and prediction of stochastic block structures. *Journal of American Statistical Association*, 96:1077–1087, 2001.
- Olhede, S. C. and Wolfe, P. J. Network histograms and universality of blockmodel approximation. ArXiv:1312.5306. 2013.
- Orbanz, P. and Roy, D. M. Bayesian models of graphs, arrays and other exchangeable random structures. Unpublished manuscript. 2013.
- Tang, M., Sussman, D. L., and Priebe, C. E. Universally consistent vertex classification for latent positions graphs. *The Annals of Statistics*, 41:1406–1430, 2013.
- Wasserman, L. *All of Nonparametric Statistics*. Springer, 2005.
- Wolfe, P. J. and Olhede, S. C. Nonparametric graphon estimation. ArXiv:1309.5936. 2013.
- Yang, J. J., Han, C. Q., and Airoldi, E. M. Nonparametric estimation and testing of exchangeable graph models. In *Journal of Machine Learning Research, W & CP (AISTATS)*, 2014. In press.