A Non-Parametric Perspective on the Analysis of Massive Networks

A non-parametric perspective on the analysis of massive networks

A dissertation presented

by

Thiago Barros Rodrigues Costa

to

The Harvard School of Engineering and Applied Science

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Applied Mathematics

Harvard University

Cambridge, Massachusetts

October 2014

Edoardo Airoldi Thiago Barros Rodrigues Costa

A non-parametric perspective on the analysis of massive networks

Abstract

This dissertation develops an inferential framework for a highly non-parametric class of network models called graphons, which are the limit objects of converging sequences in the theory of dense graph limits. The theory, introduced by Lovász and co-authors, uses structural properties of very large networks to describe a notion of convergence for sequences of dense graphs. Converging sequences define a limit which can be represented by a class of random graphs that preserve many properties of the networks in the sequence. These random graphs are intuitive and have a relatively simple mathematical representation, but they are very difficult to estimate due to their non-parametric nature. Our work, which develops scalable and consistent methods for estimating graphons, offers an algorithmic framework that can be used to unlock the potential of applications of this powerful theory.

To estimate graphons we use a stochastic blockmodel approximation approach that defines a notion of similarity between vertices to cluster vertices and find the blocks. We show how to compute these similarity distances from a given graph and how to properly cluster the vertices of the graph in order to form the blocks. The method is non-parametric, i.e., it uses the data to choose a convenient number of clusters. Our approach requires a careful balance between the number of blocks created, which is associated with stochastic blockmodel approximation of the graphon, and the size of the clusters, which is associated with the estimation of the stochastic blockmodel parameters. We prove insightful properties regard-

ing the clustering mechanism and the similarity distance, and we also work with important variations of the graphon model, including a sparser type of graphon.

As an application of our framework, we use the stochastic blockmodel nature of our method to improve identification of treatment response with social interaction. We show how the graph structure provided by our algorithm can be explored to design optimal experiments for assessing social effect on treatment response.

Contents

Dedicated to my mother Marusia and to my father Efísio.

Acknowledgments

First I would like to express my profound gratitude to my advisor, Professor Edoardo Airoldi, for all the support he gave me along these many years of graduate school, not only because his guidance helped keep my research at the highest intellectual standards but also because his friendship and trust helped me pass through the frustrations of research with confidence and motivation. I also thank the other members of my committee, in special Professor Michael Brenner, whose pieces of advice were crucial for the success of my dissertation. And I thank Harvard, in particular my department, whose incredibly rich and diverse environment has so deeply shaped my academic and intellectual maturity.

My Ph.D. wouldn't have been possible without the support of a precious network of friends. Starting with the ones who introduced me to the joy of math and science, including my academic mentors back home in Brazil, Professors Plamen Koshlukov, Carlos Gustavo Moreira, and Jacob Palis and all my friends from the Brazilian Math Olympiad. I also would like to mention those who have always been on my side during the good and the difficult times of graduate school, with special thanks to André Carneiro and Renato Leme. Thank you so much.

Last but not least, I thank my family, specially my mother Marusia and my father Efísio, whose dedication and love gave me fuel to become the person I am today. I also thank my brothers, Daniel and Victor, who always illuminated my life with inspiration and love. Finally, I thank Andrei, whose companionship and dedication has paved a path towards the pursuit of my dreams.

1

Introduction to the theory of graph limits

This chapter is an introduction to the theory of graph limits, which the essential background and mathematical foundation of most of the work presented in this dissertation.

1.1 Graphons and convergence of dense graph sequences

Recently, L. Lovász and co-authors developed a theory of graph limits that beautifully unifies different notions of convergence for a sequence of dense graphs and predicts the existence of a limit object that preserves many local and global properties of the graphs in the sequence. These objects, called graphons, can be represented in many ways and one intuitive definition is described as follows. We say that (G_n) is convergent if, for every graph F, the proportion of copies of F as a subgraph of G_n converges, i.e., if the density $t(F, G_n)$ of adjacency-preserving maps from F to G_n converges as $n \to \infty$. Converging sequences are associated with a limit object that can be used to describe a highly non-parametric class of random graph models. In this dissertation, we develop a scalable computational framework to explore these random graphs in a way to unlock the potential of this rich mathematical theory for applications.

To precisely define $t(F, G)$ in a general setting, let G be a weighted graph with a weight $\alpha_i > 0$ on each node i and a weight $\beta_{ij} \in \mathbb{R}$ on each edge ij (for unweighted graphs we set $\alpha_i = 1$ for all i, $\beta_{ij} = 1$ for all edges, $\beta_{ij} = 0$ for all non-edges). If F is a simple graph, let the number of homomorphisms from F to G be:

$$hom(F, G) = \sum_{\phi: V(F) \to V(G)} \prod_{i \in V(F)} \alpha_{\phi(i)}(G) \prod_{ij \in E(F)} \beta_{\phi(i)\phi(j)}(G). \tag{1.1}$$

Normalization of $hom(F, G)$ gives the homomorphism density:

$$t(F, G) = \frac{hom(F, G)}{\alpha_g^k}, \tag{1.2}$$

where k is the number of nodes in F and $\alpha_G = \sum_i \alpha_i(G)$ (the notation followed from [16]).

When the limit $\lim_{n\to\infty} t(F, G_n)$ exists for every simple graph F, we say that the sequence (G_n) is *left-convergent*. Lovász shows in [41] that if (G_n) left-converges then there exists a symmetric measurable function $w : [0,1]^2 \to [0,1]$ such that $\lim_{n\to\infty} t(F, G_n) = t(F, w)$, where $t(F, w)$ is the density of F in w defined by

$$t(F, w) = \int_{[0,1]^{|V(F)|}} \prod_{ij \in E(F)} w(x_i, x_j) dx. \tag{1.3}$$

For a given converging sequence, the function w is unique up to measure preserving transformations. These equivalence classes defining the limit objects are called graphons.

Lovász's representation of the limit object provides a natural way to interpret graphons as random graphs. Given a symmetric measurable function $w : [0,1]^2 \to [0,1]$ and an integer n, one can define the random graph $G(n, w)$ with n vertices by first sampling i.d.d. $u_1, \ldots, u_n \sim \text{Uniform}[0,1]$, then independently connecting every pair of vertices (i, j) with probability $w(u_i, u_j)$. Clearly, $G(n, w)$ doesn't change with any measure preserving transformation on w, therefore it only depends on the graphon defined by w. Conversely, given $G(n, w)$, one may get the graphon associated with it by taking the limit of a sequence of graphs (G_n) sampled from $G(n, w)$, with $n \to \infty$. It is proved in [41] that a sequence (G_n) generated this way converges, with high probability, to the graphon associated with w.

Given the highly non-parametric nature of graphons, estimating $G(n, w)$ from observed data is a complex task. In this thesis, we develop a class of stochastic

blockmodel approximation algorithms to solve the problem of graphon estimation with a scalable computational framework. We show how our results connect with the theory and develop applications in indetification of treatment response with social interactions.

1.2 Szemerédi partitions of graphs

The methods used by L. Lovász to develop the graph limits theory for dense graphs strongly rely on the Szemerédi Regularity Lemma. The lemma shows that one can partition the vertex set of large graphs in a way that connections between partitions have some interesting regularity patterns. In this section we review this result, as it provides intuition and motivation for much of the work presented in the following chapters.

For a given weighted graph G and given subsets $S, T \subset V(G)$, let

$$e_G(S,T) = \sum_{i \in S, j \in T} \alpha_i(G)\alpha_j(G)\beta_{ij}(G). \tag{1.4}$$

When G is unweighted, $e_G(S,T)$ corresponds to the number of edges connecting vertices of S to vertices of T.

Given two weighted graphs G and G' on the same vertex set V, their *cut distance* (or *rectangular distance*) is defined as:

$$d_\square(G,G') = max_{S,T \subset V} \frac{1}{\alpha_G^2} \left| e_G(S,T) - e_{G'}(S,T) \right|. \tag{1.5}$$

A small *cut distance* between G and G' means that the frequency of connections

between any two subsets of vertices is similar in the two graphs.

To have a distance that doesn't depend on the labeling of the vertices, one may consider

$$\hat{\delta}_{\square}(G, G') = \min_{\tilde{G} \cong G} d_{\square}(\tilde{G}, G'), \tag{1.6}$$

where $\tilde{G}$ ranges over graphs that are isomorphic to G.

The notion of *cut distance* can be extended to graphs of different sizes. Consider two weighted graphs G and G', with respective sizes n and n', assuming that their total nodeweight is 1. We say that an $n \times n'$ matrix X is a *fractional overlay* of G and G' if

$$\sum_{u=1}^{n'} X_{iu} = \alpha_i(G), \ \sum_{i=1}^{n} X_{iu} = \alpha_u(G'). \tag{1.7}$$

The space of fractional overlays is denoted by $\mathcal{X}(G, G')$. Given $X \in \mathcal{X}(G, G')$, one may define the *overlaid graphs* $G[X]$ and $G'[X^T]$: the weight of an node $(i, u) \in [n] \times [n']$ is X_{iu}; the weight of an edge $((i, u), (j, v))$ in $G[X]$ is β_{ij}, and in $G'[X^T]$ is β'_{uv}. Since $G[X]$ and $G'[X^T]$ have the same vertex set, the *cut distance* $d_{\square}(G[X], G'[X^T])$ is well defined. Analogous to (1.6) we let

$$\delta_{\square}(G, G') = \min_{X \in \mathcal{X}(G,G')} d_{\square}(G[X], G[X^T]). \tag{1.8}$$

We apply this definition to describe the Szemerédi partitions. For a partition $\mathcal{P} = \{V_1, \ldots, V_k\}$ of $V(G)$, the weighted graph $G/\mathcal{P}$ (*quotiente graph*) on k vertices is defined as: $\alpha_i(G/\mathcal{P}) = \alpha_{V_i}/\alpha_G$ are the nodeweights and $\beta_{ij}(G/\mathcal{P}) = \frac{e_G(V_i,V_j)}{\alpha_{V_i}\alpha_{V_j}}$ are the edgeweights, where $\alpha_{V_i} = \sum_{x \in V_i} \alpha_x(G)$. $G/\mathcal{P}$ is in some sense an average of G in the given partitions. The Weak Regularity Lemma (introduced in [27] but

reproduced here from [16]) states that:

Lemma 1. *(Weak Regularity Lemma) For every $\epsilon > 0$, every weighted graph G has a partition $\mathcal{P}$ into at most $4^{1/\epsilon^2}$ classes such that*

$$d_{\square}(G, G_{\mathcal{P}}) \leq \epsilon ||G||_2, \tag{1.9}$$

where $||G||_2 = \left(\sum_{i,j} \frac{\alpha_i \alpha_j}{\alpha_G^2} \beta_{ij}^2\right)^{1/2}$

The lemma means that every G can be approximated to a smaller weighted $G/\mathcal{P}$ in a way to somehow preserve the frequency of connections between subsets of nodes. It is a very strong and powerful result, which is used by Lovász to prove the existence of graphons as limits of a sequence of graphs. In his developments, the graphon function $w : [0,1]^2 \to [0,1]$ emerges as the limit of step functions defined from the Szemerédi partitions. This idea of approximating w with step functions (or, equivalently, approximating $G(n,w)$ with stochastic blockmodels) is the heart of our SBA algorithm.

1.3 Related notions of convergence

The *left-convergence* defined above gives a very good intuition about some useful local properties of the limit object, as it preserves density of any type of subgraph of the graphs in the sequence. It turns out that, under certain conditions, *left-convergence* is related to other notions of convergence, enriching the theory of graph limits with a global flavor and empowering the space of graphons with more structure. These connections are explored in this section.

1.3.1 Exchangeable arrays

The first interesting connection we consider is deeply discussed in [25] and relates the theory of graphons to the theory of exchangeable arrays of Aldous-Hoover. We review the main points from [25] here.

Let X_{ij}, $1 \leq i, j < \infty$, be a collection of binary random variables. We say that they are *separately exchangeable* if

$$\mathbb{P}\left(X_{ij} = e_{ij}, 1 \leq i, j \leq n\right) = \mathbb{P}\left(X_{ij} = e_{\sigma(i)\tau(j)}, 1 \leq i, j \leq n\right), \tag{1.10}$$

for all n, all permutations σ, τ of $[n]$ and all $e_{ij} \in \{0, 1\}$. They are *jointly exchangeable* if the above equation holds in the particular case of $\tau = \sigma$.

A binary random array is defined as follows: let u_i, v_j, $1 \leq i, j \leq \infty$ be independent and Uniform$[0, 1]$. Consider $W : [0, 1]^2 \to [0, 1]$ and let $X_{ij} \sim$ Binomial$(W(u_i, v_j))$.

The Aldous-Hoover theorem (replicated from [25]) states that:

Theorem 2. *(Aldous-Hoover). Let $X = \{X_{ij}\}, 1 \leq i, j \leq \infty$, be a separably exchangeable random array. Then, there is a probability μ such that*

$$\mathbb{P}\{X \in A\} = \int P_w(A)\mu(dw). \tag{1.11}$$

Diaconis draws a connection between the measure P_w of the Aldous-Hoover theorem and the limit object w of the theory of graph limits ([25]): every proper graph limit corresponds to an extreme point in the set of distributions of ex-

changeable random graphs. It is important to point out that the graph limits theory brings a whole new perspective to Algous-Hoover's work, as it allows useful algorithmic developments and builds important connections with other fields.

1.3.2 Right-convergence

The notion of *left-convergence* described before, in which $t(F, G_n)$ converges to every F, permits a very local perspective to the theory of graph limits as it is entirely based on counting subgraphs. Surprisingly, *left-convergence* is equivalent to another type of convergence based on global properties G_n, the *right-convergence* [17]. Instead of counting homomorphisms from a small F to G_n, *right-convergence* counts homomorphisms from G_n to F, being essentially a global coloring of G_n using nodes of F as colors.

To precisely define *right-convergence* we first need some notation (which we follow from [17]). Let (G_n) be a sequence of simple graphs and H a weighted *soft-core* graph, i.e., a graph with all loops present, positive nodeweights $\alpha_i(H) > 0$ and positive edgeweights $\beta_{ij}(H) = \beta_{ji}(H) > 0$. If q is the number vertices in H, let Pd_q be the set of vectors $a \in \mathbb{R}_q$ for which $\sum_{i=1}^{q} a_i = 1$, $a_i > 0 \forall i$. Define:

$$\Omega_a(G) = \{\phi : V(G) \to [q] : \left||\phi^{-1}(\{i\})| - a_i|V(G)|\right| \leq 1,\ \forall i \in [q]\} \tag{1.12}$$

and

$$hom_a(G, H) = \sum_{\phi \in \Omega_a(G)} \prod_{uv \in E(G)} \beta_{\phi(u)\phi(v)}(H). \tag{1.13}$$

We say that a sequence of simple graph G_n converges if, for every *soft-core* graph

H and every probability distribution $a \in Pd_q$ in $V(H)$, the expression

$$\frac{1}{|V(G)|^2} \ln hom_a(G_n, H) \tag{1.14}$$

converges as $n \to \infty$.

1.3.3 The cut distance

Left and right convergence are equivalent to convergence under $\delta_\square$, the metric defined in (1.8): a sequence of simple graphs (G_n) is *left-convergent* if and only if it is a Cauchy sequence on $\delta_\square$ [17].

An analogous notion of *cut distance* can also be defined in the space of graphons. Given a graphon W, let

$$||W||_\square = \sup_{S,T \subset [0,1]} \left| \int_{S \times T} W(x,y) dx dy \right| \tag{1.15}$$

The *cut distance* of two graphons is defined by

$$\delta_\square(U, W) = \inf_{\phi:[0,1]\to[0,1]} ||U - W^\phi||, \tag{1.16}$$

where $\phi : [0,1] \to [0,1]$ is invertible and is such that both ϕ and it's inverse are measurable preserving, and $W^\phi(x,y) = W(\phi(x), \phi(y))$.

It is not hard to prove that every weighted graph G has a natural graphon associated with it such that $\delta_\square(G, G') = \delta_\square(W_G, W_{G'})$. W_G can be described as follows: from a family of disjoint intervals $I_1, \ldots, I_n \subset [0,1]$ of respective length $\frac{\alpha_1(G)}{\alpha_G}, \ldots, \frac{\alpha_n(G)}{\alpha_G}$, let $W_G : [0,1]^2 \to [0,1]$ be the step-function such that $W_G(x,y) =$

$\beta_{u,v}(G)$ if $x \in I_u$ and $y \in I_v$. This allows to define the *cut distance* between a graph and a graphon by $\delta_\square(G, W) = \delta_\square(W_G, W)$. [42] proves that the space of graphons empowered with metric $\delta_\square$ is compact.

The topology defined by the *cut distance* on the space of graphons has some interesting applications. [21] and[22] devolpes a notion of large deviation for Erdös-Rényi random graphs, i.e., random graphs $G(n, p)$ defined by constant graphons, which have been used to prove that a large class of exponential random graph models [51] are asymptotically equivalent to Erdös-Rényi random graphs.

1.4 Estimating graphons

It is great that such rich mathematical theory can be materialized into this simple limit object called graphon. And the fact that graphons are closely related to an intuitive class of random graph models $(G(n, w))$ gives the whole theory tremendous potential for applications. But estimating $G(n, w)$ from observed data is very difficult because of the highly non-parametric nature of w. In this section we discuss this problem under the perspective considered by this thesis, which involves approximating $G(n, w)$ using stochastic blockmodels.

Stochastic blockmodels are a particular class of $G(n, w)$ for which w has a step-function representation. Intuitively, it is like a random graph in which individuals of an heterogenous population are assigned to homogeneous subpopulations (blocks) in a way that connections happen independently with a probability that only depends on which block they belong to. It is a well studied model which has been explored in many variations and was largely applied in practice. How-

ever, to the best of our knowledge, this is the only stochastic blockmodels method that consistently estimates graphons and is at the same time computationally tractable.

Other attempts to develop stochastic blockmodels methods for estimating graphons are limited in scope either because they make unreasonable assumptions on the data or because they are computationally unfeasible. Patrick Wolfe's [64] uses likelihood methods to prove consistent estimation of graphons, but he doesn't provide a way for finding the blocks of the stochastic blockmodels, leaving the exponential-size space of partitions to be explored. Other approaches, such as [18], make the assumption that the vertex degree is enough to define the clusters, ignoring the fact that in practice graphs have vertex with similar degree but very different patterns of connection. It turns our that finding a reasonable and applicable solution is a hard challenge, and this problem is solved by the stochastic blockmodel approximation framework presented in the following chapters.

1.5 Stochastic blockmodel approximation

The stochastic blockmodel approximation (SBA) framework developed here gives a complete solution for the problem of estimating graphons. It is based on very parallelizable local computations that can be easily implemented in a distributed frameworks such as Map Reduce, offering great scalability. Because it consistently approximates graphons, our non-parametric approach can be applied to a general setting to find a stochastic blockmodel that preserves many properties of the estimated graph, such as density of subgraphs and other characteristics predicted by the theory of graph limits. In this section, we introduce the main results presented

in this thesis.

The SBA has two major steps, first it clusters vertices that are similar according to a *similarity distance* to find the blocks of the blockmodel, then it estimates the probabilities of connection between blocks averaging the edges connecting them. The clustering step is a greedy algorithm that tests if vertices are "similar" enough in order to be allowed in the same block, by "similar" we mean the *similarity distance* is smaller than a threshold parameter. We show how to choose this parameter in a way to guarantee a good balance between defining enough clusters to have a good step-function approximation of the graphon and having enough nodes in each block in order to estimate the probabilities of connection. We prove several theorems showing that this choice of parameters leads to an overall consistent method.

The heart of our algorithms is the computation of the similarity distances. The challenges of finding a metric of similarity using only local calculations brought us to find a solutions that requires at least two observations of $G(n, w)$. Even though this requirement doesn't stop the method from having some useful applications, it undesirably limits its scope. This problem led us to design a method that mimics a second observation of the edges in the graph by pairing every vertex i with a *twin* i' for which the curves $w(u_i, \cdot)$ and $w(u_{i'}, \cdot)$ should be very similar. This way we use the edges from i' as second observations of the edges from i, eliminating the unwanted assumption without compromising the consistency of the method.

We present some possible applications of the developed methodology. First we show two uses of SBA motivated by the graphon's theory: one is to compute the density any type of subgraph using the estimated stochastic blockmodel, and the

other is to compute the percolation threshold (the threshold in which in the large connected component of the network breaks down if you start deleting edges with given probability, see[11]). Second, we present a stochastic blockmodels methodology to optimally design experiments for identification of treatment responses with social interactions.

The thesis also discusses some possible variations of the models. The first variation allows analysis of undirected graphs. The second works with a "sparser" type of graphon in which the probability of connections are multiplied by a scaling factor. We give asymptotic bounds for which our model still works under the sparser scenario.

1.6 USVT

The only other alternative for estimating graphons that we know to be at the same time consistent and computationally feasible is Chatterjee's *universal singular value decomposition* (USVT) [20]. His work was developed in parallel with SBA but uses a very different strategy: his spectral approach de-noises the adjacency matrix of the graph to recover the probabilities of connection of the edges, while SBA approximates graphons with stochastic blockmodels. The method works great and requires just one singular value decomposition, but from our simulations it seems to underperform SBA in many scenarios.

1.7 Conclusion

We present a new non-parametric perspective for the analysis of massive networks. Our methodology, which offers consistent and scalable algorithms, adds a valuable inferential framework to the theory of dense graph limits. The theory defines a notion of convergence for sequences of graphs and predicts the existence of a limit object, called graphon, that is closely related to an intuitive but highly non-parametric class of random graph models. We propose a method for estimating graphons using stochastic blockmodels. Because the theory offers powerful results connecting intuitive ideas in a variety of settings, we believe that it has tremendous potential for applications. We hope the work presented in this dissertation will help to unlock this potential.

2

Stochastic blockmodel approximation

This chapter presents a fast non-parametric algorithm that uses stochastic blockmodels to consistently approximate graphons. Given a set of graphs observed from $G(n, w)$, we find a function $w' : [0,1]^2 \to [0,1]$ such that $w'(u_i, u_j)$ is close to $w(u_i, u_j)$ for any two nodes i and j. We prove that, with proper choice of parameters, the algorithm is consistent, i.e, the error in estimation vanishes with high probability as the size of the network increases. The algorithm, which works based

on local computations, seems to outperform existing models both in terms of time complexity and error estimation.

2.1 Setup and Algorithm

Suppose that $w : [0,1]^2 \to [0,1]$ is piecewise Lipschitz with Q blocks, i.e., there exist $\alpha_0 = 0 < \alpha_1 < \alpha_2 < \cdots < \alpha_Q$ and a constant $L > 0$ such that, for each $i, j \in \{1, 2, \ldots, Q\}$ and every $(x_1, y_1), (x_2, y_2) \in I_{ij} = (\alpha_{i-1}, \alpha_i) \times (\alpha_{j-1}, \alpha_j)$,

$$|w(x_1, y_2) - w(x_2, y_2)| \leq L(|x_1 - x_2| + |y_1 - y_2|).$$

Let $G(n, w)$ be the random graph with n vertices defined as follows: first sample $u_1, u_2, \ldots, u_n \sim$ Uniform$[0,1]$, i.i.d., then connect any two vertices i and j with probability $w(u_i, u_j)$. If $u = (u_1, u_2, \ldots, u_n)$ is given, define $G(n, w, u)$ similarly: it is the random graph with n vertices that assigns an edge between vertices i and j with probability $w(u_i, u_j)$. Suppose that $G_1, G_2, \ldots, G_T$ are $T \geq 2$ observations from $G(n, w, u)$, where $u = (u_1, u_2, \ldots, u_n)$ is a realization of $u_1, u_2, \ldots, u_n \sim$ Uniform$[0,1]$. Given $G_1, G_2, \ldots, G_T$, our goal is to find $w' : [0,1]^2 \to [0,1]$ such that $G(n, w')$ is a stochastic blockmodel that approximates $G(n, w)$, and $w'(u_i, u_j)$ is close to $w(u_i, u_j)$ for any pair of vertices i, j.

In a stochastic blockmodel, the existence of an edge between two vertices only depends on which blocks they belong to, i.e., two vertices in the same block use similar rule to connect to all other vertices in the graph. Since the edges from a vertex i are generated using the function $f_i(\cdot) = w(u_i, \cdot)$, in a stochastic blockmodel that approximates $G(n, w)$, two vertices i and j that belong to the same block

should have similar $f_i(\cdot)$ and $f_j(\cdot)$. The similarity between f_i and f_j can be measured by their mean squared difference:

$$d_{ij} = \int (f_i(x) - f_j(x))^2 dx. \tag{2.1}$$

We call d_{ij} the similarity distance between i and j, and we say that i and j are "similar" if d_{ij} is small.

Computing similarities is a fundamental piece of our algorithm, which can be described in two steps: first cluster vertices that are close with respect to the distance d_{ij}, then use the clusters to estimate the blockmodel $G(n, w')$. In the clustering step, an estimator $\hat{d}_{ij}$ computes d_{ij} by considering the connections from i and j to a randomly chosen subset of vertices S_{ij}, where the size S of S_{ij} is a fixed parameter of the model called the precision parameter. Using this notion of distance, the clustering scheme is designed to have the property that each cluster B_i has a pivot $b_i \in B_i$ which is at least Δ^2-close to any other vertex $v \in B_i$, i.e, $\hat{d}_{b_i v} < \Delta^2$ for any $v \in B_i$, where Δ is an accuracy parameter. The resulting clusters represent the blocks of the blockmodel, and the probability of connection between vertices in any two blocks A and B is estimated by averaging the number of edges from A to B. More detailed description of the algorithm is presented in the following sections.

2.1.1 Clustering

Let $G_1, G_2, \ldots, G_T$ be $T > 1$ graphs with common vertex set V observed from some $G(n, w, u)$, as in the above setting. The distance between two vertices i and j is

given by:

$$d_{ij} = \int_0^1 (f_i(x) - f_j(x))^2 dx = \int_0^1 f_i(x)^2 dx + \int_0^1 f_i(x)^2 dx - 2\int_0^1 f_i(x) f_j(x) dx. \quad (2.2)$$

Define

$$r_{ij} = \int_0^1 f_i(x) f_j(x) dx = \int_0^1 w(u_i, x) w(u_j, x) dx. \quad (2.3)$$

Then

$$d_{ij} = r_{ii} - r_{ij} - r_{ji} + r_{jj}$$

Take $k \in V$ such that $k \neq i$ and $k \neq j$, and let u_k be the position of k in the $[0, 1]$ interval. Consider the estimator

$$\hat{r}_{ij}^k = \left(\frac{1}{\left\lfloor \frac{T+1}{2} \right\rfloor} \sum_{1 \leq t_1 \leq \left\lfloor \frac{T+1}{2} \right\rfloor} G_{t_1}[i, k] \right) \left(\frac{1}{T - \left\lfloor \frac{T+1}{2} \right\rfloor} \sum_{\left\lfloor \frac{T+1}{2} \right\rfloor < t_2 \leq T} G_{t_2}[j, k] \right), \quad (2.4)$$

Since each $G_t[i, k]$ are independent observations from a Bernoulli$(w(u_i, u_k))$, and $u_k \sim$ Uniform$[0, 1]$,

$$\mathbb{E}[\hat{r}_{ij}^k | u_i, u_j, u_k] = w(u_i, u_k) w(u_j, u_k)$$

Integrating out u_k,

$$\mathbb{E}[\hat{r}_{ij}^k | u_i, u_j] = \int_0^1 w(u_i, u_k) w(u_j, u_k) du_k = r_{ij} \quad (2.5)$$

To estimate r_{ij}, the idea now is to sample at random a subset of vertices S_{ij} from

$V \setminus \{i,j\}$, whose size is given by the precision parameter S, and then to average $\hat{r}_{ij}^k$ over $k \in S_{ij}$:

$$\hat{r}_{ij} = \frac{1}{S} \sum_{k \in S_{i,j}} \hat{r}_{ij}^k. \tag{2.6}$$

The estimator for d_{ij} is defined by

$$\hat{d}_{ij} = \hat{r}_{ii} + \hat{r}_{jj} - \hat{r}_{ij} - \hat{r}_{ji}. \tag{2.7}$$

Using this estimator to compute similarities, the clustering algorithm is described as follows. Initially, there are no blocks, i.e., the set of pivots is empty. Then, vertices are sequentially assigned to blocks. If v is the first vertex in the sequence, a new block is created having v as pivot. Otherwise, the algorithm searches blocks b_v for which the distance between v and the pivot of b_v is at most Δ^2. If the distance between v and the closest pivot is less than Δ^2, v is assigned to the corresponding closest block. If no pivot is at distance less than Δ^2 from v, a new block is started with v as pivot. Algorithm 1 presents pseudocode for this procedure. The code runs in $O(T * S * K * n)$ steps, where T is the number of observed graphs, S is the size of the local subsets we use to estimate similarities, and K is the number of blocks in $G(n, w)$.

Input: A set of observed graphs $\{G_1, G_2, \ldots, G_T\}$ defined in a common vertex set V; accuracy parameter Δ; precision parameter S. We assume that $T \geq 2$, $\Delta > 0$ and $S \leq |V| - 2$.

Output: Cluster assignment representing the blocks in the blockmodel. The output is a vector indexed by elements $v \in V$ (let's call it *Block*) such that $Block[v] == b$ iff vertex v belongs to block b.

begin

  $NumberOfBlocks \longleftarrow 0$; `/* Number of blocks. */`

  $Pivot \longleftarrow \emptyset$; `/*` $Pivot[b] == i$ `if vertex` i `is pivot of block` b. `*/`

  $Block \longleftarrow \emptyset$; `/*` $Block[v] == b$ `if vertex` v `belongs to block` b. `*/`

  for $v \in V$ **do**

    if $NumberOfBlocks == 0$ **then**

      $NumberOfBlocks \longleftarrow NumberOfBlocks + 1$;

      $Pivot[NumberOfBlocks] \longleftarrow v$;

      $Block[v] = NumberOfBlocks$;

    else

      for $b \in$ `range(1,`$NumberOfBlocks$`)` **do**

        Sample set S_{vb} of size S uniformly from $V \backslash$\{v,b\} ;

        Compute $\hat{d}_{v,b}$ (given S_{vb}) from ;

      end

      $ClosestBlock = argmin_{b=1}^{NumberOfBlocks}(\hat{d}_{vPivot[b]})$;

      if $\hat{d}_{v,Pivot[ClosestBlock]} \leq \Delta^2$ **then**

        $Block[v] = ClosestBlock$;

      else

        $NumberOfBlocks \longleftarrow NumberOfBlocks + 1$;

        $Pivot[NumberOfBlocks] \longleftarrow v$;

        $Block[v] = NumberOfBlocks$;

      end

    end

  end

  return *Block*;

end

Algorithm 1: Clustering algorithm.

The choice of the L^2 norm to measure similarity

The similarity distance d_{ij} as defined above is the square of the L^2 distance between the curves $w(u_i, \cdot)$ and $w(u_j, \cdot)$. In this section we discuss why L^2. Would it be possible to use other measures, such as L^1 or L^p? If yes, how do they compare to L^2?

L^1 norm

A key property of our algorithm is fact that the similarity distance between two vertices i and j are computed using only local information, as they depend only on the connections between i, j and the other vertices . Would it be possible to design such an estimator to compute the L^1 distance $d_{ij}^1 = \int_0^1 |w(u_i, \cdot) - w(u_j, \cdot)|$? The answer is no, because the sufficient statistics describing the pair of vectors $G[i, \cdot]$ and $G[j, \cdot]$ are not enough to estimate d_{ij}^1. Since we ignore the order of the vertices, the information given by the pair of vectors can be summarized by:

$$\hat{s_1}(i,j) := \text{number of vertices } k \text{ such that } G[i,k] = G[j,k] = 0$$

$$\hat{s_2}(i,j) := \text{number of vertices } k \text{ such that } G[i,k] = G[j,k] = 1$$

$$\hat{s_3}(i,j) := \text{number of vertices } k \text{ such that } G[i,k] = 1, G[j,k] = 0$$

$$\hat{s_4}(i,j) := \text{number of vertices } k \text{ such that } G[i,k] = 0, G[j,k] = 1$$

Consider a graphon defined by

$$w(x,y) = \begin{cases} \frac{1}{2} & \text{if } x \geq \frac{1}{2} \text{ or } y \geq \frac{1}{2}; \\ 1 & \text{if } x \leq \frac{1}{4} \text{ or } y \leq \frac{1}{4}; \\ 0 & \text{otherwise.} \end{cases}$$

and let v_1, v_2, v_3 be three vertices such that $u_{v_1}, u_{v_2} > \frac{1}{2}$ and $u_{v_3} < \frac{1}{4}$. Then the expected value of the sufficient statistics s_1, s_2, s_3, s_4 for the pairs (v_1, v_2) and (v_1, v_2) are:

$$\begin{aligned} s1(v_1, v_2) = s1(v_1, v_3) = \\ s2(v_1, v_2) = s2(v_1, v_3) = \\ s3(v_1, v_2) = s3(v_1, v_3) = \\ s4(v_1, v_2) = s4(v_1, v_3) = 1/4 \end{aligned} \tag{2.8}$$

So, it is impossible to distinguish v_2 and v_3 when we compare their similarity to v_1 with respect to the way they connect to the other vertices. However, the L^1 distances are different, as $L^1(f1, f2) = 0$ and $L^1(f1, f3) = 1/2$. This counter example shows that the data is not enough to compute the similarity distances d_{ij}^1 using the same type of method we used in the L^2 case. So L^1 isn't an option.

L^p NORMS

Now suppose that we wish to use L^p, where p is an even number greater than 2. Then the intuition to define an estimator for $d_{ij}^p = \int_0^1 (w(u_i, \cdot) - w(u_j, \cdot))^p$ is similar to the case L^2 but it requires at least p samples of the network instead of 2. The p independent samples are necessary to compute the p-th powers of the expression $(w(u_i, \cdot) - w(u_j, \cdot))^p$. The approach works, but we wouldn't recommend for two

reasons: first, it requires more sampled graphs, which might not be available; second, estimating d_{ij}^p requires larger graphs. In this L^p setup we are trying to estimate $\int_0^1 (w(u_i,\cdot) - w(u_j,\cdot))^p$, which is a quantity much smaller then $\int_0^1 (w(u_i,\cdot) - w(u_j,\cdot)^2$ as w is bounded in $[0,1]$. Therefore, it is much harder to estimate d_{ij}^p than our original d_{ij}^2, as the error in estimation resulting from the sample size might become smaller than the values we are trying to estimate. So, we believe that there aren't many advantages of using L^p instead of L^2.

2.1.2 Histogram

The last step of the algorithm estimates the stochastic blockmodel $G(n, w')$, whose blocks are defined by the clusters $B_1, B_2, \ldots, B_K$ obtained in the clustering step. There are two types of parameters we need to compute to describe $G(n, w')$: the probability that a vertex belongs to a given block and the probability of connection between vertices in each pair of blocks. The probability of belonging to block B_I is estimated by

$$p_I = \frac{|B_I|}{n}, \tag{2.9}$$

and the probability of connection between elements of blocks B_I and B_J is

$$p_{IJ} = \frac{1}{|B_I||B_J|} \sum_{x_i \in B_I, y_j \in B_J} \frac{G_1[x_i, y_j] + G_2[x_i, y_j] + \ldots + G_T[x_i, y_j]}{T}. \tag{2.10}$$

One can get an explicit representation of the function $w : [0,1]^2 \to [0,1]$ by splitting the unit interval $[0,1]$ into subintervals Z_I of size p_I, $I \in \{1, \ldots, K\}$, and defining, $\forall u_i, u_j \in [0,1]$,

$$w(u_i, u_j) = p_{IJ}$$

where I and J are such that $u_i \in Z_I$ and $u_j \in Z_J$. Since graphons are invariant under any measure preserving transformation, the way $[0,1]$ splits into $Z_1 \cup Z_2 \ldots \cup Z_K$ doesn't change the graphon associated with w.

To specify the probability of connection between two vertices in the observed graphs, one can directly use the graphon estimated above. If $v_1, v_2, \ldots, v_n$ are the vertices, and $u_1, u_2, \ldots, u_n$ are their respective position in the interval $[0,1]$, the ground-truth probability of observing an edge between $v_i \in B_I$ and $v_j \in B_J$ is given by $w(u_i, u_j)$. Define the estimator $\hat{w}_{v_i v_j}$ for $w(u_i, u_j)$ by

$$\hat{w}_{v_i v_j} = p_{IJ}. \tag{2.11}$$

2.2 Results and Consequences

2.2.1 Estimating similarities

The distance d_{ij} is a measure of how similarly two vertices i and j connect to the other vertices. To compute the estimator $\hat{d}_{ij}$, we pick a randomly selected subset S_{ij} of $V \backslash \{i, j\}$, whose size is given by the precision parameter S, and study the connections from i and j to the vertices in S_{ij}. While increasing S clearly improves accuracy, one might want to bound this parameter to gain algorithmic efficiency. The follow theorem explains how S relates to the precision of $\hat{d}_{ij}$. A proof is given in the appendix.

Theorem 3. *The estimator $\hat{d}_{ij}$ for d_{ij} is unbiased and satisfies*

$$\mathbb{P}(|d_{ij} - \hat{d}_{ij}| > \epsilon) \leq 8e^{-\frac{S\epsilon^2}{\frac{256}{T-1} + \frac{8\epsilon}{3}}}, \tag{2.12}$$

for any $\epsilon > 0$,

From theorem 3 and because both d_{ij} and $\hat{d}_{ij}$ are bounded in $[0, 1]$,

$$\mathbb{E}[|d_{ij} - \hat{d}_{ij}|] \leq \epsilon * \mathbb{P}(|d_{ij} - \hat{d}_{ij}| \leq \epsilon) + |d_{ij} - \hat{d}_{ij}| * \mathbb{P}(|d_{ij} - \hat{d}_{ij}| > \epsilon) \leq \epsilon + 16e^{-\frac{S\epsilon^2}{\frac{256}{T-1} + \frac{8\epsilon}{3}}}$$

To see that this estimator is consistent, it is enough to make $\epsilon \in \omega(S^{-\frac{1}{2}}) \cap o(1)$.

2.2.2 Number of blocks

The performance of the algorithm is sensitive to the number and size of blocks it defines: in one hand the number of clusters needs to be large enough so the bias from the blockmodel approximation is small, on the other hand clusters have to be large enough to allow accurate estimation of the blockmodel parameters. This tradeoff can be controlled with a proper choice of the accuracy parameter Δ, as in general it is expectable that large Δ defines a small number of large clusters, while small Δ defines a large number of small clusters. The following theorem explains the relationship between Δ and the number of clusters in $G(n, w')$.

Theorem 4. *Let Δ be the accuracy parameter and S be the precision parameter used in the SBA algorithm. Then, the number K' of blocks in $G(n, w')$ the algorithm estimates*

satisfies

$$\mathbb{P}(K' > \frac{c_w}{\Delta}) \leq n^2 e^{-\frac{c_0 S \Delta^4}{\frac{1}{T-1}+\Delta^2}}, \tag{2.13}$$

where c_0 is a constant and c_w only depends on the graphon defined by w.

The proof is in the appendix.

2.2.3 Consistency

This section presents the main consistency result of the SBA algorithm: we prove that, with proper choice of parameters, the algorithm consistently estimates the probabilities of connection between any two vertices in the observed graphs. The error in estimation vanishes as $n \to \infty$ even if the number of observations doesn't increase, i.e., the additional information provided by increasing the number of vertices is enough to improve accuracy in estimation of all edges.

Let $v_1, \ldots, v_n$ be the vertices of the observed graph, and let $u_1, \ldots, u_n$ be their respective position in the interval $[0, 1]$. Consider the estimator $\hat{w}$ defined in equation (2.11), and define the error of estimation as

$$\text{Err}(\hat{w}) = \frac{1}{n^2} \sum_{i,j \in V} |w(u_i, u_j) - \hat{w}_{ij}| \tag{2.14}$$

We say that $\hat{w}$ is consistent if

$$\lim_{n \to \infty} \mathbb{E}\,[\,\text{Err}(\hat{w})] = 0 \tag{2.15}$$

As explained in sections 2.2.1 and 2.2.2, an appropriate choice of S and Δ is im-

portant to ensure good performance of the algorithm. The following theorem, for which we give a complete proof in appendix A, explains how the precision and accuracy parameters relate to $\mathrm{Err}(\hat{w})$.

Theorem 5. *a) If* $S \in \Theta(n)$ *and* $\Delta \in \omega\left(\left(\frac{\log(n)}{n}\right)^{\frac{1}{4}}\right) \cap o(1)$, *then*

$$\lim_{n\to\infty} \mathbb{E}\left[Err(\hat{w})\right] = 0.$$

b) There exists a constant c_0 *depending only on w such that, if* $S \in \Theta(n)$ *and* Δ *is constant, then*

$$\lim_{n\to\infty} \mathbb{E}\left[Err(\hat{w})\right] \leq c_0\sqrt{\Delta}.$$

2.2.4 Choosing accuracy parameter

Theorem 5 shows that, with appropriate choice of parameters, our estimator is consistent. For instance, if one chooses $S \in \Theta(n)$ and $\Delta = n^{-\frac{1}{5}}$, it is expected that the error of $\hat{w}$ approaches zero as $n \to \infty$. However, since these are asymptotic results, it is not clear how large n needs to be so the suggested setup for S and Δ leads to good performance. To account for practical situations, where the value of n might be small, we suggest a cross validation method of choosing Δ based on the observed data.

The parameter Δ relates to the accuracy of $\hat{w}$, as decreasing Δ reduces the bias of the estimator. However, if Δ is small, the algorithm might create a large number of clusters, some of them too small, what could compromise the estimation of the blockmodel parameters. We use a cross validation score from the theory of histogram estimation (see [62]) to study the risk of the algorithm provide a

clustering scheme that doesn't allow a good estimation of the parameters of the model. The idea is to test a range of values for Δ to choose the one giving smaller risk.

Let $\hat{B}_1, \ldots, \hat{B}_K$ be the blocks given by the clustering algorithm. For each block j, define $\hat{p}_j = \frac{|\hat{B}_j|}{n}$. The cross validation estimator of risk is computed as:

$$\hat{J} = \frac{2}{K(n-1)} - \frac{n+1}{k(n-1)} \sum_{j=1}^{m} \hat{p}_j^2. \tag{2.16}$$

2.2.5 Choosing precision parameter

From Theorem 3, it is clear that the parameter S is closely related to the precision of the similarity estimator $\hat{d}_{ij}$, in the sense that, the larger the S the best is the estimation. The greatest possible value for S is $n-2$, which is the size of $V \backslash \{i, j\}$, but, as Theorem 5 shows, any setup $S \in \Theta(n)$ is enough for consistency. A consequence of the result is that, if we assume that data is missing at random with some probability $p < 1$ per edge, we would still have asymptotic consistency. In that case, to compute $\hat{d}_{ij}$ one could just ignore missing edges and use only the observed data.

In case the size of the network is very large, if one is comfortable having some level of error, it might make sense to bound S to improve algorithmic efficiency. As we discussed in section 2.1.1, the algorithm runs in $O(T * S * K * n)$ steps, so bounding S could have a considerable positive effect in the complexity of the algorithm.

2.3 Variations

In this section we consider two variations of $G(n, w)$ for which SBA can be used to estimate the kernel function of the underlying random graph model: first, we assume w changes with n by a scaling factor, i.e., we consider the random graph model $G(n, w_n)$, where $w_n = \frac{1}{\rho_n} w$; second, we consider the asymmetric version of the problem, where w isn't necessarily symmetric and $G(n, w)$ is a random model for undirected graphs.

2.3.1 Sparsity

A limitation of the theory of graph limits developed in [41] is that it only works for dense graphs. In the definition of limit, graphs in a converging sequence should preserve the density of any subgraph, in particular, the overall probability of an edge should converge. The fact that graphons arise as the limit objects in this notion of convergence gives important evidence that the random graph model $G(n, w)$ is a good representation for very large networks. However, in this type of model, the degree of a node grows as $\Omega(n)$, but in many real world large network the average degree is small compared to n. So a reasonable variation of $G(n, w)$ is to assume that w changes with n by a scaling factor ρ_n. In this section, we consider observations of $G(n, w_n)$, where $w_n = \rho_n \nu$ and $\int_0^1 \int_0^1 \nu(x, y) dx dy = 1$. Our goal is to find an estimator $\hat{\nu}$ for ν and to show conditions for ρ_n that guarantee consistency of $\hat{\nu}$.

Let $G_1, G_2, \ldots, G_T$ be graphs observed from $G(n, w_n, u)$, for some $u \sim \text{Uniform}(0, 1)$. We estimate $\nu = \frac{w_n}{\rho}$, where $\rho_n = \int_0^1 \int_0^1 w_n(x, y) dx dy$ in two steps: first use SBA to

get an estimator $\hat{w}_n$ for w_n, then we divide $\hat{w}_n$ by an estimator of ρ_n.

The estimator for ρ_n, which is the probability of an edge, is straightforward:

$$\hat{\rho}_n = \frac{1}{Tn^2} \sum_{t=0}^{T} \sum_{i=1}^{n} \sum_{j=1}^{n} G_T[i,j]$$

Having applied the SBA algorithm on the data to get an estimator $\hat{w}_n$ for w_n, we define the estimator $\hat{\nu}$ for ν as

$$\hat{\nu} = \frac{\hat{w}_n}{\hat{\rho}_n}$$

The error of this estimator is defined as

$$\mathrm{Err}_\nu(\hat{\nu}) = \frac{1}{n^2} \sum_{i,j \in V} |\nu(u_i, u_j) - \hat{\nu}_{ij}| \tag{2.17}$$

The following theorem shows that, if $\rho_n \in \omega(\sqrt{\Delta})$, then $\hat{\nu}$ is a consistent estimator for ν. This means that, with proper choice of Δ, the algorithm works well if the overall probability of an edge is at least $\omega\left(\left(\frac{\log(n)}{n}\right)^{\frac{1}{8}}\right)$.

Theorem 6. *Let $S \in \Theta(n)$ and $\Delta \in \omega\left(\left(\frac{\log(n)}{n}\right)^{\frac{1}{4}}\right) \cap o(1)$. If $\rho_n \in \omega(\sqrt{\Delta})$, then*

$$\lim_{n \to \infty} \mathbb{E}\left[Err_\nu(\hat{\nu})\right] = 0.$$

A proof is presented in the appendix.

Sparsity, L^p graphons and scale free networks

Graphons, as originally defined in the theory of graph limits, represent probabilities of connection and are bounded in $[0, 1]$. In the case of sparse graphons, however, it might make sense to generalize this idea and let w to be unbounded: $w : [0, 1^2] \to \mathbb{R}$. In this case, the generative process of the random graph $G(n, w_n)$ defines the probability of connection between two vertices i and j as $\min(w_n(i, j), 1)$, where $w_n = \rho_n \nu$. Note that ρ_n and ν are not well defined as they can be rescaled by a factor η in a way to leave $w_n = (\rho_n \eta)(\frac{1}{\eta}\nu)$ constant. But the intuition behind this description is that ρ_n controls the sparsity of the generated graph as n increases, while $\nu : [0, 1]^2 \to \mathbb{R}$ describes structural properties of the graph. These structural properties represented by ν are generally true for any level of sparsity, but now as w_n might be greater than 1 there might be hidden structures that only happen in very sparse settings.

The ideas described above were first introduced in [12], and then generalized and further developed in [14] and [15]. In [12], Bollobás and Riordan define a notion of sparse graphons that allows w_n to be greater than 1 but requires ν to have a "bounded" density. It explains the intuition that two graphs with different densities might have similar structure, but is does that under a boundedness assumption which requires the random graph to have no specially dense spot, i.e., all regions of the graph have asymptotically the same density. This is an important limitation as many real world networks, such as the ones with scale free properties, have some areas that are intrinsically denser than the rest of the network and some nodes with potentially unbounded degrees as the graph grows. Borgs and co-authors introduces then in [14] and [15] the L^p graphons, a type of sparse

graphons that requires ν to be bounded in the L^p space but allows the function $\nu : [0,1]^2 \to \mathbb{R}$ to be unbounded. These unbounded regions are fundamentally denser than other parts of the graph. The papers develops a whole new theory of graph limits under this setting and even prove a generalization of the Szemerédi regularity theorem.

This generalized notion of sparse graphons define a random graph model with a generating process similar to the one defined by original graphons, as the probabilities of connection $w_n^* = \min(w_n(i,j), 1)$ are bounded in $[0,1]$ for any particular level of sparsity. Out stochastic blockmodel framework can be used to estimate w_n^*, but since w_n^* it is just an approximation of w_n we cannot guarantee that we will be able to estimate ν consistently under the L^p graphon settings. Certainly we will need to impose conditions on ρ_n, like the ones considered in theorem 6. To find these conditions, however, we need to relax an important assumption of our methodology: because of the unbounded regions we cannot require the graphon to be piecewise Lipschitz anymore. An idea is to approximate the L^p graphon to a piecewise Lipschitz graphon and let the Lipschitz constant increase with n. We then need to find a balance between how much sparsity we want to allow, i.e., how much we let ρ_n to decrease, and how dense we want certain areas of the graphon to be.

2.3.2 The directed networks case

The theory we developed so far considers undirected graphs generated from $G(n, w)$, where $w : [0,1]^2 \to [0,1]$ is a symmetric measurable function. One can easily extend the idea to directed graphs, by assuming that $w : [0,1]^2 \to [0,1]$

is not necessarily symmetric, and considering the following data generating process: first sample $u_i \sim \text{Uniform}[0,1]$ for each vertex i, then construct a directed edge from vertex i to vertex j with probability $w(u_i, u_j)$, for each pair (i,j). Let $G_d(n,w)$ be the directed random graph generated this way, and, if u is given, let $G_d(n,w,u)$ be the corresponding random graph conditional to u. In this section we discuss ways to adapt the SBA algorithm to the directed networks case. We leave simulations and deeper analysis of this method for a later paper, but we record the procedure here for future purposes.

The general idea of the modified method is to change the way we compute the similarity score by considering the similarity distances in both dimensions. The overall score of similarity is defined by the sum of similarities in both dimensions. In the first dimension, we use the same similarity score as before:

$$\hat{d}_{1ij} = \hat{r}_{1ii} + \hat{r}_{1jj} - \hat{r}_{1ij} - \hat{r}_{1ji}, \tag{2.18}$$

where

$$\hat{r}_{1ij} = \frac{1}{S} \sum_{k \in S_{i,j}} \hat{r}^{k}_{1ij}, \tag{2.19}$$

and

$$\hat{r}^{k}_{1ij} = \left(\frac{1}{\left\lfloor \frac{T+1}{2} \right\rfloor} \sum_{1 \le t_1 \le \left\lfloor \frac{T+1}{2} \right\rfloor} G_{t_1}[i,k] \right) \left(\frac{1}{T - \left\lfloor \frac{T+1}{2} \right\rfloor} \sum_{\left\lfloor \frac{T+1}{2} \right\rfloor < t_2 \le T} G_{t_2}[j,k] \right) \tag{2.20}$$

In the second dimension, we compute $\hat{r}_{ij}^{k}$ differently,

$$\hat{r}_{2ij}^{k} = \left(\frac{1}{\left\lfloor \frac{T+1}{2} \right\rfloor} \sum_{1 \leq t_1 \leq \left\lfloor \frac{T+1}{2} \right\rfloor} G_{t_1}[k,i] \right) \left(\frac{1}{T - \left\lfloor \frac{T+1}{2} \right\rfloor} \sum_{\left\lfloor \frac{T+1}{2} \right\rfloor < t_2 \leq T} G_{t_2}[k,j] \right). \quad (2.21)$$

The overall score of similarity is defined by

$$\hat{d}_{ij} = \hat{d}_{1ij} + \hat{d}_{2ij} \quad (2.22)$$

This distance is then used in the clustering step, and the histogram step proceeds as before.

2.4 Simulations

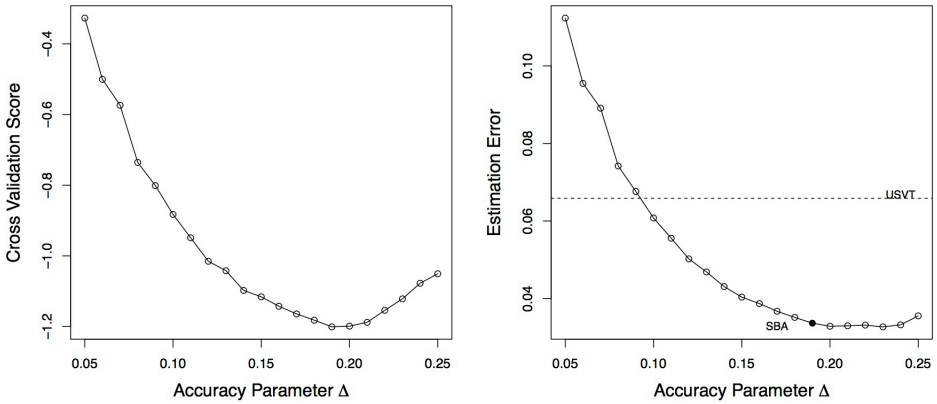
In this simulation study we use data generated from multiple samples of $G(n, w, u)$, where $u \sim$ Uniform[0,1] is assumed to be common for all samples and is called the ground truth position of the vertices. We test different values of functions $w : [0,1]^2 \to [0,1]$ to define the graphon, and the goal is to use $\hat{w}_{ij}$ to recover $w(u_i, u_j)$. The error is estimated using (2.14), and our model is compared with the universal singular value decomposition threshold (USVT), introduced in [20]. We show that our model is able to provide very good estimation of the considered graphons, and that the estimation gets better and better as the size of the network increases.

We use six types of data generating process. First, data is generated from a stochastic blockmodel with 10 blocks, each block having the same probability $\frac{1}{10}$. We randomly choose the probability of connection of each pair of blocks i and j by

sampling $p_{ij} \sim \text{Uniform}[0, 1]$. We also test another type of stochastic blockmodel, with two blocks in a core-periphery structure: the size of the core is sampled from $\text{Uniform}[0, 0.3]$; any two members in the core are connected with a common high probability sampled from $\text{Uniform}[0.8, 1]$; members in the periphery don't connect with each other, but they connect with the core with a fixed moderate probability sampled from $\text{Uniform}[0.3, 0.5]$. We then test other three different types of networks that can be generated by graphons: scale free, small world, and networks generated by a latent space model. For scale free network, we use the graphon $w(u_i, u_j) = \frac{1}{100\sqrt{u_i^* . u_j^*}}$, where $u_k^* = u_k$ if $u_k > \frac{1}{100}$, or $u_k^* = \frac{1}{100}$ otherwise (as a reference, [50] presents a deep analysis on similar types of graphons that define scale free networks). The small world networks are created with the following graphon: $w(u_i, u_j) = 0.9$ if $|u_i - u_j| < 0.05$ or $|u_i - u_j| > 0.95$, and $w(u_i, u_j) = 0.1$ otherwise. Finally, we execute the algorithm with data generated by the latent space model defined by the graphon $w(u_i, u_j) = \frac{1}{1+e^{50(-u_i-u_j+1)}}$. For each of these data generating processes, we run our model in 100 samples, where a sample is composed of 2, 4 or 8 observations of graphs generated from the same graphon.

The accuracy parameter Δ is chosen as described in section 2.2.4: run the algorithm in a range of values for Δ (in this case from 0.05 to 0.25, with increments of 0.01), and pick the Δ that gives the lowest cross validation score $\hat{J}$, defined in (2.16). Figure 2.1 compares the cross validation score and the ground truth estimation error for the different values of Δ, considering the stochastic blockmodels simulations with 10 blocks, on data composed by 2 observation of graphs with 250 nodes. Notice that the estimation error of the Δ defining the lowest cross validation score is fairly close to the minimum estimation error.

Figure 2.1: Cross validation score and estimation error



Tables 2.1 compares the estimation error from our model with the estimation error from USVT. For USVT, which is a model that estimates random matrices using a single observation, we take each of the 2, 4 or 8 observed graphs, and individually estimate the graphon. The final estimation is the average of the 2,4, or 8 estimated graphons. Notice that our model outperforms UVST in most cases.

Table 2.1: Simulations of SBA and USVT using 2,4,8 samples

Stochastic Blockmodel (10 blocks)									
Model	2 samples			4 samples			8 samples		
	250 vertices	500 vertices	1000 vertices	250 vertices	500 vertices	1000 vertices	250 vertices	500 vertices	1000 vertices
SBA	0.036±0.014	0.017±0.008	0.008±0.003	0.017± 0.009	0.009±0.003	0.006±0.002	0.012±0.008	0.008±0.010	0.004±0.002
USVT	0.066±0.005	0.047±0.004	0.033±0.002	0.055±0.006	0.037±0.004	0.026±0.003	0.047±0.007	0.030±0.005	0.020±0.003

Stochastic Blockmodel (Core-periphery)									
Model	2 samples			4 samples			8 samples		
	250 vertices	500 vertices	1000 vertices	250 vertices	500 vertices	1000 vertices	250 vertices	500 vertices	1000 vertices
SBA	0.003±0.002	0.001±0.001	0.001±0.001	0.002±0.001	0.001±0.001	0.001±0.001	0.002±0.001	0.001±0.001	0.001±0.001
USVT	0.019±0.023	0.013±0.021	0.013±0.031	0.016±0.014	0.010±0.017	0.008±0.018	0.012±0.011	0.007±0.009	0.005±0.008

Latent Space Model									
Model	2 samples			4 samples			8 samples		
	250 vertices	500 vertices	1000 vertices	250 vertices	500 vertices	1000 vertices	250 vertices	500 vertices	1000 vertices
SBA	0.029±0.007	0.021±0.004	0.016±0.001	0.028±0.014	0.017±0.007	0.012±0.000	0.024±0.015	0.013±0.007	0.011±0.008
USVT	0.049±0.005	0.031±0.001	0.023±0.002	0.049±0.004	0.031±0.001	0.023±0.002	0.052±0.013	0.031±0.001	0.023±0.002

Small World									
Model	2 samples			4 samples			8 samples		
	250 vertices	500 vertices	1000 vertices	250 vertices	500 vertices	1000 vertices	250 vertices	500 vertices	1000 vertices
SBA	0.053±0.004	0.039±0.004	0.031±0.001	0.047±0.006	0.036±0.009	0.026±0.008	0.045±0.007	0.037±0.011	0.029±0.012
USVT	0.094±0.004	0.074±0.002	0.060±0.002	0.085±0.004	0.067±0.002	0.055±0.002	0.079±0.004	0.064±0.002	0.051±0.002

Scale-Free									
Model	2 samples			4 samples			8 samples		
	250 vertices	500 vertices	1000 vertices	250 vertices	500 vertices	1000 vertices	250 vertices	500 vertices	1000 vertices
SBA	0.017±0.002	0.016±0.001	0.015±0.001	0.015±0.002	0.015±0.001	0.014±0.003	0.013±0.002	0.011±0.001	0.011±0.001
USVT	0.029±0.009	0.008±0.000	0.006±0.000	0.026±0.012	0.006±0.003	0.004±0.000	0.025±0.012	0.004±0.002	0.003±0.000

2.5 Applications

2.5.1 Counting subgraphs

A defining property that characterizes graphons as the limit of a sequence of graphs is the fact that they preserve density of subgraphs, in the sense that, if (G_n) converges, the density of any graph F as a subgraph of G_n converges to the density of F in the limiting graphon. Here we call density of F in G the number of adjacency preserving mappings from vertices of F to vertices of G, divided by the total number of mappings from F to G. In the graphon space, the density of F in w is defined by

$$t(F, w) = \int_{[0,1]^{|V(F)|}} \prod_{ij \in E(F)} w(x_i, x_j) dx.$$

This property suggests a useful application for our model. If the stochastic blockmodel approximation $\hat{w}$ is close enough to the graphon w, we might be able to estimate the density of a subgraph F in a set of graphs observed from w by computing $t(F, \hat{w})$. The densities of subgraphs are important structural properties of the graphs, and they are largely used by the machine learning community as parameters of random network models. These densities can be very hard to compute, but our algorithm gives an efficient solution for the problem: since the stochastic blockmodel $\hat{w}$ is parametric and finite, one can find an analytic formula for $t(F, \hat{w})$ as a function of a relatively small number of parameters.

Given a stochastic blockmodel $\hat{w}$ on K blocks $B_1 \ldots, B_k$, where the probability of a block B_i is p_i and the probability of connection between elements of blocks B_i

and B_j is m_{ij}, the density of any graph F can be computed by

$$t(F, \hat{w}) = \sum_{1 \leq v_1, v_2, \ldots v_n \leq K} \prod_{ij \in E(F)} m_{v_i v_j} \prod_{i=1}^{n} p_{v_i}$$

In table 2.2 we test this idea by estimating the density of 3, 5 and 10 cycles on the data described in section 3.3, comparing the ground truth density in the observed graphs with the density in the estimated stochastic blockmodel. As the table shows, the error in estimation is very small in all simulations.

Table 2.2: Frequency of subgraphs.

Data Generating Process		Frequency of subgraphs		
		3-cycles	5-cycles	10-cycle
Stochastic Blockmodel	Estimation	0.1370 ± 0.0339	0.0378 ± 0.0161	0.0017 ± 0.0017
10 blocks	Ground Truth	0.1370 ± 0.0339	0.0379 ± 0.0161	0.0017 ± 0.0017
Core-periphery	Estimation	0.0150 ± 0.0133	0.0014 ± 0.0017	0.0000 ± 0.0000
	Ground Truth	0.0150 ± 0.0133	0.0014 ± 0.0017	0.0000 ± 0.0000
Latent Space Model	Estimation	0.2460 ± 0.0192	0.1017 ± 0.0128	0.0106 ± 0.0027
	Ground Truth	0.2462 ± 0.0193	0.1019 ± 0.0128	0.0106 ± 0.0026
Small World	Estimation	0.0087 ± 0.0003	0.0002 ± 0.0000	0.0000 ± 0.0000
	Ground Truth	0.0091 ± 0.0001	0.0002 ± 0.0000	0.0000 ± 0.0000
Scale Free	Estimation	0.0001 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
	Ground Truth	0.0002 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000

2.5.2 Percolation threshold

Consider a very large network G, initially connected, and randomly delete some of its edges. Clearly, if the proportion of removed edges is small, the resulting network will likely have a giant connected component containing most of the vertices in G, but, as the number of deleted edges increases, the giant component will eventually break down into small pieces, until all vertices are disconnected. The problem of understanding the collapse of the connected components in G is called percolation, and one of the most striking results in percolation theory is the fact that many networks have a percolation threshold that defines a rapid transition between two very different qualitative states: if the probability of deleting an edge is smaller than the threshold, the network tends to have a giant component, but if this probability is larger than the threshold, the network tends to disintegrate into several small isolated components. This surprising result is very useful for applications, because it helps understand how robust to attacks networks are. In this section we explain how to use the SBA algorithm to estimate the percolation threshold of the observed graphs.

To be more specific about the percolation process we are considering, we assume that each edge in a graph G is kept with probability p and deleted with probability $1 - p$, for some $p \in [0, 1]$. Under this setting, the case of percolation in graphons is a direct consequence of percolation in inhomogeneous random graph (see [11] and [10] for reference). In an inhomogeneous random graph, the probability of connection between two vertices i and j is given by

$$p_{ij} = min\left\{\frac{k(u_i, u_j)}{n}, 1\right\},$$

where $k : [0,1]^2 \to \mathbb{R}^+$ is a kernel function and n is the size of the graph. In the context of graphons, we consider $k(u_i, u_j) = n \cdot w(u_i, u_j)$, so $p_{ij} = w(u_i, u_j)$.

Theorem 3.1 in [11] shows that the percolation threshold is a determined by the norm of a functional operator T_k defined by:

$$(T_k f)(x) = \int_{[0,1]} k(x,y) f(y) dy,$$

where f is any measurable function defined in $[0,1]$. The norm of T_k is given by

$$||T_k|| = \sup \{||T_k f||_2 : f \geq 0, ||f||_2 \leq 1\} \leq \infty.$$

where $||f||_2 = (\int_0^1 f(x)^2 dx)^{\frac{1}{2}}$. Expanding this expression using $T_k = n \cdot w$

$$||n.w|| = n. \sup \left\{ \left(\int_0^1 \int_0^1 \int_0^1 f(y_1) w(y_1, x) w(x, y_2) f(y_2) dx dy_1 dy_2 \right)^{\frac{1}{2}} : f \geq 0, ||f||_2 \leq 1 \right\}. \tag{2.23}$$

The follow theorem formalizes the notion of phase transition in this percolation scheme. It is a direct consequence of theorem 3.1 in [11], so we skip the proof in this paper.

Theorem 7. *Let $G_p(n,w)$ be a random graph with n vertices generated from the graphon w, followed by a percolation procedure that deletes edges with probability $1-p$, and keep edges with probability p, and call $C_1(G_p(n,w))$ the size of the largest connected component of $G_p(n,w)$. If $p < \frac{1}{||n \cdot w||}$, then $C_1(G_p(n,w)) = o(n)$, while if $p > \frac{1}{||n \cdot w||}$, then $C_1(G_p(n,w)) = \Theta(n)$ whp. We call $p_t = \frac{1}{||n \cdot w||}$ the percolation threshold of $G(n,w)$.*

The integral (2.23) is non-trivial to be computed for general w. But, by approximating the graphon with a stochastic blockmodel, we discretize the original model into a finite parametric form. This simple representation allow us to compute (2.23) very efficiently. The following theorem, for which a proof is given in the appendix, explains how to compute the expected percolation threshold of a stochastic blockmodel.

Proposition 8. *Let w be a stochastic blockmodel with B blocks defined by a probability matrix M, for the connections, and a probability vector p, for the blocks. Let M' be the matrix whose i, j entry is $p_i^{\frac{1}{2}} M_{ij} p_j^{\frac{1}{2}}$, and λ be the largest eigenvalue, in absolute value, of M'. Then the percolation threshold of theorem 7 is:*

$$p_t = \frac{1}{n.\lambda.} \tag{2.24}$$

Note that the percolation threshold is not an intrinsic property of the graphon w, but it is a property of the random graph model defined by w if we specify its size. So, when using the matching mechanism to run SBA on a single network, one should divide p_t by 2 in order to estimate the percolation threshold of the initial network.

Figure 2.2: Phase transition and percolation threshold for latent space model data.

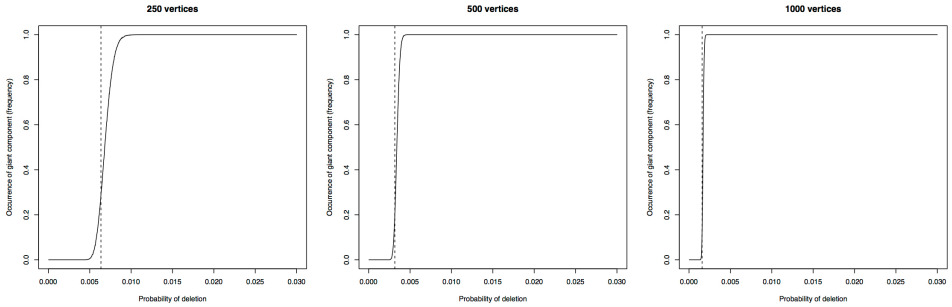


Figure 2.3: Phase transition and percolation threshold for small world data.

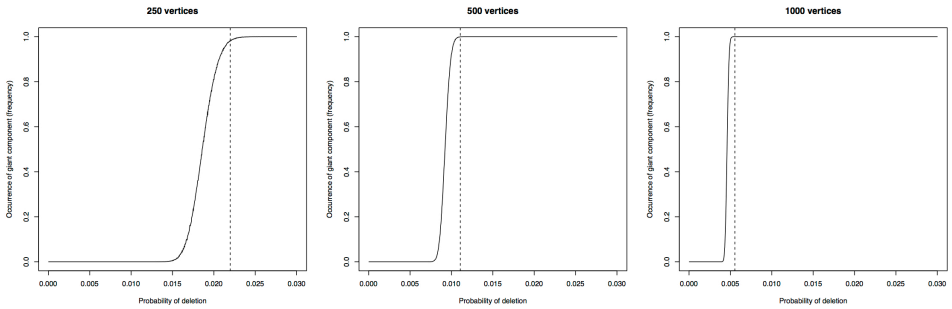
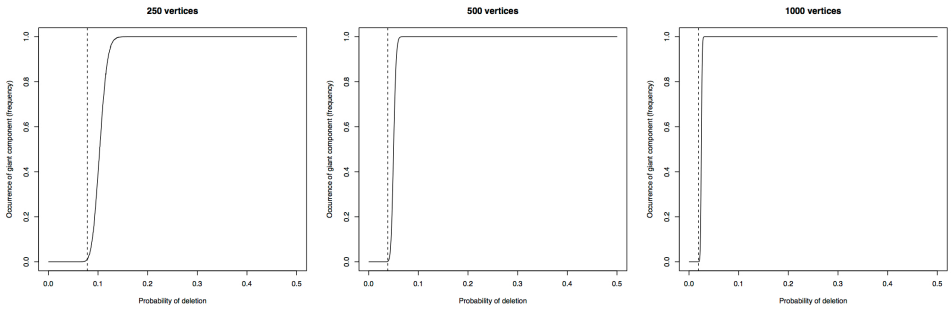


Figure 2.4: Phase transition and percolation threshold for scale-free data.



We test our results in networks generated from three types of non-blockmodel

graphons: scale free, small world and latent space models. For each type, we use networks of size 250, 500 and 1000. The idea is to compare our predicted percolation threshold p_t with empirical observations of the phase transition. To compute the threshold, we use the simulation output from section 2.4: for each graphon and each network size, we run SBA and use the estimated stochastic blockmodel to compute the p_t according with proposition 8. To observe the phase transition, we consider each type of simulation and generate 100 samples, each one with 100 observation of $G(n, w, u)$. We take the observed networks, and simulate percolation with probability p varying in a range of 1000 values. For each level or percolation, we take the proportion of graphs that end up with a large component, defining large component as any component of size at least $\frac{4n}{5}$. Results are presented figures 2.2, 2.3 and 2.4. The vertical line represents the estimated percolation threshold, and the curves show the rate of occurrence of a giant component.

3

Estimating vertex similarity from single observation

The setup of SBA requires more than one observation from $G(n, w, u)$, so the algorithm can compute the similarity distances d_{ij}. A second observation is necessary because, to estimate the terms $r_{ii} = \int_0^1 w(u_i, x)w(u_i, x)dx$ from equation (2.3), we need at least two independent samples from Benoulli$(w(u_i, u_k))$, for every u_k. In

this chapter, we suggest an extension of SBA that can be applied to single networks. The idea is to match "similar" vertices in a way that every vertex i is associated with a twin vertex i' such that $f_i(\cdot) = w(u_i, \cdot)$ is close to $f_{i'}(\cdot) = w(u_{i'}, \cdot)$, and then run SBA by assuming that twin edges e_{ij} and $e_{i'j}$ are multiple observations of the same $w(u_i, u_j)$. The matching mechanism not only relaxes the requirement of multiple observation, but does that offering consistent estimation of the graphon.

3.1 Matching procedure

To define the matching mechanism, we create a new notion of similarity that can be computed using a single observation. The idea behind the method comes from the fact that even though at least two samples are required for computing r_{ii}, when $i \neq j$ it is possible to estimate r_{ij} from a single graph using the expression:

$$\hat{r}_{ij} = \frac{1}{n-2} \sum_{k \neq i,j} G[i,k]G[j,k]. \tag{3.1}$$

Moreover, if the similarity distance between i_1 and i_2 is small, then one should expect that $r_{i_1k} = \int_0^1 f_{i_1}(x)f_k(x)dx$ is close to $r_{i_2k} = \int_0^1 f_{i_2}(x)f_k(x)dx$ for every vertex k, since d_{ij} is the square of the L_2 distance between $f_{i_1}(\cdot) = w(u_{i_1}, \cdot)$ and $f_{i_2}(\cdot) = w(u_{i_2}, \cdot)$. Therefore, one way to measure the similarity between two vertices i and j is to compare the values r_{ik} and r_{jk}, for $k \neq i, j$.

This intuition can be generalized as follows. Associate to every vertex i a transformation $\mathcal{F}_i : \mathbb{F} \to [0,1]$, from the space of functions $\mathbb{F} = \{g : [0,1] \to [0,1]\}$ to the unit interval, defined by $\mathcal{F}_i(g) = \int_0^1 f_i(x)g(x)dx$. Then d_{ij} can be derived from $\mathcal{F}_i$ and $\mathcal{F}_j$ as $d_{ij} = \mathcal{F}_i(f_i) + \mathcal{F}_j(f_j) - \mathcal{F}_i(f_j) - \mathcal{F}_j(f_i)$. Note that two vertices i and j

are such that $f_i(\cdot) = f_j(\cdot)$ if and ony if $\mathcal{F}_i(g) = \mathcal{F}_j(g)$ for every g.

These ideas are used to define a notion of distance which is the basis of the matching mechanism. For any two vertices i and j, define the "matching" distance m_{ij} by

$$m_{ij} = \frac{1}{n-2} \sum_{k \neq i,j} |r_{ik} - r_{jk}| = \frac{1}{n-2} \sum_{k \neq i,j} |\mathcal{F}_i(f_k) - \mathcal{F}_j(f_k)|, \tag{3.2}$$

and the estimator $\hat{m}_{ij}$ by

$$\hat{m}_{ij} = \frac{1}{n-2} \sum_{k \neq i,j} |\hat{r}_{ik} - \hat{r}_{jk}|. \tag{3.3}$$

The matching algorithm is described by the steps:

1. Randomly choose a vertex i that doesn't have a twin yet.
2. Find a vertex i' that is closest to i according to the matching distance. The vertex i' is called the twin of i, and it is randomly chosed among all vertices with smallest positive distance $\hat{m}_{ii'}$, including the ones that already have a twin.
3. Repeat the process until all vertices have a twin.

Note that the twin relation is not symmetric, in the sense that if i' is the twin of i, then the twin of i' could be a vertex i'' different from i.

3.2 Consistency

The method described above, which, for every vertex i, finds another vertex i' whose edges mimic a second observation of the edges from i, lets SBA run on a single graph, as the algorithm assumes that the edges from i and the edges from i' are generated using the same $w(u_i, \cdot)$. This assumption, however, might just be an approximation, since i and i' are not in the same position, i.e., $u_i \neq u_{i'}$. In this section, we study how this approximation affects the performance of the algorithm, and we show that, as the size of the network increases, the twin couples obtained in the matching process are arbitrarily similar, what makes this extended version of SBA to be consistent.

Let's start by defining r_ij and $\hat{r}_ij$ as in section 2.1.1:

$$r_{ij} = \int_0^1 w(u_i, x)w(u_j, x)dx. \tag{3.4}$$

and

$$\hat{r}_{ij}^k = G[v_i, v_k]G[v_j, v_k]$$

Since $\mathbb{E}(G[x, y]) = w(u_x, u_y)$,

$$\mathbb{E}[\hat{r}_{ij}^k | u_{v_i}, u_{v_j}, u_{v_k}] = w(u_{v_i}, u_{v_k})w(u_{v_j}, u_{v_k}) \tag{3.5}$$

Assuming that $u_{v_k} \sim \text{Uniform}[0, 1]$, the above expression implies that every $\hat{r}_{ij}^k$ is

an unbiased estimator for r_{ij}:

$$\mathbb{E}[\hat{r}_{ij}^{k}|u_{v_i}, u_{v_j}] = \int w(u_{v_i}, x)w(u_{v_j}, x)dx \tag{3.6}$$

Thus, from the Hoeffding inequality,

$$\mathbb{P}\left(|\hat{r}_{ij} - r_{ij}| > \epsilon|u_{v_i}, u_{v_j}\right) \leq 2e^{-2S\epsilon^2}$$

Integrating out the $u's$ and considering the union over i, j

$$\mathbb{P}\left(\max_{ij}|\hat{r}_{ij} - r_{ij}| > \epsilon\right) \leq 2n^2e^{-2S\epsilon^2} \tag{3.7}$$

Suppose the matching mechanism generates the pairs $(v_1, v_1'), \ldots, (v_n, v_n')$, and let $\xi_n = \max_{i,k\in\{1,2,\ldots,n\}} |w(u_{v_i}, u_{v_k}) - w(u_{v_i'}, u_{v_k})|$. For each i, j, k let

$$r_{ij}^{*} = \int w(u_{v_i}, x)w(u_{v_j'}, x)dx$$

and consider the estimators:

$$\hat{r}_{ij'}^{k} = G[v_i, v_k]G[v_j', v_k],$$

$$\hat{r}_{ij}^{*} = \frac{1}{n-2}\sum_{k\neq i,j}\hat{r}_{ij'}^{k},$$

and

$$\hat{d}_{ij}^{*} = \hat{r}_{ii}^{*} + \hat{r}_{jj}^{*} - \hat{r}_{ij}^{*} - \hat{r}_{ji}^{*},$$

Theorem 9, which has a proof in the appendix, shows how $\hat{d}_{ij}^*$ relates to ξ_n.

Theorem 9. *The estimator $\hat{d}_{ij}^*$ satisfies*

$$\mathbb{P}(|d_{ij} - \hat{d}_{ij}^*| > \epsilon + 4\xi_n) \leq 8n^2 e^{-\frac{S\epsilon^2}{8}}, \tag{3.8}$$

for any $\epsilon > 0$,

The following theorem 10 is the equivalent to theorem 4 for this noisy scenario. A proof is presented in the appendix.

Theorem 10. *Consider a single observation G of $G(n, w)$. Apply the matching procedure defined above and run SBA using accuracy parameter $\Delta > \sqrt{2\xi_n}$ and precision parameter S. Then, the estimated number of blocks satisfies*

$$\mathbb{P}\left(K^* > \frac{c_w}{\sqrt{\Delta^2 - 8\xi_n}}\right) \leq 2n^4 e^{-\frac{S\Delta^4}{32}}, \tag{3.9}$$

where c_w is a constant that only depends on the graphon defined by w.

Note that theorem 10 requires $\Delta > \sqrt{8\xi_n}$. This is related to the fact that, if the bias of the estimator $\hat{d}_{ij}^*$ is too large compared with Δ^2, we won't be able to decide if the two vertices are similar enough so they belong to the same block. This means the the accuracy parameter Δ defines how much bias it is possible to accept from adding noise to the computation of similarity distances.

The following theorem shows that, with the matching procedure defined here, ξ_n vanishes as n increases.

Theorem 11. *Suppose that the matching procedure generates the following pairs*

of twins $(v_1, v_1'), (v_2, v_2'), \ldots, (v_n, v_n')$*, and let* $\xi_n = \max_{i,k\in\{1,2,\ldots,n\}} |w(u_{v_i}, u_{v_k}) - w(u_{v_i'}, u_{v_k})|$*. Then, for any* $\zeta \in (0,1)$

$$\mathbb{P}\left(\xi_n > (64L)^{\frac{1}{4}} n^{-\frac{1-\zeta}{8}}\right) \leq$$

$$4n^5 e^{-8\frac{n-2}{n}n^\zeta} + n^2 e^{-(n-1)^\zeta} + 8L^{1/2} n^{-\frac{3+\zeta}{4}}$$

In particular, for any $\epsilon > 0$

$$\lim_{n\to\infty} \mathbb{P}(\xi_n > \epsilon) = 0. \tag{3.10}$$

A full proof for this result if provided in the appendix, and it is based on four observations:

1. $\hat{m}_{ii'}$ is an unbiased estimator for $m_{ii'}$.
2. As n increases, the matching distance $m_{ii'}$ between any vertex i and its twin i' decreases.
3. Small matching distance $m_{ii'}$ implies small similarity distance $d_{ii'}$.
4. Small similarity distance between vertices and its twins implies small ξ_n.

This theorem, together with theorem 9, proves that the bias of $\hat{d}_{ij}^*$ vanishes as $n \to \infty$. As a result, we have consistency of the entire process.

Theorem 12. *Given a network with n vertices sampled from* $G(n, w)$*, apply the matching procedure and run the SBA algorithm using parameters S and* Δ *to find an estimator* $\hat{w}$ *for* w *.*

a) If $S \in \Theta(n)$ *and* $\Delta \in \omega\left(n^{-\frac{1}{16}}\right) \cap o(1)$*, then*

$$\lim_{n' \to \infty} \mathbb{E}\left[Err(\hat{w})\right] = 0.$$

b) There exists a constant c_0 *depending only on w such that, if* $S \in \Theta(n)$ *and* Δ *is constant, then*

$$\lim_{n \to \infty} \mathbb{E}\left[Err(\hat{w})\right] \leq c_0(\Delta^2 + \xi_n)^{\frac{1}{4}}.$$

We sketch the proof of this theorem in the appendix.

3.3 One sample vs. two samples

The procedure developed in this chapter adapts the original SBA algorithm, which requires two sampled graphs, so it can run using only one sample. But what if we have the choice of either observing a single graph of size n or two graphs of size $\frac{n}{2}$ (for an experiment in which de budget depends on the number of people in the sample)? Of maybe one graph of size n vs. two of size $\frac{n}{\sqrt{2}}$ (for experiments in which the cost is defined by the number of edges in the sample)?

Theorem 12.b) states that in case only one graph is observed.

$$\lim_{n \to \infty} \mathbb{E}\left[\mathrm{Err}(\hat{w})\right] \leq c_0(\Delta^2 + \xi_n)^{\frac{1}{4}}, \tag{3.11}$$

where $\xi_n = \max_{i,k \in \{1,2,\ldots,n\}} |w(u_{v_i}, u_{v_k}) - w(u_{v'_i}, u_{v_k})|$.

The two graphs observation is equivalent to $\xi_n = 0$.

$$\lim_{n\to\infty} \mathbb{E}\left[\mathrm{Err}(\hat{w})\right] \leq c_0(\Delta^2)^{\frac{1}{4}}, \tag{3.12}$$

As $\Delta \to 0$ the estimation is consistent as stated by theorem 5.

Asymptotically, the one sample observation is clearly worse or at most equivalent to the two samples option, as suggested by equations (3.11) and (3.12). From theorem 5 we notice the two sample case allows Δ to decrease with rate $\omega\left(\left(\frac{\log(n)}{n}\right)^{\frac{1}{4}}\right)$, which faster than the required by theorem 12 for the one sample case, which needs $\Delta \in \omega\left(n^{-\frac{1}{16}}\right)$. Despite the asymptotical disadvantage, however, it might be prudent to choose the one sample procedure as it works with weaker assumptions, since the two samples method relies on the fact that the two observation are sampled without much noise. Unfortunately, this assumption is unreasonable for many real world applications, where it is hard to even guarantee that the observations are being sampled from the same (or even a similar) graphon. Suppose for instance that the second sample is not exactly from the same $G(n, w, u)$, but from $G(n, w, u')$, where u' is a perturbation of u such that $|w(u_{v_i}, u_{v_k}) - w(u'_{v_i}, u_{v_k})|$ are not exactly zero. Essentially suppose that $\xi'_n = \max_{i,k\in\{1,2,\ldots,n\}} |w(u_{v_i}, u_{v_k}) - w(u'_{v_i}, u_{v_k})| > 0$. As observed in theorem 10, our consistency results require $\Delta > \sqrt{8\xi'_n}$. This happens because, if Δ is much smaller than $\sqrt{\xi'}$, the error in estimation of the similarity distances would be larger than the accuracy parameter (see theorem 9), what could badly affect the whole clustering process, since in our algorithm the parameter Δ works as a threshold that defines the maximum similarity distances allowed between elements of a cluster. If the error is too large, very distinct vertices could pass the similarity threshold.

In fact, statements of theorem 10 regarding the number of clusters generated by SBA are not true if $\Delta < \sqrt{8\xi'_n}$, and so in that case the procedure might not be consistent. Therefore the noise in the second sample limits the decrease of the accuracy parameter.

In the case of a single graph being transformed in two observations using the procedure described in this chapter, we show that as n increases it is always possible to find, for each vertex v_i, another vertex v'_i inside the graph which is close enough to v_i, i.e., the noise becomes smaller and smaller guaranteeing to achieve consistency. So, in practice it is recommendable to work with a single observation using the full SBA procedure.

3.4 Simulations

In this section, we simulate the setup of section 2.4 to test the idea of running SBA on a single graph G with n vertices by using the matching mechanism to duplicate the edges of G. We use the same graphons as in section 2.4: stochastic blockmodels, core-periphery, latent space, small world, and scale free. For each type of dataset we generate 100 networks with 1000 vertices, and compare the performance of SBA and USVT. Results are shown on table 3.1.

We also compare the matching mechanism with two other ways of applying SBA. First, instead of using a single graph with 1000 vertices and $1mi$ possible edges, we consider two graphs with 707 vertices each and a total number of edges of $2 \cdot 707^2 \sim 1mi$. This accounts for the practical situations where the cost of running the experiment depends only on the number of edges in the graph, and so we

have to make a decision between observing a single large graph or observing two smaller graphs with the same total number of edges. We run these simulation by generating 100 samples of $G(707, w, u)$ with 2 observations each and then applying SBA. The second scenario we consider accounts for the case where the cost of the experiment depends on the number of time a vertex is observed. So, instead of observing 1000 vertices a single time, we consider that 500 are observed twice. For this case, we generate 100 samples of $G(500, w, u)$ with 2 observations each. Results are shown on table 3.1.

We notice that the SBA outperforms USVT in most cases. The idea of using the matching mechanism to apply SBA to a single network, not only makes the method more applicable to real situation, where more than one sample might not be available, but it also shows to be almost equivalent to the two scenarios we assume it is possible to make multiple observations.

Table 3.1: 1 sample of size 500 transformed into 4 samples of size 250

Simulation Method **Data**	**matching+SBA** 1 obs, N=1000	**SBA** 2 obs, N=500	**SBA** 2 obs, N=707	**USVT** 1 obs, N=1000
Stochastic Blockmodel	0.016 ± 0.004	0.017 ± 0.008	0.011 ± 0.004	0.044 ± 0.002
Core-periphery	0.004 ± 0.004	0.001 ± 0.001	0.001 ± 0.001	0.017 ± 0.048
Latent Space Model	0.017 ± 0.003	0.021 ± 0.003	0.018 ± 0.001	0.023 ± 0.002
Small World	0.043 ± 0.007	0.039 ± 0.004	0.034 ± 0.003	0.068 ± 0.001
Scale Free	0.013 ± 0.001	0.016 ± 0.001	0.015 ± 0.001	0.008 ± 0.000

3.5 Conclusion

This is a non-parametric method of estimating graphons based on stochastic blockmodel approximation. The algorithm, which in its initial form requires at least two samples of graphs observed from the random model defined by the graphon, works by defining a similarity score between vertices, then clustering similar vertices to estimate the blocks of the stochastic blockmodel. We present a method of applying SBA to single observation by using a matching mechanism to obtain a second observation of each edge in the graph. The theory of graph limits guarantees that, if our estimation is good enough, the resulting stochastic blockmodel preserves many properties of the original graphon, such as the density of subgraphs. We prove that, with proper choice of parameters, our estimator is consistent, i.e., the error in estimation vanishes with high probability as the size of the network increases. Our applications confirm that the method can be used to efficiently compute network properties such as density of subgraphs and percolation threshold.

4

A stochastic blockmodels framework for the analysis of treatment response with social interaction

In this chapter we use the stochastic blockmodels approximation framework to develop a methodology for assessment of social interaction in treatment response,

assuming that the response of each individual is influenced by the treatment given in his social neighborhood. We show how information provided by the SBA about the structure of the network can be used to improve identifiability and to optimize estimation of social effect. Identifiability of treatment response is determined by the space of realized effective treatments, so in order to design optimal assignments it is necessary to describe this space. Understanding the space of realized effective treatment associated with a given treatment strategy can be a difficult challenge, given the complex ways the individuals' reference groups usually intersect. We propose a method to design treatment assignments that uses connection patterns given by the SBA to control the formation of effective treatments, potentially increasing the identification power of the experiment and improving estimation of model parameters. We apply our ideas to develop a methodology of experimental design for optimal estimation of treatment and social effects in linear models.

4.1 Background

Rubin's potential outcomes model for causal inference has been used by social scientists as the main framework for learning about the relationship between counterfactuals of interests and estimated conditional probabilities [54, 55]. The model works under the stable-unit-treatment-value-assumption (SUTVA), which doesn't allow interference between units. If SUTVA is satisfied, random assignment is believed to balance observed and unobserved confounders across treatment and control groups in a way to permit unbiased causal inference. In practice, however, response to a treatment is often influenced by social effects, and to account for

these effects it is necessary to model social interactions. In this section, we review the social science literature, in particular the work by Manski introduced in [46], which builds a theoretical framework for analyzing the impact of social interaction on identifiability under various model assumptions.

We start with a few definitions. Let J be a population of size n, T the space of potential treatments and Y the set of possible outcomes. Suppose that every $j \in J$ has a response function $y_j(\cdot) : T^J \to Y$ mapping a vector of potential treatments $t^J \in T^J \equiv \times_{k \in J} T$ into an outcome $y_j(t^J)$, so a person's outcome varies not only with his own treatment but also with the treatment given to other members of J. For each $j \in J$ we observe a realized treatment $z_j \in T$ and a realized outcome $y_j \in Y$. Given a treatment t^J and a subset $K \subset J$, we call $y^K(t^J)$ the vector indexed by elements of K whose k-th position is $y_k(t^J)$. We also let $\{y^K(t^J)\}$ be the distribution of elements in the vector $y^K(t^J)$, i.e., it is a probability measure on Y that assigns for each $y_o \in Y$ a weight proportional to the number of times y_o appears in the vector $y^K(t^J)$. Note that $\{y^J(z^J)\}$ is the distribution of observed outcomes across the population.

A common question in treatment response is: given empirical data $(z^J, y^J(z^J))$, is it possible to identify the distribution of outcomes for other potential treatments t^J? Manski approaches this problem by analyzing how various assumptions on the shape of $y_i(\cdot)$ and on the joint distribution of $(z^J, y^J(z^J))$ affects the identification of $\{y^J(t^J)\}$ given experimental data. In the general treatment response with social interaction setup, the identification region of $\{y^J(t^J)\}$ is described as

$$H[\{y^J(t^J)\}] \equiv [\{y^J(z^J)\} \cdot 1(z^J = t^J) + \delta \cdot 1(z^J \neq t^J), \delta \in \Delta_Y], \qquad (4.1)$$

where Δ_Y denotes the space of all probability measures on Y. Notice that if no restriction is made on $Y(\cdot)$, this region is degenerate: it is a singleton when $z^J = t^J$ and the entire Δ_Y when $z^J \neq t^J$. Under these settings, the identification power of $H[\{y^J(t^J)\}]$ is very low, but if we assume certain conditions on y it is possible to identify $\{y^J(t^J)\}$ for values $t^J \neq z^J$. These conditions are described by Manski in [46], and we review his results as follows. Here, knowledge about the sampling process, the topology of the network or its generating process doesn't help to estimate $\{y^J(t^J)\}$ because we have no information about the shape of y on t^J.

Under the above assumptions, the identification power of $H[\{y^J(t^J)\}]$ is very low, but if we impose certain conditions on y it is possible to identify $\{y^J(t^J)\}$ for values $t^J \neq z^J$. These conditions are described by Manski in [46], and we review his results as follows.

Individualistic treatment response (ITR)

Assumption that a person's outcome varies only with his own treatment, not being influenced by other members of the population. It is equivalent to Rubin's SUTVA. Under ITR, the identification region becomes:

$$H[\{y^J(t^J)\}] \equiv \left[\{y^{J(z=t)}(z^J)\} \cdot \frac{|J(z=t)|}{|J|} + \delta \cdot \frac{|J(z \neq t)|}{|J|}, \delta \in \Delta_Y \right], \qquad (4.2)$$

where $J(z = t) = \{j \in J | z_j = t_j\}$ and $J(z \neq t) = \{j \in J | z_j \neq t_j\}$. Here sampled data reveals the distributions $P(z)$ and $\{y^{J(z=t)}(t^J)\}$, but it is not informative about missing treatments $y^{J(z \neq t)}(t^J)$.

This is equivalent to standard Rubin's SUTVA assumption, in which the topol-

ogy of the network is ignored and $\{y^J(t^J)\}$ only depends on the shape of y and on the sampling process.

Constant treatment response (CTR)

Assumption that a person's outcome remains constant when t^J varies within specified subsets of T^J. These subsets, called the person's *effective treatments*, can be represented by a function $c_j(\cdot) : T^J \to C_j$. Under this notation, CTR is equivalent to

$$c_j(t^J) = c_j(s^J) \Rightarrow y_j(t^J) = y_j(s^J). \tag{4.3}$$

Using the Law of Total Probability and the above equation, the identification region becomes:

$$\begin{aligned} H[\{y^J(t^J)\}] \equiv \Bigg[\{y^{J(c(z^J)=c(t^J))}(z^J)\} \cdot \frac{|J(c(z^J) = c(t^J))|}{|J|} + \\ \delta \cdot \frac{|J(c(z^J) \neq c(t^J))|}{|J|}, \delta \in \Delta_Y \Bigg], \end{aligned} \tag{4.4}$$

where $J(c(z^J) = c(t^J)) = \{j \in J | c(z_j^J) = c(t_j^J)\}$ and $J(c(z^J) \neq c(t^J)) = \{j \in J | c(z_j^J) \neq c(t_j^J)\}$.

The effective treatment function c doesn't need to be defined in the context of a network. For instance, one could let y_i to be a function on the number of people receiving the same treatment as i. In this toy case, t^J would fall in the same effective treatment as z^J if for each individual i the number of people receiving treatment t_i in t^J is the same as the number of people receiving treatment z_i in z^J. The effective treatment of the population would be given by the vector $c(z^J)$ defined with entry $c_i(z^J) = |\{j \in J : c_j(z^J) = c_i(z^J)\}|$ for each i. With proper sampling strategy,

the larger the number of people j receiving the same effective treatment as i, the best we will be able to estimate the distribution of the response function in the subpopulation with effective treatment $c_i(z^J)$. On the other hand, estimation of $P(y(t^J))$ for $t^J \neq z^J$ is limited to the $t^{J\prime}s$ with overall effective treatment vector $c(z^J)$.

Interactions with reference groups (IRG)

This is a particular case of CTR where each member j of the population has a known reference group $G(j) \in J$. The outcome of person j is assumed to depend only on the treatment received by the members of his reference group, i.e., y_j depends on $t^{G(j)} = [t_k, k \in G(j)]$. IRG can be described using CTR notation by letting $C_j = T^{G(j)}$ and $c_j(t^J) = t^{G(j)}$ for any $j \in J$ and $t^J \in T^J$.

In our notation, we assume that each person j always belongs to his own reference group $t^{G(j)}$ and we separate $G(j) = \{j\} \cup N_j$, where $N_j = G(j)/\{j\}$ is called the neighborhood of j. Note that the set of reference groups can be described by a directed graph G, where each person $j \in J$ is represented by a node with value t_j, and an edge $G[i, j]$ exists if and only if j is in the neighborhood N_i. The effective treatment person i is exposed to depends on the treatment given to his reference group, so two treatment strategies z^J and t^J have similar effective treatment if for each i the treatment in z^{N_i} is similar to the treatment in t^{N_i}.

Anonymous interactions (AI)

IRG can be strengthened with the assumption that interactions are anonymous, i.e., the outcome of an individual is invariant under permutation of the treatments received by members of his neighborhood. In this case, the outcome depends only on the person's treatment, the size and the distribution of treatments describing the neighborhood. Here, the effective treatment of individual i is $c_i(z^J) = (z_i, |N_i|, p_i)$, where p_i is the distribution of treatment in node i. Clearly, this treatment response setting depends a lot on the network topology, and given the complex connection patterns one might find in a network, it is generally not trivial to understand how does the space of realized effective treatments look like in a general network.

To build intuition on how the network topology affects the realization of effective treatments, let's consider a toy example of a simplistic core-periphery network generating process. Assume that the population is composed by three disjoint subgroups: J_A, the national core, is 10% of the population and has the most popular individuals, each one being connected to any other individual with probability 80%; J_B is the local core, it has 20% of the population and its elements are connected to members of $J \backslash J_A$ with probability 40%; J_C is the periphery, it contains 70% of the population and each individual is connected within J_C with probability 10%. This is a stochastic blockmodel with three blocks of sizes 0.1, 0.2 and 0.7 and probability matrix

$$\begin{pmatrix} 0.8 & 0.8 & 0.8 \\ 0.8 & 0.4 & 0.4 \\ 0.8 & 0.4 & 0.1 \end{pmatrix}.$$

Also, assume that the treatment space is binary, meaning that a person can only be treated or not treated, and that people are treated at random with probability ρ. Note that people in block J_A, which are connected with probability 80% to the rest o the population, have an expected number of friends $0.8 * |J|$ and its neighborhood has an expected number of treatment assignments $0.8 * \rho * |J|$. Similarly, the expected number of people treated in neighborhoods of elements in J_B and J_C are, respectively, $0.44 * \rho * |J|$ and $0.23 * \rho * |J|$. So, the distribution of effective treatments is concentrated in three modes, which are the expected values for elements of J_A, J_B and J_C. In this case, any treatment strategy performed at random without considering the topology of the network, will imply very rigid space of realized effective treatments, so structural properties of the network is essentially the only factor determining the shape of the distribution of effective treatments.

Distributional interactions (DI)

An anonymous interaction that doesn't depend on the size of the neighborhood is called distributional interaction. Assumption DI implies that the treatment response varies with the person's own treatment and with the distribution of treatment in his neighborhood. So, the effective treatment given to person i is $c_i(z^J) = (z_i, p_i)$.

In the case, the space of realized effective treatments is even more restricted by the network topology. If we consider the core-periphery toy example of the previous section, the expected number of friends receiving treatment in each neighborhood is ρ, which is a constant across the network if treatment happens at random. So the distribution of the social treatments, i.e, the part of the effective treat-

ment associated with the neighbors, has only one mode centered in ρ. As briefly discussed in the CTR assumption, the space of treatment assignments t^J for which we can infer he distribution $\{y^J(t^J)\}$ out of the observations from $y(z^J)$ is determined by these realized effective treatments. So, the identification region of an experiment whose sampling strategy ignores the topology of the network is too restricted by the network. The theory developed in this chapter defines treatment strategies that will help to better shape the identification region, leaving it less dependent of structural properties of the network.

Statistical Independence (SI)

Manski explains that, despite the unobserved data, it might be possible to transparently estimate the distribution of $y(t^J)$ from experimental data if, in addition to restrictions on the shape of $y(\cdot)$, we assume **statistical independence (SI)** of potential outcomes and realized treatments. In the case of ITR, assumption SI allows point identification of potential outcomes whenever there is positive probability that realized treatment z and potential treatment t coincide.

Manski also analyzes the combination of assumptions CTR and SI. He decomposes the population into sets of effective treatment types $m \in M$ and shows that point identification is possible if and only if every potential effective treatment has positive probability of appearing in the support of realized effective treatments.

Formally, following Manski's notation, we say that two persons i and j are of the same type if there exists a permutation $\pi_{ij} : T^J \rightarrow T^J$ for which $c_i(t^J) = c_j(\pi_{ij}(t^J))$ for all $t^J \in T^J$. Call $J_m \subset J$ the subset of individuals of type m and C_m

the set of effective treatments for persons of type m. Given $t^J \in T^J$ and $\gamma \in C_m$, denote $J_{m\gamma}$ as the set of persons having effective treatment γ when the potential treatment is t^J. The statistical independence assumption states that for each $J_{m\gamma}$ with $P(J_{m\gamma}) > 0$,

$$\{y^{J_{m\gamma}}(t^J)\} = \{y^{J_{m\gamma}(c(z^J)=\gamma)}(t^J)\}, \tag{4.5}$$

where $\{y^{J_{m\gamma}(c(z^J)=\gamma)}(t^J)\}$ is the distribution of outcomes of elements $j \in J_{m\gamma}$ satisfying $c(z_i^J) = \gamma$. Manski shows that, given SI and CTR, the identification region for $\{y^J(t^J)\}$ is

$$\begin{aligned} H[\{y^J(t^J)\}] = &\sum_{m\in M,\gamma\in C_m:P[c(z^J)=\gamma|J_{m\gamma}]>0} \{y^{J_{m\gamma}(c(z^J)=\gamma)}(t^J)\} \cdot \frac{|J_{m\gamma}(c(z^J)=\gamma)|}{|J|} \\ &+\delta. \sum_{m\in M,\gamma\in C_m:P[c(z^J)=\gamma|J_{m\gamma}]=0} \frac{|J_{m\gamma}(c(z^J)\neq\gamma)|}{|J|}, \delta \in \Delta_Y. \end{aligned} \tag{4.6}$$

Thus, $\{y^J(t^J)\}$ is point-identified if and only if $P[c(z^J) = \gamma|J_{m\gamma}] > 0$ for all $m \in M$ and $\gamma \in C_m$ such that $P(J_{m\gamma}) > 0$, i.e., every effective treatment for all treatment types must occur with positive probability. As noted in [46], this condition is difficult to satisfy, specially when reference groups are large or when there exist people belonging to several reference groups.

Usually, treatments are assigned at random to guarantee statistical independence, but, as we will see later in this chapter, using a treatment distribution that ignores the topology of the network describing the reference groups exposes the population to a very limited set of realized effective treatments. This poor setup potentially limits the identification power of the experiment. We present a stochastic blockmodel framework for treatment assignment that offers more flex-

ibility to shape the set of realized effective treatments, as it allows different distributions of treatment for different groups of the population. The groups, which are essentially the blocks of the stochastic blockmodel provided by the SBA, capture important structural properties of the network. This new method of assigning treatment considering social interactions is the main result of this chapter, and we show how to apply our ideas to find optimal designs of experiments.

4.2 Basic Setup and assumptions

To model the distribution of outcomes from an experiment in which social interaction affects treatment response, we describe in the section two types of assumptions: first, we impose restrictions on the shape of $y(\cdot)$ to explain how effective treatments influence outcomes; second, we assume that the generating process of the network is a stochastic blockmodel, this will help us describe the distribution of effective treatments later in the chapter.

4.2.1 Assumptions on the response function

Our basic assumption on the shape of the response function $y_i(\cdot)$ is that the outcome depends only on the individual's treatment t_i and the treatment t^{N_i} assignment in his social neighborhood N_i. We consider two cases: first, we suppose that y_i only depends on t_i and on the distribution of treatments in N_i; second, we let y_i also vary with the size of N_i.

Anonymous interactions with reference groups

Assume that interactions are with reference groups, i.e., the outcome of each person j varies only with the treatment given to his reference group $G(j)$, which is composed by j itself and by his neighborhood $N_j \subset J$. In addition, suppose that interactions are anonymous and that the space of possible treatments is finite of size $k = |T| < \infty$. We call this set of assumptions **anonymous interactions with reference groups (AIRG)**.

Because $T = \{\tau 1, \ldots, \tau_k\}$ is finite and the interactions are anonymous, the effective treatment $c_j(t^J)$ of person j can be described by a $(k+2)$-vector $(t_j, m_j, p_j) \in T \times \mathbb{Z}^+ \times [0,1]^k$, where t_j is j's potential treatment, $m_j = |N_j|$, and p_j is a k-vector in $[0,1]$ whose q-th component represents the proportion of neighbors of j receiving treatment τ_q (here the sum of elements in p_j is 1). Manski's *effective treatments types* are represented in this context by the person's neighborhood size m_j, since it is possible to define bijections $\pi_{ij} : T^J \to T^J$ mapping t^{N_i} to t^{N_j} and $t^{\{i\}}$ to $t^{\{j\}}$ whenever $m_i = m_j$. Note that response for members of the same effective treatment type m can be described by a common effective treatment function $c_m(\cdot,\cdot)$ on $T \times [0,1]^k$:

$$c_j(t^J) = c_m(t_j, p_j) = (t_j, m_j = m, p_j), \tag{4.7}$$

where p_j is a vector representing the treatment distribution of t^{N_j}. It is worthy to observe that for a particular m, every entry of p_j belongs to a discrete subset $\{0 = \frac{0}{m}, \frac{1}{m}, \ldots, \frac{m}{m} = 1\}$. We denote Υ_m the (discrete) set of possible distributions of treatment p_j in the neighborhood of elements of treatment type m.

Distributional interactions with reference groups

We also work with a variation of AIRG based on distributional interaction, the **distributional interactions with reference groups (DIRG)**. In distributional interactions, the response function $y_j(\cdot)$ of person j varies only with the treatment t_j attributed to j and with the distribution of treatments in his neighborhood N_j, namely p_j. The effective treatments are given by $c_j(t^J) = (t_j, p_j)$.

4.2.2 Stochastic Blockmodels for social interactions

Manski's analysis on treatment response with social interaction gives some theoretical understanding on the identification power of randomized treatment strategies under various assumptions shaping the response function. However, it doesn't provide a methodology to study the complex patterns these social structures might form throughout the network. In order to estimate response, it is essential to understand the formation of the set of effective treatments associated with the given assignments. In practice most networks generate complex sets of effective treatments and developing a methodology of treatment design that takes into account the network structure and the corresponding distribution of effective treatments is generally quite challenging.

In our framework we model social interaction using stochastic blockmodels. There are two main reasons for choosing this type of model: first, the theory of graphs limits shows that graphons and stochastic blockmodels are good representations to massive dense networks; second, as we will see later in the chapter, patterns of connection provided by stochastic blockmodels are useful to describe

effective treatments under reasonable assumptions on the response function and under flexible treatment assignments. In our procedures, we apply SBA to estimate a stochastic blockmodel from the network and use the model parameters to find optimal treatment strategies.

4.3 The treatment assignment

To motivate our methodology, let's analyze how I.I.D treatment assignments generate distributions of effective treatment across the population. We start considering the assumption DIRG. Since for any two individuals i and j the sets N_i and N_j are exposed to similar distributions of treatment, if $|N_i|$ and $|N_j|$ are large we should expect p_i and p_j to be very close. Thus, if i and j happen to receive the same treatment $t_i = t_j = t$, they should have similar $c_i(t^J) = (t, p_i)$ and $c_j(t^J) = (t, p_j)$, as all individuals are exposed to similar social effect. It is expected that the outcomes will be observed from a very homogeneous set of response functions. The same happens for AIRG. In that case, individuals i and j with the same treatment and the same number of neighbors would have similar effective treatment $c_i(t^J) = (t, m, p_i)$ and $c_j(t^J) = (t, m, p_j)$. In both cases, the space of realized effective treatments is too rigid. The homogeneity described above limits the space of potential effective treatments and, consequently, the identification power of the experiment.

In order to generate richer sets of realized effective treatments, it is necessary to design treatment strategies that somehow considers the topology of the social graph. But using complicated assignments brings many challenges to the analysis of the set of realized effective treatment. As effective treatments are defined by

the treatment in the reference groups and as the reference groups usually intersect forming quite complex structures, estimating the distribution of realized affective treatments can be very difficult. The main question we explore in this section is: how to define a treatment strategy that uses the social graph structure to possibly sample treatments from different distributions for different people in a way to better control the set of realized effective treatments?

4.3.1 Stochastic blockmodel treatment assignment

The idea of our method is to use the regular connections given by the stochastic blockmodels to design a treatment assignment mechanism that creates rich sets of realized effective treatments. The stochastic blockmodels regularities help us understand how overlapping reference groups participate in mutual interactions and how these interdependencies affect the realization of effective treatments. The method consists of assigning a particular probability of treatments to each block of the blockmodel.

Formally, let G be a graph representing the social ties defined by the reference groups: every member of J is a node, and an edge between I and J exists if and only if $J \in N_I$. Apply the SBA algorithm on G and let the outcome be a stochastic blockmodel $(\boldsymbol{\alpha}, M_{q\times q})$ whose blocks $B_1, \ldots, B_q$ have respective sizes (probabilities) $A_1, \ldots, A_q$ and are connected with probabilities given by a $q \times q$ matrix M, i.e., $i \in B_I$ connects with $j \in B_J$ with probability M_{IJ}. Finally, suppose that to each block B_i we assign a distribution of treatments defined by the vector of probabilities $F_I = (f_{I1}, \ldots, f_{Ik})$ and let F be a $k \times q$ matrix whose columns are the vectors F_I.

4.4 Distribution of effective treatments

We would like to estimate the distribution of effective treatments across the population for the treatment assignment described above when we assume AIRG and DIRG.

Let's first use the outcome of the SBA algorithm to approximate G to a random graph $\widetilde{G}$ defined in the same set of vertices. In a realization $\widehat{G}$ of $\widetilde{G}$, vertices have the same label and belong to the same blocks $B_1, \ldots, B_q$ as in G, but connection between $v_i \in B_I$ and $v_j \in B_J$ are assigned independently with probability M_{ij}. The treatment assignment in $\widehat{G}$ follows the same procedure described in last section, with elements of block B_I being treated independently with distribution $\text{Mult}(1, F_I) = (f_{I1}, \ldots, f_{Ik})$. Since vertices i_1 and i_2 in the same block B_I connect with vertices from any other block B_J with the same frequency, and since treatment within each B_J is assigned from using the same distribution F_J, i_1 and i_2 are exposed to the same distribution of social treatments r_I and p_I.

Now, let's define some notation. For given Blocks B_I and B_J, treatment $\tau \in T$ and individual $v_i \in B_I$, let $A_{IJ}^{\tau}(v_i)$ be the number of individuals in $N_{v_i} \cap B_J$ receiving treatment τ, A_I^{τ} be the number of individuals in B_I receiving treatment τ, and A_I be the size of block B_I. Note that the number of neighbors of $v_i \in B_I$ receiving treatment τ is $C_i^{\tau}(v_i) = \sum_J A_{IJ}^{\tau}(v_i)$, so the distribution of treatments in $N_I(v_i)$ is given by the vector $r_I(v_i) = (\sum_J A_{IJ}^{\tau_1}(v_i), \ldots, \sum_J A_{IJ}^{\tau_k}(v_i))$, for the case AIRG, and $p_I(v_i) = \frac{(\sum_J A_{IJ}^{\tau_1}(v_i), \ldots, \sum_J A_{IJ}^{\tau_k}(v_i))}{\sum_{\tau} \sum_J A_{IJ}^{\tau}(v_i)}$, for the case DIRG. The following proposition decomposes the distribution of effective social treatment with respect to the random variables defined above.

Proposition 13. *The probability distribution of social treatments for a node $v_i \in B_I$ is*

$$P\left(\{s_I(v_i)\}_{I,v_i\in B_I}|A_1,\ldots,A_q,F,M\right) = \sum_{\sum_\tau A_I^\tau = A_I,\forall I} \prod_I \left[P(A_I^{\tau_1},\ldots,A_I^{\tau_k}|A_I,F_i) \prod_{v_i\in B_I} P\left(s_I(v_i)|A_1^{\tau_1},\ldots,A_q^{\tau_k},M\right)\right] \tag{4.8}$$

where s_i equals p_i or r_i for DIRG or AIRG, respectively. Moreover,

$$P(A_I^{\tau_1},\ldots,A_I^{\tau_k}|A_I,F_I) = Mult(A_I^{\tau_1},\ldots,A_I^{\tau_k};A_I,F_I). \tag{4.9}$$

Proof. The result is a straightforward consequence of the definitions presented in the previous two paragraphs. □

Evaluating the expression in equation 4.8 involves dealing with the sum of an exponential number of terms. To facilitate computations and motivated by the fact that the A_I's tend to increase as the number of vertices in the network increases, we approximate $(A_I^{\tau_1},\ldots,A_I^{\tau_k})$ to its average $(A_If_{I1},\ldots,A_If_{Ik})$, what intuitively means that the number of people receiving treatment τ in each block B_I is its expected value $A_If_{I\tau}$. This requirement can actually be accomplished with a small change in the treatment assignment: instead of treatment within a block B_I being assigned independently with distribution F_I, assignment is made at random in a way to satisfy the constraint $(A_I^{\tau_1},\ldots,A_I^{\tau_k}) = (A_If_{I1},\ldots,A_If_{Ik})$, i.e., every possible assignment satisfying the constraint receives the same probability.

In this scenario, equation 4.8 becomes:

$$P\left(\{s_I(v_i)\}_{I,v_i\in B_I}|A_1,\ldots,A_q,F,M\right)=\prod_I\left[\prod_{v_i\in B_I}P\left(s_I(v_i)|A_1^{\tau_1}=A_1f_{1\tau_1},\ldots,A_q^{\tau_k}=A_qf_{q\tau_k},M\right)\right] \tag{4.10}$$

4.4.1 Distribution of effective treatments under AIRG

Proposition 14. *Under AIRG,*

$$P(\{r_I(v_i)\}_{I,v_i\in B_I}|A_1,\ldots,A_q,F,M)\cong\prod_I\prod_{v_i\in B_I}N\left(r_I(v_i);\mu_I,\varsigma_I^2\right)$$

where

$$\mu_I=\left(\sum_J A_Jf_{j\tau_1}M_{IJ},\ldots,\sum_J A_Jf_{J\tau_k}M_{IJ}\right)$$

and

$$\varsigma_I^2=diag\left(\sum_J A_Jf_{J\tau_1}M_{IJ}(1-M_{IJ}),\ldots,\sum_J A_Jf_{j\tau_k}M_{IJ}(1-M_{IJ})\right)$$

Proof. Considering the stochastic blockmodels data generating process for the network and using the definitions described in section 4.4,

$$P(A_{IJ}^{\tau}(v_i)|A_J^{\tau},M_{ij})=\mathrm{Bin}(A_{IJ}^{\tau}(v_i);A_J^{\tau},M_{IJ}). \tag{4.11}$$

Because A_J is large, we approximate (4.11) to normal:

$$P(A_{IJ}^{\tau}(v_i)|A_J^{\tau},M_{IJ})\cong\mathrm{N}(A_{IJ}^{\tau}(v_i);A_J^{\tau}M_{IJ},A_J^{\tau}M_{IJ}(1-M_{IJ})) \tag{4.12}$$

From the definition of r_i,

$$P(r_I(v_i)|A_1^{\tau_1},\ldots,A_q^{\tau_1},\ldots,A_1^{\tau_k},\ldots,A_q^{\tau_k},M) \cong \\ \mathrm{N}\left[\left(\sum_J A_{IJ}^{\tau_1}(v_i),\ldots,\sum_J A_{IJ}^{\tau_k}(v_i)\right);\mu_I,\varsigma_I^2\right] \tag{4.13}$$

where $\mu_I = \left(\sum_J A_J^{\tau_1} M_{IJ},\ldots,\sum_J A_J^{\tau_k} M_{IJ}\right)$ and

$$\varsigma_I^2 = diag\left(\sum_J A_J^{\tau_1} M_{IJ}(1-M_{IJ}),\ldots,\sum_J A_J^{\tau_k} M_{IJ}(1-M_{IJ})\right)$$

The result now comes from the fact that $A_I^{\tau} = A_I f_{I\tau}$ (as discussed in section (as discussed in section 4.4).

□

4.4.2 Distribution of effective treatment under DIRG

Proposition 15. *Under DIRG,*

$$P(\{p_I(v_i)\}_{I,v_i\in B_I}|A_1,\ldots,A_q,F,M) \cong \prod_I \prod_{v_i\in B_I} N\left(p_I(v_i);\mu_I,\varsigma_I^2\right)$$

where

$$\mu_I \cong \left(\frac{\sum_J A_J f_{J\tau_1} M_{IJ}}{\sum_j M_{IJ} A_J},\ldots,\frac{\sum_J A_J f_{J\tau_k} M_{IJ}}{\sum_J M_{IJ} A_J}\right)$$

and

$$\varsigma_I^2 = \frac{1}{(\sum_j M_{IJ} A_J)^2} diag\left(\sum_J A_J f_{J\tau_1} M_{IJ}(1-M_{IJ}),\ldots,\sum_J A_J f_{j\tau_k} M_{IJ}(1-M_{IJ})\right)$$

Moreover, as $n \to \infty$, the social effect of node $v_i \in B_I$ asymptotically approaches

$$p_I(v_i) \cong \left(\frac{\sum_J A_J f_{J\tau_1} M_{IJ}}{\sum_J M_{IJ} A_J}, \ldots, \frac{\sum_J A_J f_{J\tau_k} M_{IJ}}{\sum_J M_{IJ} A_J} \right) \tag{4.14}$$

Proof. The result is a direct consequence of proposition 14 and of the fact that $p_I(v_i) = \frac{r_i}{\sum_\tau \sum_J A^\tau_{IJ}(v_i)}$,. When $n \to \infty$ each $A_I \to \infty$, so the variance ς_I^2 approaches zero and p_i becomes assumptotically equivalent to m_I. We call $W_{IJ} = \frac{A_J M_{IJ}}{\sum_J A_J M_{IJ}}$, so the expression for μ_I becomes $\mu_I \cong \left(\sum_J f_{J\tau_1} W_{IJ}, \ldots, \sum_J f_{J\tau_k} W_{IJ} \right)$. □

Under the asymptotic setting of proposition (15), individuals in the same block of the stochastic blockmodel have approximately the same distribution of treatments in their neighborhood. The effective treatment of v_i is then defined by $(t_i, p_I(v_i))$, where t_i is the treatment received by v_i and $p_I(v_i)$ is defined above. Since the treatment assignment in B_I is given by F_I, for each τ approximately $A_I f_{I\tau}$ individuals of block B_I should have effective treatment $(\tau, p_I(v_i))$.

4.5 Optimal design for identification of social interactions

In this section we show how the stochastic blockmodel treatment assignment can be used to improve estimation of treatment and social effects in a linear model. The setup and the distribution of effective treatments follow the description in sections 4.3.1 and 4.4, but here we assume that the space of treatments is $T = \{0, 1\}$, where 0 represents no-treatment and 1 represents treatment.

We consider the following linear model for the response function

$$\begin{aligned} y_{v_i} &= \alpha + \beta t_{v_i} + \gamma s_{v_i}(v_i) + \epsilon_{v_i}, \\ \epsilon_{v_i} &\sim \mathrm{N}(0, \sigma^2_{v_i}), \end{aligned} \tag{4.15}$$

where the $s_{v_i} = p_{v_i}$ in case of DIRG and $s_{v_i} = r_{v_i}$ in case of AIRG. Since the distribution of treatments is randomized within blocks, and since members of the same blocks are exposed to similar distribution of social effect s_{v_i}, for a matter of estimating the overall distribution of outcomes y, we assume that they share the same σ_{v_i}, i.e., the variance of ϵ_{v_i} can be described by a σ_I common among members of block $B_I \ni v_i$.

Our goal is to design a treatment assignment that optimizes the estimation of α, β and γ. More precisely, we want to minimize the variance of their estimators. The covariance of the estimators is the inverse of the Fisher information matrix, and the Fisher information is computed as a function of the stochastic blockmodels parameters and the treatment probability in each block. We want to find treatment probabliities that provide good estimation for α, β and γ satisfying a certain budget constraint that limits the number of people to be treated.

We present optimal and suboptimal designs for estimating treatment and social effects. For optimality, we consider the network as a random graph model defined from the output of SBA. In this case, we consider the social effect a nuisance parameter that is integrated out in the computation of the Fisher information. For the suboptimal designs, we explore limiting properties of the stochastic blockmodels distribution.

4.5.1 Optimal design for DIRG and AIRG

To find optimal designs for DIRG and AIRG models, we use the stochastic block-model given by the SBA to model the connections of the network. We assume that the block assignment doesn't change, i.e., vertices are assigned to the same blocks as in the output of the algorithm. The connections happen independently with probabilities given by the connection matrix provided by the SBA. We assume a directed graph to facilitate computation, as independence in edge direction implies more independence in the formation of effective treatments among units. This random graph is used to integrate out the social component of the likelihood of the response in the linear model, so we use the likelihood to find a Fisher information matrix associated with the regression parameters. The optimal design finds treatment assignments that maximize the information about the parameters or, equivalently, minimizes the variance of the estimators.

Under this setup, the Fisher information is described in the following theorem.

Theorem 16. *Consider a random graph model in which vertices have been assigned to blocks $B_1, B_2, \ldots, B_k$ and connection between any pair of vertices $i \in B_I$ and $j \in B_J$ happen independently with known probability M_{IJ} that depends only on which block they belong to. Assume that the response function is given by the linear model 4.15, where the social effect s_i is p_i, for the case DIRG, or r_i, for the case of AIRG. Then the fisher information matrix associated with model parameters α, β, γ is given by*

$$I = -\begin{pmatrix} L & \mathsf{C} & \mathsf{D} \\ \mathsf{C} & \mathsf{C} & \mathsf{E} \\ \mathsf{D} & \mathsf{E} & \mathsf{F} \end{pmatrix},$$

where the entries are $L = \frac{1}{n}\sum_I \frac{A_I}{\sigma_I^2}\left(\frac{\gamma^2\varsigma_I^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)} - 1\right)$, $C\frac{1}{n}\sum_I \frac{A_I f_I}{\sigma_I^2}\left(\frac{\gamma^2\varsigma_I^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)} - 1\right)$, $D = -\frac{1}{n}\sum_I \frac{A_I\mu_I}{(\gamma^2\varsigma_I^2+\sigma_I^2)}$, $E = -\frac{1}{n}\sum_I \frac{A_I\mu_I f_I}{(\gamma^2\varsigma_I^2+\sigma_I^2)}$, *and*

$$F = \frac{1}{n}\sum_I A_I \left[\frac{\gamma^2\varsigma_I^4 - \varsigma_I^2\sigma_I^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2} + \frac{(\varsigma_I^2+\mu_I^2)^2\gamma^2\varsigma_I^2}{\sigma^2(\gamma^2\varsigma_I^2+\sigma_I^2)} - 4\gamma\varsigma_I^2\frac{\left(\frac{(\varsigma_I^2+\mu_I^2)^2\gamma^3\varsigma_I^2}{\sigma_I^2} + \mu^2\gamma\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2}\right.$$

$$\left. + \frac{(\varsigma_I^2\gamma^2 - \sigma_I^2)\left[(\mu_I\gamma^2\varsigma_I^2 + \mu_i\sigma_I^2)^2 + \gamma^4\varsigma_I^6\right]}{\sigma_I^2(\gamma^2\varsigma_I^2+\sigma_I^2)^3}\right]$$

Here A_I *is the size of block* B_I, f_I *is the probability of treatment in block* B_I, σ_I^2 *is the variance of the response for members of block* B_I, γ *is the social effect parameter of the linear model, and* (μ_i, ς_I^2) *are defined in proposition 14 for AIRG models, or in proposition 15 for DIRG models.*

Proof. The proof is given in appendix C. □

As I^{-1} is multidimensional, to define a notion of optimality we need consider a unidimensional functional of I. In following subsections we analyze optimal strategies the social effect parameter, represented by the γ, and treatment effect paremeter, represented by β.

Optimizing estimation of social effect

If the goal is to find optimal experimental design to estimate the social effect parameter, we could focus on minimizing the variance of the γ estimator. This corresponds to minimizing the element in position $(3,3)$ of I^{-1}, i.e., we would like to minimize $H(f) = \frac{1}{det(I)}\left[LC - C^2\right]$. The minimization problem we need to solve in

order to design an optimal experiment for estimation of the social effect is:

- Find $f_{11}, \ldots, f_{q,1}$.
- That minimize

$$H = \frac{LC - C^2}{LCF + 2CDE - CD^2 - C^2F - LE^2} \tag{4.16}$$

- Given the constraints:
  - $\frac{1}{n} \sum_{i=1}^{q} A_I f_{I1} = S$.
  - $0 \leq f_I \leq 1, \forall I$.

The constraints guarantee that the f_I's are indeed probabilities and that the budge constraint is satisfied.

Optimizing estimation of treatment effect

For estimating treatment effect, we should minimize the element in position $(2, 2)$ of I^{-1}, i.e., our goal should be to minimize

$$H(f_1, \ldots, f_q) = \frac{1}{det(I)} \left[LF - D^2\right] =$$

Subject to

- $\frac{1}{n} \sum_{i=1}^{q} A_I f_{I1} = S$.
- $0 \leq f_I \leq 1, \forall I$.

4.5.2 Suboptimal designs

The methodology developed in section 4.5 involves computing the Fisher information I described in theorem 16. The matrix I, however, depends on the parameter model γ, which is actually what we would like to estimate. Here we develop a suboptimal methodology that considers the asymptotic setup of proposition 15 to define an approximation of the Fisher information matrix in a way that it doesn't depend on the regression parameters.

Designing suboptimal experiments for DIRG linear models

We consider DIRG linear models and the asymptotic setup of theorem 16, in which the social treatment of an individual v_i in block B_I is given by

$$p_I(v_i) \cong \sum_J f_{J1} W_{IJ}$$

Therefore the treatment type of v_i can be written as

$$c_{v_i}(z^J) = \left(t_{v_i}, \sum_J f_{J1} W_{IJ} \right). \tag{4.17}$$

For each $v_i \in B_I$, there are approximately $A_I f_{I1}$ individuals receiving effective treatment $\left(1, \sum_j f_{J1} W_{IJ}\right)$ and $A_i(1 - f_{I1})$ receiving treatment $\left(0, \sum_J f_{J1} W_{IJ}\right)$.

The fisher information under this setup is then

$$I = \frac{1}{n}\sum_{B_I}\sum_{v_i \in B_I}\frac{1}{\sigma^2_{B_I}}\begin{pmatrix} 1 & t_{v_i} & p_I(v_i) \\ t_{v_i} & t^2_{v_i} & t_{v_i}p_I(v_i) \\ p_i(v_i) & t_{v_i}p_i(v_i) & p_I(v_i)^2 \end{pmatrix} =$$

$$\frac{1}{n}\sum_{I=1}^{q}\frac{A_I f_{I1}}{\sigma^2_{B_I}}\begin{pmatrix} 1 & 1 & \sum_j f_{j1}W_{IJ} \\ 1 & 1 & \sum_j f_{j1}W_{IJ} \\ \sum_j f_{J1}W_{IJ} & \sum_j f_{J1}W_{IJ} & \left(\sum_J f_{J1}W_{IJ}\right)^2 \end{pmatrix} +$$

$$\frac{1}{n}\sum_{I}\frac{A_I(1-f_{I1})}{\sigma^2_I}\begin{pmatrix} 1 & 0 & \sum_J f_{J1}W_{IJ} \\ 0 & 0 & 0 \\ \sum_J f_{J1}W_{IJ} & 0 & \left(\sum_J f_{J1}W_{IJ}\right)^2 \end{pmatrix} = \begin{pmatrix} L & C & D \\ C & C & E \\ D & E & F \end{pmatrix},$$

where

$$L = \frac{1}{n}\sum_I \frac{A_i}{\sigma^2_{B_I}}, C = \frac{1}{n}\sum_I \frac{A_I}{\sigma^2_{B_I}}f_{I1}, D = \frac{1}{n}\sum_I \frac{A_I}{\sigma^2_{B_I}}\left(\sum_J f_J W_{IJ}\right),$$

$$E = \frac{1}{n}\sum_I \frac{A_I}{\sigma^2_{B_I}}f_I\left(\sum_J f_J W_{IJ}\right) \text{ and } F = \frac{1}{n}\sum_I \frac{A_I}{\sigma^2_{B_I}}\left(\sum_J f_J W_{IJ}\right)^2.$$

This matrix can be used to find an approximated covariance matrix for the estimator is the parameters α, β and γ so we follow the optimization setup presented in section 4.5.1 to define a suboptimal treatment assignment methodology.

Designing suboptimal experiments for AIRG linear models

We follow similar ideas as in the DIRG suboptimal case and approximate r_i to its mean $\sum_J f_J A_J M_{IJ}$. The information matrix becomes

$$\begin{pmatrix} 1 & C & D \\ C & C & E \\ D & E & F \end{pmatrix},$$

where

$$C = \frac{1}{n}\sum_{I=1}^{q} A_I f_{I1}, D = \frac{1}{n}\sum_I A_I \left(\sum_J f_J A_J M_{IJ}\right),$$

$$E = \frac{1}{n}\sum_I A_I f_I \left(\sum_J f_J A_J M_{IJ}\right) \text{ and } F = \frac{1}{n}\sum_I A_I \left(\sum_J f_J A_J M_{IJ}\right)^2$$

.

Optimal estimation of the peer effect and treatment effect follows then similar analysis as the one developed in section 4.5.1.

4.6 Estimation of causal effect

In this section, we review the causal estimands of peer-influence introduced in [61] and apply them to our setup. The peer influence is measured in different levels, each level k corresponding to the number of friends receiving treatment in the person's neighborhood. Using the paper's notation, let $Z(N_i, k)$ be the set of all assignments on N_i in which exactly k neighbors are treated. There are two types of estimands:

1. Estimand for primary effect:

$$\xi = \frac{1}{n} \sum_i Y_i(1, z = 0) - Y_i(0) \tag{4.18}$$

2. Estimand of level k for peer-influence:

$$\delta_k = \frac{1}{|V_k|} \sum_{i \in V_k} \left[\frac{n_i!}{k!(n_i - k)!} \sum_{z \in Z(N_i, k)} Y_i(0, z) - Y_i(0) \right], \tag{4.19}$$

where V_k is the set of nodes with at least k neighbors.

In the paper's model-based approach for causal inference, the authors work with a linear model defined as follows:

$$\begin{aligned} y_{v_i} &= \alpha + \beta t_{v_i} + \gamma \left(O^T Z \right) + \epsilon_{v_i}, \\ \epsilon_{v_i} &\sim \mathrm{N}(0, \sigma^2), \end{aligned} \tag{4.20}$$

with O being a weighted direct matrix. They show that the primary estimand reduces to β:

$$\xi = \frac{1}{n} \sum_i Y_i(1, z = 0) - Y_i(0) = \frac{1}{n} \sum_i \beta = \beta. \tag{4.21}$$

The peer influence effect estimand δ_k is reduced to τ:

$$\delta_k = \frac{1}{|V_k|} \sum_{i \in V_k} \left[\frac{n_i!}{k!(n_i - k)!} \sum_{z \in Z(N_i, k)} Y_i(0, z) - Y_i(0) \right] = \frac{k\gamma}{|V_k|} \sum_i \mathbb{W}_i \propto \gamma, \tag{4.22}$$

where $\mathbb{W}_i$ is the average weight in the incoming edge to i.

Note that model 4.15 can be expressed in format 4.20 for both AIRG and DIRG: in the AIRG case the edges of O have weight 1, and in the DIRG case the edges have

weight $\frac{1}{deg(i)}$. So, the k-level peer-influence effect is given by

- In case of AIRG: $\delta_k = k\gamma$
- In case of DIRG: $\delta_k = \frac{k\gamma}{|V_k|} \sum_{i \in V_k} \frac{1}{deg(i)}$

4.7 Experimental results

In this section we present simulations to show the performance of the stochastic blockmodel treatment assignment. In our experiments, data is generated using six types of treatment strategies:

- *At random*: each vertex receives treatment I.I.D with probability p.
- *Heuristics*: randomly choose blocks from SBA and treat everybody from that block (or the maximum number of people) until it reaches a desirable number pn of treatments.
- *Suboptimal (DIRG)*: suboptimal design for estimation of social effect DIRG described in section 4.5.2.
- *Suboptimal (AIRG)*: suboptimal design for estimation of social effect under AIRG described in section 4.5.2.
- *Optimal (DIRG)*: optimal design for estimation of social effect under DIRG described in section 4.5.1.
- *Optimal (AIRG)*: optimal design for estimation of social effect under AIRG described in section 4.5.1.

The outcome is generated using three types of response functions:

- *DIRG linear model*: model described in 4.15 with $s_i = p_i$.
- *AIRG linear model*: model described in 4.15 with $s_i = r_i$
- *Markov Random Field*: we follow the model introduced in [43] where a Markov Random Field defines a response mechanism with social interaction in which the response of each individual depends on the response of people in his neighborhood. In this model, outcome is binary and the probability of $y_i = 1$ is given by a probit function of a linear combination involving the proportion of people in N_i with outcome 1. Formally,

$$P(y_i = 1|Y_{-i}, G, \psi, \gamma) = \Phi\left(\alpha + \gamma \frac{\sum_{j \in N_i} y_j}{|N_i|}\right) \tag{4.23}$$

  We work with a variation of this method in which the linear combination in the probit function has one more term that depends on the treatment the individual received:

$$P(y_i = 1|Y_{-i}, G, \psi, \gamma) = \Phi\left(\alpha + \beta t_i + \gamma \frac{\sum_{j \in N_i} y_j}{|N_i|}\right) \tag{4.24}$$

  This way, the probability of outcome y_i depends indirectly on the treatment given to the elements on N_i. We implemented the MCMC mechanism described in [43] to sample from such distribution.

4.7.1 Experimental setup

We randomly generated 1000 networks of 500 nodes each from stochastic blockmodels with 5 blocks. The parameters of the stochastic blockmodels are sampled from I.I.D. uniform distribution on $[0, 1]$ for the probabilities of connection between blocks and dirichlet distribution with parameter 1 for the probabilities of the blocks. After sampling one graph from each stochastic blockmodel, we applied all six treatment strategies described above in a way to cover in each case approximately 30% of the vertices in the graph. Then we sample the outcome using each one of the three models presented above. Finally, we compare the different treatment strategies by running a linear model on the generated data and then observing the distance to ground truth and the variance of the regression estimators.

The models are defined using the following parameters:

- DIRG linear model: the response function is defined with parameters: $\alpha = 3, \beta = 10, \gamma = 3$. The variances θ_I of the error for each block B_I are sampled independently from a inverse-gamma distribution with mean 1 and standard deviation 0.1. We use a DIRG linear regression model to evaluate the performance of the treatment strategies.
- AIRG linear mode: the response functions defined with parameters: $\alpha = 10, \beta = 20, \gamma = 0.1$. The variances θ_I of the error for each block B_I are sampled independently from a inverse-gamma distribution with mean 20 and standard deviation 2. We use a AIRG linear regression model to evaluate the performance of the treatment strategies.
- MRF: response function for the Markov Random Field is defined with pa-

rameters: $\alpha = 1, \beta = 1, \gamma = 1$. We use a DIRG linear regression model to evaluate the performance of the treatment strategies.

4.7.2 Results

Results are shown on tables 4.1, 4.2 and 4.3. Table 4.1 presents the results of a DIRG regression fit on data generated using the DIRG model described in last section. Note that the social and treatment effects parameter have lowest distance to the ground truth and lowest variance when the optimal DIRG treatment assignment is used. Table 4.2 has results of a AIRG linear regression fit on data generated using the AIRG linear model. Here the best treatment strategy for estimating social and treatment effects shows to be the optimal AIRG described in section 4.5.2. Finally, table 4.3 shows results of DIRG regression fit on data generated using the Markov Random Field. Despite the model misspecification , here again our theory is confirmed as the optimal DIRG outperforms the other treatment strategies. Also notice that the heuristics outperforms the "at random" assignment on estimation of the social effect in all three groups of simulations. This shows how useful it is to consider the graph structure, in particular the SBA block structure, when estimating social effect using these types of models.

Table 4.1: Data generated with DIRG linear model. Fit using DIRG linear regression.

		Linear Regression (DIRG)				
Treatment Type		Dist. to Ground Truth	Estimate	Std. Error	t-value	p-value
At Random	(Intercept)	0.541 ± 0.429	3.030 ± 0.690	0.689 ± 0.205	4.811± 1.817	0.007 ± 0.028
	Treatment	0.080 ± 0.058	10.005 ± 0.099	0.100± 0.009	100.890 ± 9.256	0.000 ± 0.000
	Social Effect	1.809± 1.411	2.860 ± 2.291	2.291 ± 0.674	1.364± 1.068	0.297 ± 0.302
Heuristics	(Intercept)	0.190 ±0.191	3.003 ± 0.269	0.234 ± 0.135	16.548 ± 7.922	0.000± 0.002
	Treatment	0.089± 0.072	10.000 ± 0.115	0.110± 0.018	92.722 ± 12.371	0.000 ± 0.000
	Social Effect	0.621± 0.602	2.999 ±0.865	0.765 ±0.429	4.969±2.469	0.026 ± 0.092
Suboptimal Social	(Intercept)	0.140 ± 0.142	2.951 ± 0.193	0.182 ± 0.103	20.553 ± 9.702	0.000 ± 0.000
	Treatment	0.117 ± 0.076	10.044 ± 0.133	0.120± 0.018	85.258 ± 11.083	0.000 ± 0.000
	Social Effect	0.482± 0.432	3.135 ± 0.633	0.582 ± 0.310	6.478 ± 2.721	0.007 ± 0.059
Suboptimal Treatment	(Intercept)	0.273 ± 0.290	3.036 ± 0.397	0.308 ± 0.178	13.798 ± 8.136	0.001 ± 0017
	Treatment	0.087 ± 0.055	9.993 ± 0.103	0.112 ± 0.019	91.126 ± 11.099	0.000 ± 0.000
	Social Effect	0.963 ± 1.106	2.843 ± 1.459	1.035 ± 0.619	4.090 ± 2.749	0.096 ± 0.214
Optimal Social	(Intercept)	0.152 ± 0.150	3.012 ± 0.214	0.194 ± 0.123	19.693 ± 8.645	0.000 ± 0.000
	Treatment	0.0911 ± 0.070	10.004 ± 0.115	0.115± 0.018	89.287 ± 12.897	0.000 ± 0.000
	Social Effect	0.412 ± 0.434	2.965 ± 0.597	0.542 ± 0.395	7.137 ± 3.262	0.013 ± 0.068
Optimal Treatment	(Intercept)	0.298 ± 0.301	3.004 ± 0.424	0.361 ± 0.192	10.650 ± 5.515	0.001 ± 0.007
	Treatment	0.073 ± 0.056	9.988 ± 0.091	0.100 ± 0.009	100.789 ± 9.043	0.000 ± 0.000
	Social Effect	0.967 ± 0.987	3.015 ± 1.382	1.184 ± 0.638	3.287 ± 1.957	0.846 ± 0.191

Table 4.2: Data generated with AIRG linear model. Fit using AIRG linear regression.

		Linear Regression (AIRG)				
Treatment Type		Dist. to Ground Truth	Estimate	Std. Error	t-value	p-value
At Random	(Intercept)	3.801 ± 3.797	10.072 ± 5.374	4.748 ± 2.323	2.535 ± 1.422	0.123± 0.222
	Treatment	1.608± 1.197	20.030 ± 2.006	1.999 ± 0.212	10.123 ± 1.406	0.000 ± 0.000
	Social Effect	0.048 ± 0.048	0.101 ± 0.068	0.061 ± 0.026	1.927 ± 1.199	1.179 ±0.256
Heuristics	(Intercept)	2.802 ± 2.724	9.832 ± 3.907	3.611 ± 2.216	3.456 ± 1.794	0.072 ± 0.178
	Treatment	1.657 ± 1.243	19.983 ± 2.072	2.202 ± 0.345	9.264 ± 1.579	0.000 ± 0.000
	Social Effect	0.034 ± 0.034	0.102 ± 0.049	0.044 ± 0.024	2.780 ± 1.504	0.099 ± 0.203
Suboptimal Social	(Intercept)	3.317 ± 3.196	9.973 ± 4.608	4.138 ± 2.383	3.080 ± 1.732	0.095 ± 0.195
	Treatment	1.872± 1.542	20.037 ± 2.427	2.419 ± 0.650	8.710 ± 1.996	0.000 ± 0.000
	Social Effect	0.035 ± 0.032	0.099 ± 0.048	0.045 ± 0023	2.723 ± 1.498	0.114 ± 0.216
Suboptimal Treatment	(Intercept)	3.262 ± 3.465	9.870 ± 4.759	4.179 ± 2.292	2.964 ± 1.676	0.088 ± 0.177
	Treatment	1.522 ± 1.146	19.943 ± 1.905	2.133 ± 0.320	9.516 ± 1.489	0.000 ± 0.000
	Social Effect	0.040 ± 0.039	0.101 ± 0.055	0.052 ± 0.025	2.382 ± 1.426	0.132 ± 0.213
Optimal Social	(Intercept)	1.996 ± 1.861	10.000 ± 2.731	2.592 ± 1.275	4.393 ± 1.680	0.019 ± 0.088
	Treatment	1.737 ± 1.402	20.027 ± 2.234	2.176 ± 0.288	9.350 ± 1.521	0.000 ± 0.000
	Social Effect	0.023 ± 0.020	0.100 ± 0.030	0.030 ± 0.012	3.663 ± 1.413	0.032 ± 0.114
Optimal Treatment	(Intercept)	3.234 ± 3.026	10.113 ± 4.430	4.224 ± 2.140	2.874 ± 1.467	0.086 ±0.177
	Treatment	1.383 ± 1.088	19.945 ± 1.760	1.994 ± 0.204	10.103 ± 1.338	0.000 ± 0.000
	Social Effect	0.041 ± 0.037	0.100 ± 0.055	0.053 ± 0.024	2.224 ± 1.339	0.152 ±0.235

Table 4.3: Data generated with Markov Random Field. Fit using DIRG linear regression.

Treatment Type		Linear Regression (DIRG)			
		Estimate	Std. Error	t-value	p-value
At Random	(Intercept)	0.276 ± 0.326	0.306 ± 0.089	0.950 ± 1.061	0.376 ± 0.309
	Treatment	0.385 ± 0.045	0.044 ± 0.001	8.703 ± 1.194	0.000 ± 0.000
	Social Effect	-0.028 ± 1.088	1.017 ± 0.292	-0.005 ± 1.005	0.481 ± 0.287
Heuristics	(Intercept)	0.233 ± 0.126	0.105 ± 0.061	2.827 ± 1.644	0.104 ± 1.193
	Treatment	0.380 ± 0.051	0.049 ± 0.007	7.847 ± 1.391	0.000 ± 0.000
	Social Effect	0.121 ± 0.404	0.341 ± 0.193	0.464 ± 1.025	0.463 ± 0.305
Suboptimal Social	(Intercept)	0.236 ± 0.093	0.873 ± 0.057	3.400± 1.707	0.055 ± 1.132
	Treatment	0.378 ± 0.050	0.053 ± 0.009	7.379 ± 1.532	0.000 ± 0.000
	Social Effect	0.120 ± 0.314	0.284 ± 0.184	0.600 ± 0.998	0.463 ± 0.304
Suboptimal Treatment	(Intercept)	0.220 ± 0.180	0.135 ± 0.085	2.251 ± 1.669	0.171 ± 0.258
	Treatment	0.382 ± 0.052	0.050 ± 0.011	7.854 ± 1.571	0.000 ± 0.001
	Social Effect	0.159 ± 0.584	0.444 ± 0.281	0.431 ± 1.076	0.458 ± 0.317
Optimal Social	(Intercept)	0.229 ± 0.103	0.088 ± 0.059	3.315 ± 1.671	0.064 ± 0.159
	Treatment	0.381 ± 0.054	0.051 ± 0.006	7.569 ± 1.449	0.000 ± 0.000
	Social Effect	0.143 ± 0.307	0.245 ± 0.192	0.742 ± 1.102	0.399 ± 0.293
Optimal Treatment	(Intercept)	0.231 ± 0.190	0.161 ± 0.088	1.844 ± 1.332	0.218 ± 0.282
	Treatment	0.381 ± 0.046	0.044 ± 0.001	8.587 ± 1.172	0.000 ± 0.000
	Social Effect	0.133 ± 0.629	0.529 ± 0.292	0.303 ± 0.976	0.499 ± 0.299

Increase in identifiability

The optimal treatment strategies provide a clear gain to the estimation of the social effect parameter, as the results presented on tables 4.1, 4.2 and 4.3 confirm that the variance of these estimators decrease when we use SBA to optimize assignment. This improvement is directly associated with the increase in identifiability: the stochastic blockmodel framework allows the construction of more flexible and disperse set of effective treatments, potentially leading to optimal inputs to the regression.

Figures 4.1 and 4.2 illustrate the increase in identifiability by showing the distribution of the social effects p_i and r_i, for a DIRC and a AIRC simulation respectively, under different treatment strategies. Note that the optimal treatment assignments generate more modes in the distribution, what allows the exploration of different areas of the space of inputs.

Figure 4.1: Distribution of DIRC social effect under different treatment strategies.

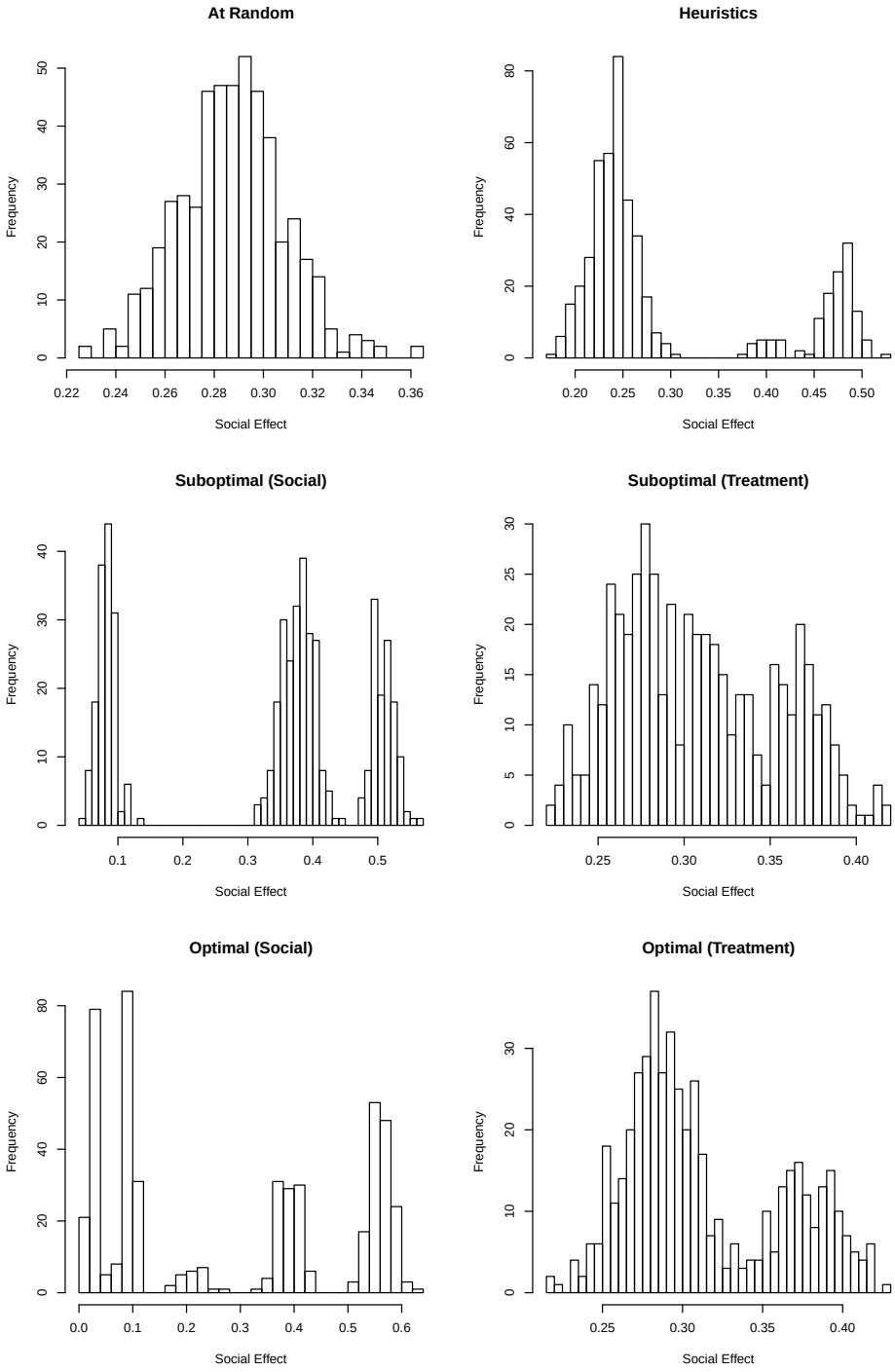
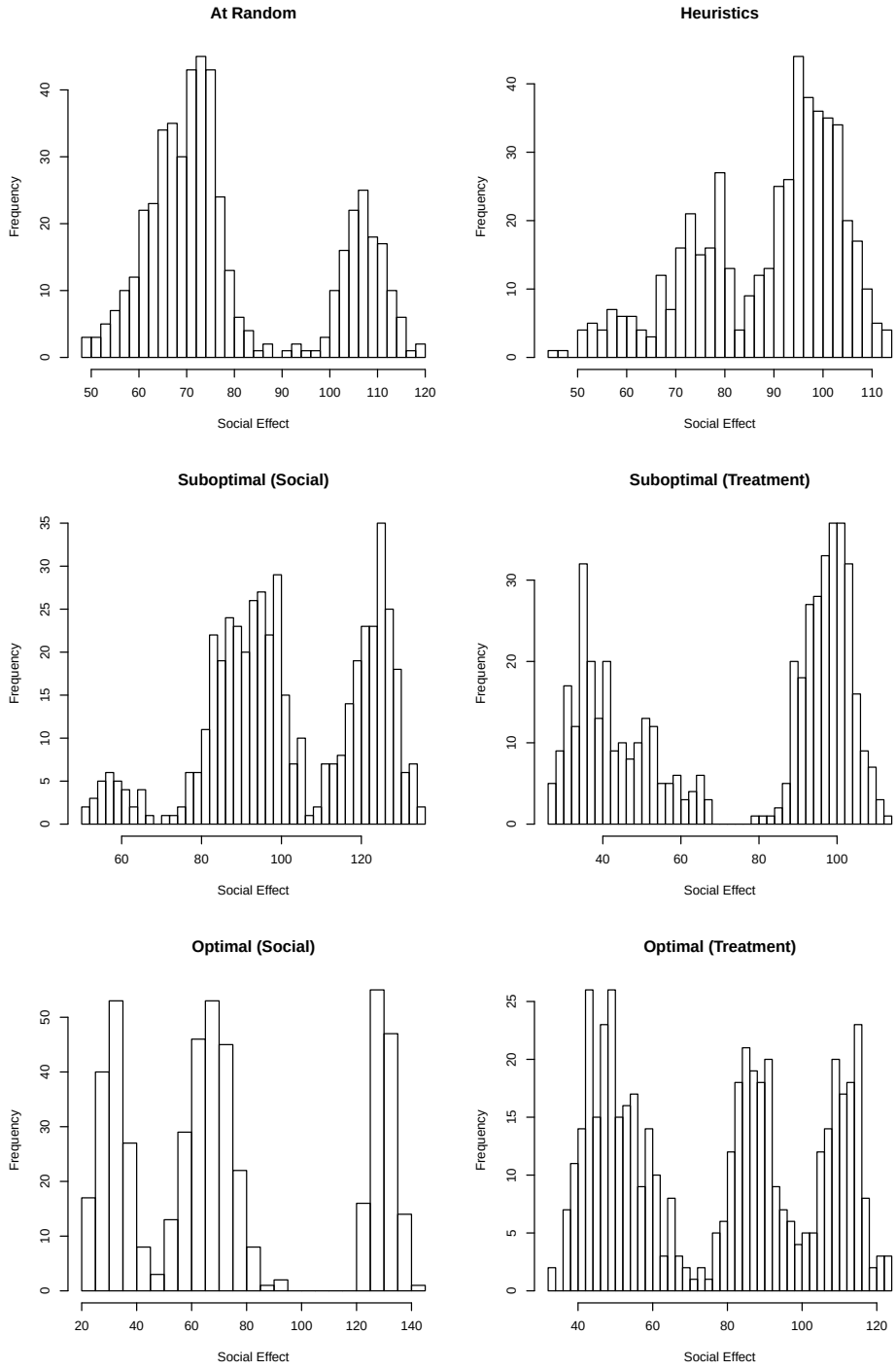


Figure 4.2: Distribution of AIRC social effect under different treatment strategies.



The increase in the number of modes is also clear from the results of table 4.4. For each observed treatment described in the experimental setup, we compute the DIRG social effect p_i and the AIRG social effect r_i. The p_i's and the r_i's are then clustered using a non-parametric kernel algorithm described in [5, 47] and implemented by the R package 'pdfCluster'. Table 4.4 reports the number of clusters and the variance of p_i and t_i. Note the the number of clusters and the variances are higher for the optimal assignments. This shows that the optimal strategies generate richer sets of effective treatments and allow better identifiability.

Table 4.4: DIRC and AIRC Social Effects under different treatment strategies

DIRG Social Effect		
	Number of Clusters	Var(pi)
At random	1.394 ± 0.587	0.00051 ± 0.00034
Heuristics	2.421 ± 0.630	0.00890 ± 0.00907
Suboptimal (Social Effect)	2.487 ± 0.634	0.01370 ± 0.01323
Suboptimal (Treatment)	2.318 ± 0.724	0.00966 ± 0.01099
Optimal (Social Effect)	2.579 ± 0.656	0.01862 ± 0.01472
Optimal (Treatment)	2.164 ± 0.723	0.00313 ± 0.00352

AIRG Social Effect		
	Number of Clusters	Var(ri)
At random	2.279 ± 0.706	340.895 ± 234.789
Heuristics	2.424 ± 0.750	891.254 ± 690.870
Suboptimal (Social Effect)	2.429 ± 0.701	1039.032 ± 808.796
Suboptimal (Treatment)	2.357 ± 0.716	611.107 ± 549.751
Optimal (Social Effect)	2.499 ± 0.687	1422.044 ± 717.596
Optimal (Treatment)	2.284 ± 0.703	464.967 ± 355.567

4.8 Conclusion

We developed a methodology for assessment of treatment response with social interaction based on the stochastic blockmodels approximation framework. Under the assumption that an individual's response to a treatment depends not only on his treatment but also on the treatment assigned to his neighbors in a social graph, we show a way to use the block structure generated by the SBA to design optimal treatment strategies for estimating treatment and social effects. Two classes of models for assessment of social interaction are considered, and for each class we test our ideas considering data generated from linear and non-linear models and using different treatment strategies. Experimental results confirm the optimality of our methodology for parameters' estimation.

Bibliography

[1] E. Airoldi, X. Bai, and K. Carley. Network sampling and classification: An investigation of network model representations. *Decision Support Systems*, 51:506–518, Jun. 2011.

[2] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(1981-2014):3, 2008.

[3] David J Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.

[4] Tim Austin et al. On exchangeable random variables and the statistics of large graphs and hypergraphs. *Probab. Surv*, 5(80-145):10, 2008.

[5] Adelchi Azzalini and Nicola Torelli. Clustering via nonparametric density estimation. *Statistics and Computing*, 17(1):71–80, 2007.

[6] P. Bickel and A. Chen. A nonparametric view of network models and Newman-Girvan and other modularities. *Proc. Natl. Acad. Sci. USA*, 106:21068–21073, 2009.

[7] Peter J Bickel, Aiyou Chen, Elizaveta Levina, et al. The method of moments and degree distributions for network models. *The Annals of Statistics*, 39(5):2280–2301, 2011.

[8] Lawrence E Blume, William A Brock, Steven N Durlauf, and Yannis M Ioannides. Identification of social interactions. Technical report, Reihe Ökonomie/Economics Series, Institut für Höhere Studien (IHS), 2010.

[9] Lawrence E Blume, William A Brock, Steven N Durlauf, and Rajshri Jayaraman. Linear social interactions models. Technical report, National Bureau of Economic Research, 2013.

[10] Béla Bollobás, Christian Borgs, Jennifer Chayes, Oliver Riordan, et al. Percolation on dense graph sequences. *The Annals of Probability*, 38(1):150–183, 2010.

[11] Béla Bollobás, Svante Janson, and Oliver Riordan. The phase transition in inhomogeneous random graphs. *Random Structures & Algorithms*, 31(1):3–122, 2007.

[12] Béla Bollobás and Oliver Riordan. Metrics for sparse graphs. *arXiv preprint arXiv:0708.1919*, 2007.

[13] C. Borgs, J. Chayes, L. Lovasz, V. Sos, B. Szegedy, and K. Vesztergombi. Graph limits and parameter testing. In *Proc. ACM Symposium on Theory of Computing*, pages 261–270, 2006.

[14] Christian Borgs, Jennifer T Chayes, Henry Cohn, and Yufei Zhao. An l^p theory of sparse graph convergence i: limits, sparse random graph models, and power law distributions. *arXiv preprint arXiv:1401.2906*, 2014.

[15] Christian Borgs, Jennifer T Chayes, Henry Cohn, and Yufei Zhao. An l^p theory of sparse graph convergence ii: limits, sparse random graph models, and power law distributions. *arXiv preprint arXiv:1408.0744*, 2014.

[16] Christian Borgs, Jennifer T Chayes, László Lovász, Vera T Sós, and Katalin Vesztergombi. Convergent sequences of dense graphs i: Subgraph frequencies, metric properties and testing. *Advances in Mathematics*, 219(6):1801–1851, 2008.

[17] Christian Borgs, Jennifer T Chayes, László Lovász, Vera T Sós, and Katalin Vesztergombi. Convergent sequences of dense graphs ii. multiway cuts and statistical physics. *Annals of Mathematics*, 176(1):151–219, 2012.

[18] Stanley H Chan and Edoardo M Airoldi. A consistent histogram estimator for exchangeable graph models. *arXiv preprint arXiv:1402.1888*, 2014.

[19] A. Channarond, J. Daudin, and S. Robin. Classification and estimation in the Stochastic Blockmodel based on the empirical degrees. *Electronic Journal of Statistics*, 6:2574–2601, 2012.

[20] Sourav Chatterjee. Matrix estimation by universal singular value thresholding. *arXiv preprint arXiv:1212.1247*, 2012.

[21] Sourav Chatterjee, Persi Diaconis, et al. Estimating and understanding exponential random graph models. *The Annals of Statistics*, 41(5):2428–2461, 2013.

[22] Sourav Chatterjee and SRS Varadhan. The large deviation principle for the erdős-rényi random graph. *European Journal of Combinatorics*, 32(7):1000–1017, 2011.

[23] David Choi, Patrick J Wolfe, et al. Co-clustering separately exchangeable network data. *The Annals of Statistics*, 42(1):29–63, 2014.

[24] David S Choi, Patrick J Wolfe, and Edoardo M Airoldi. Stochastic blockmodels with a growing number of classes. *Biometrika*, 99(2):273–284, 2012.

[25] Persi Diaconis and Svante Janson. Graph limits and exchangeable random graphs, 2007.

[26] Donniell E Fishkind, Daniel L Sussman, Minh Tang, Joshua T Vogelstein, and Carey E Priebe. 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.

[27] Alan Frieze and Ravi Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

[28] A. Goldenberg, A. Zheng, S. Fienberg, and E. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2:129–233, 2009.

[29] P. Hoff. Modeling homophily and stochastic equivalence in symmetric relational data. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pages 657–664, 2008.

[30] P. Hoff, A. Raftery, and M. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

[31] D. Hoover. Relations on probability spaces and arrays of random variables. Technical report, Institute for Advanced Study, Princeton, 1979.

[32] David N Hoover. Row-column exchangeability and a generalized model for probability. *Exchangeability in Probability and Statistics, North-Holland, Amsterdam*, pages 81–291, 1982.

[33] D. R. Hunter and M. S. Handcock. Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*, 15(3):565–583, 2006.

[34] O. Kallenberg. Symmetries on random arrays and set-indexed processes. *Journal of Theoretical Probability*, 5:727–765, 1992.

[35] O. Kallenberg. *Probabilistic Symmetries and Invariance Principles*. Springer, 2005.

[36] R. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Trans. Information Theory*, 56:2980–2998, Jun. 2010. MATLAB code: `http://web.engr.illinois.edu/~swoh/software/optspace/code.html`.

[37] E. Kolaczyk. *Statistical Analysis of Network Data: Methods and Models*. Springer, 2009.

[38] D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[39] D. Lawrence and R. Urtasun. Non-linear matrix factorization with gaussian processes. In *Proc. Intl. Conf. Machine Learning (ICML)*, 2009.

[40] J. Lloyd, P. Orbanz, Z. Ghahramani, and D. Roy. Random function priors

for exchangeable arrays with applications to graphs and relational data. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[41] László Lovász and Balázs Szegedy. Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B*, 96(6):933–957, 2006.

[42] László Lovász and Balázs Szegedy. Szemerédi's lemma for the analyst. *GAFA Geometric And Functional Analysis*, 17(1):252–270, 2007.

[43] Simon Lunagomez and Edoardo Airoldi. Bayesian inference from non-ignorable network sampling designs. *arXiv preprint arXiv:1401.4718*, 2014.

[44] Charles F Manski. Identification of endogenous social effects: The reflection problem. *The review of economic studies*, 60(3):531–542, 1993.

[45] Charles F Manski. *Partial identification of probability distributions*. Springer, 2003.

[46] Charles F Manski. Identification of treatment response with social interactions. *The Econometrics Journal*, 16(1):S1–S23, 2013.

[47] Giovanna Menardi and Adelchi Azzalini. An advancement in clustering via nonparametric density estimation. *Statistics and Computing*, pages 1–15, 2013.

[48] K. Miller, T. Griffiths, and M. Jordan. Nonparametric latent fature models for link prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[49] K. Nowicki and T. Snijders. Estimation and prediction of stochastic blokstructures. *Journal of American Statistical Association*, 96:1077–1087, 2001.

[50] Oliver Riordan. The small giant component in scale-free random graphs. *Combinatorics Probability and Computing*, 14(5/6):897, 2005.

[51] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (< i> p</i>< sup>*</sup>) models for social networks. *Social networks*, 29(2):173–191, 2007.

[52] Karl Rohe, Sourav Chatterjee, Bin Yu, et al. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

[53] D. Roy and Y. Teh. The Mondrian process. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[54] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

[55] Donald B Rubin. Comment: Neyman (1923) and causal inference in experiments and observational studies. *Statistical Science*, 5(4):472–480, 1990.

[56] R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[57] M. Schmidt and M. Morup. Nonparametric Bayesian modeling of complex networks. *IEEE Signal Processing Magaine*, 30:110–128, May 2013.

[58] C. Shalizi and A. Rinaldo. Consistency under sampling of exponential random graph models. *Annals Statistics*, 41(2):508–535, 2013.

[59] Cosma Rohilla Shalizi, Alessandro Rinaldo, et al. Consistency under

sampling of exponential random graph models. *The Annals of Statistics*, 41(2):508–535, 2013.

[60] Tom AB Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100, 1997.

[61] Panos Toulis and Edward Kao. Estimation of causal peer influence effects. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1489–1497, 2013.

[62] Larry Wasserman. *All of nonparametric statistics*. Springer, 2006.

[63] S. Wasserman. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[64] Patrick J Wolfe and Sofia C Olhede. Nonparametric graphon estimation. *arXiv preprint arXiv:1309.5936*, 2013.

[65] Z. Xu, F. Yan, and Y. Qi. Infinite Tucker decomposition: nonparametric Bayesian models for multiway data analysis. In *Proc. Intl. Conf. Machine Learning (ICML)*, 2012.

[66] Yunpeng Zhao, Elizaveta Levina, Ji Zhu, et al. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292, 2012.

A

Appendix to Chapter 2

A.1 Proof of Theorem 3

Proof. Equation (2.5) shows that

$$\mathbb{E}[\hat{r}_{ij}^{k}|u_i, u_j] = r_{ij}.$$

From the definition of $\hat{r}_{ij}^k$ in (2.4), one can bound its variance:

$$Var(\hat{r}_{ij}^k|u_i,u_i) = \frac{Var(G_t[i,k]) * Var(G_t[j,k])}{\left(T - \left\lfloor \frac{T+1}{2} \right\rfloor\right)\left(\left\lfloor \frac{T+1}{2} \right\rfloor\right)} + \frac{Var(G_t[i,k])}{\left(T - \left\lfloor \frac{T+1}{2} \right\rfloor\right)}\mathbb{E}(G[j,k])^2 +$$

$$\frac{Var(G_t[j,k])}{\left(\left\lfloor \frac{T+1}{2} \right\rfloor\right)}\mathbb{E}(G[i,k])^2 = \frac{w(u_i,u_k)(1-w(u_i,u_k))w(u_j,u_k)(1-w(u_j,u_k))}{\left(T - \left\lfloor \frac{T+1}{2} \right\rfloor\right)\left(\left\lfloor \frac{T+1}{2} \right\rfloor\right)} +$$

$$\frac{w(u_i,u_k)(1-w(u_i,u_k))}{\left(T - \left\lfloor \frac{T+1}{2} \right\rfloor\right)}w(u_j,u_k)^2 + \frac{w(u_j,u_k)(1-w(u_j,u_k))}{\left(\left\lfloor \frac{T+1}{2} \right\rfloor\right)}w(u_i,u_k)^2$$

As w is bounded in $[0,1]$,

$$Var(\hat{r}_{ij}^k|u_i,u_j) \leq \frac{1}{\left(T - \left\lfloor \frac{T+1}{2} \right\rfloor\right)\left(\left\lfloor \frac{T+1}{2} \right\rfloor\right)} + \frac{1}{\left(T - \left\lfloor \frac{T+1}{2} \right\rfloor\right)} + \frac{1}{\left(\left\lfloor \frac{T+1}{2} \right\rfloor\right)} \leq \frac{8}{T-1}$$

Define $\hat{r}_{ij}$ as average over all observations $\hat{r}_{ij}^k$, $k \in S_{ij}$, and using Bernstein inequality for bounded random variables with known variance,

$$\mathbb{P}\left(|\hat{r}_{ij} - r_{ij}| > \epsilon\right) \leq 2e^{-\frac{S\epsilon^2}{\frac{16}{T-1} + \frac{2\epsilon}{3}}} \tag{A.1}$$

Since

$$d_{ij} = r_{ii} - r_{ij} - r_{ji} + r_{jj}$$

$$\hat{d}_{ij} = \hat{r}_{ii} - \hat{r}_{ij} - \hat{r}_{ji} + \hat{r}_{jj}$$

we know that

$$|d_{ij} - \hat{d}_{ij}| \leq |r_{ii} - \hat{r}_{ii}| + |r_{ij} - \hat{r}_{ij}| + |r_{ji} - \hat{r}_{ji}| + |r_{jj} - \hat{r}_{jj}|$$

Therefore,

$$\mathbb{P}(|d_{ij} - \hat{d}_{ij}| > \epsilon) \leq \mathbb{P}(|r_{ii} - \hat{r}_{ii}| + |r_{ij} - \hat{r}_{ij}| + |r_{ji} - \hat{r}_{ji}| + |r_{jj} - \hat{r}_{jj}| > \epsilon) \leq$$

$$\mathbb{P}(|r_{ii} - \hat{r}_{ii}| > \frac{\epsilon}{4}) + \mathbb{P}(|r_{ij} - \hat{r}_{ij}| > \frac{\epsilon}{4}) +$$
$$\mathbb{P}(|r_{ji} - \hat{r}_{ji}| > \frac{\epsilon}{4}) + \mathbb{P}(|r_{jj} - \hat{r}_{jj}| > \frac{\epsilon}{4}) \leq$$

$$8e^{-\frac{S\epsilon^2}{\frac{256}{T-1} + \frac{8\epsilon}{3}}}$$

That finishes the proof.

□

A.2 Proof of Theorem 4

Proof. Let $B_1, B_2, \ldots, B_{K'}$ be the blocks of the blockmodel $G(n, w')$, and let $b_1, b_2, \ldots, b_{K'}$ be their respective pivots. Divide each of the intervals $I_1 = (\alpha_0, \alpha_1)$, $\ldots$, $I_Q = (\alpha_{Q-1}, \alpha_Q)$ that define the pieces in which w is Lipschitz in $R = \frac{L\sqrt{2}}{\Delta}$ subintervals of equal size. Clearly, the size of each subinterval is at most $\frac{1}{R}$, because the union of these disjoint subintervals is in $[0, 1]$. Thus, since L is the Lipschitz constant, two points i and j in the same subinterval must satisfy $d_{ij} = \int_0^1 \left(f_i(x) - f_j(x)\right)^2 dx < \left(L\frac{1}{R}\right)^2 = \frac{\Delta^2}{2}$. Supposing that $K' > QR = \frac{QL\sqrt{2}}{\Delta}$, by the pigeonhole principle, there should be at least two pivots b_i and b_j in the same subinterval, for which $d_{b_i b_j} < \frac{\Delta^2}{2}$. But we know, by the algorithm, that the estimated distance between two pivots is at least Δ^2. So $\hat{d}_{b_i b_j} \geq \Delta^2$, and therefore $\hat{d}_{b_i b_j} - d_{b_i b_j} > \frac{\Delta^2}{2}$. Let E be the event that there exists two pivots b_i' and b_j' for which

$\hat{d}_{b_i'b_j'} - d_{b_i'b_j'} > \frac{\Delta^2}{2}$. Clearly, $\mathbb{P}(K' > \frac{QL\sqrt{2}}{\Delta}) \leq \mathbb{P}(E)$, because, as we have just seen, E is a consequence of $K_n' > \frac{QL\sqrt{2}}{\Delta}$. To compute $\mathbb{P}(E)$, remember from Theorem 3 that given, b_i and b_j,

$$\mathbb{P}(|d_{b_ib_j} - \hat{d}_{b_ib_j}| > \frac{\Delta^2}{2}) \leq 8e^{-\frac{S\Delta^4}{\frac{1024}{T-1}+\frac{16\Delta^2}{3}}}, \tag{A.2}$$

So, given $b_1, b_2, \ldots, b_{K'}$,

$$\mathbb{P}(E|b_1, b_2, \ldots, b_{K'}) \leq \sum_{b_ib_j} \mathbb{P}(|d_{b_ib_j} - \hat{d}_{b_ib_j}| > \frac{\Delta^2}{2}) \leq 8n^2e^{-\frac{S\Delta^4}{\frac{1024}{T-1}+\frac{16\Delta^2}{3}}} \tag{A.3}$$

Thus,

$$\mathbb{P}(E) = \sum_{b_1,b_2,\ldots,b_{K'}} \mathbb{P}(E|b_1, b_2, \ldots, b_{K'})\mathbb{P}(b_1, b_2, \ldots, b_{K'}) \leq$$

$$\sum_{b_1,b_2,\ldots,b_{K'}} 8n^2e^{-\frac{S\Delta^4}{\frac{1024}{T-1}+\frac{16\Delta^2}{3}}} \mathbb{P}(b_1, b_2, \ldots, b_{K'}) = 8n^2e^{-\frac{S\Delta^4}{\frac{1024}{T-1}+\frac{16\Delta^2}{3}}}$$

Because $\mathbb{P}(K' > \frac{QL\sqrt{2}}{\Delta}) \leq \mathbb{P}(E)$, we finally have

$$\mathbb{P}(K' > \frac{QL\sqrt{2}}{\Delta}) \leq 8n^2e^{-\frac{S\Delta^4}{\frac{1024}{T-1}+\frac{16\Delta^2}{3}}}$$

□

A.3 Proof of Theorem 5

In this section, we prove theorem 5. As previously, assume that the numbers $\alpha_0 = 0 < \alpha_1 < \alpha_2 < \cdots < \alpha_Q$ define intervals $I_r = (\alpha_{r-1}, \alpha_r)$ such that $w : [0,1]^2 \to [0,1]$ is Lipschitz in each block $I_{ij} = I_i \times I_j$, with Lipschitz constant L. Let $\lambda = min_{i \in \{1,2,\ldots,K\}}(\alpha_i - \alpha_{i-1})$.

We start the proof with the following lemma:

Lemma 17. *For any $i, j \in [0,1]$, define $h_{ij}(\cdot) = (w(i,\cdot) - w(j,\cdot))^2$. Thus, if $0 < \epsilon < 2\lambda L$ is such that $d_{ij} = \int_0^1 h_{ij}(x)dx \leq \frac{\epsilon^2}{8L}$, then $sup_{x \in [0,1]}(h_{ij}(x)) \leq \epsilon$.*

Proof. Fix i and j, and let $h_{ij}^{sup} = \sup_{x \in [0,1]}(h_{ij}(x))$. Let $I_k = (\alpha_{k-1}, \alpha_k)$ be such that there exists a sequence $x_1, x_2, \ldots \in I_k$ satisfying $h_{ij}^{sup} = lim_{n \to \infty} h_{ij}(x_n)$, and define $\lambda_k = |\alpha_k - \alpha_{k-1}|$. For $\theta < \frac{\lambda_k}{2}$, define $h_{ij}^{sup}(\theta) = \sup_{x \in [\alpha_{k-1}+\theta, \alpha_k - \theta]}(h_{ij}(x))$. Clearly $h_{ij}^{sup} = \lim_{\theta \to 0} h_{ij}^{sup}(\theta)$.

The set $[\alpha_{i-1} + \theta, \alpha_i - \theta]$ is compact, so there exists $x_{ij}^{max}(\theta) \in [\alpha_{i-1} + \theta, \alpha_i - \theta]$ such that $h_{ij}^{sup}(\theta) = h_{ij}(x_{ij}^{max}(\theta))$. Assume, without loss of generality, that $x_{ij}^{max}(\theta) + \frac{\lambda_k}{2} - \theta \in [\alpha_{i-1} + \theta, \alpha_i - \theta]$ (if $x_{ij}^{max}(\theta) + \frac{\lambda_k}{2} - \theta > \alpha_i - \theta$, consider $x_{ij}^{max}(\theta) - \frac{\lambda_k}{2} + \theta \in [\alpha_{i-1} + \theta, \alpha_i - \theta]$).

For $0 < \epsilon_0 < \frac{\epsilon}{4L} - \theta \leq \frac{\lambda}{2} - \theta \leq \frac{\lambda_k}{2} - \theta$,

$$\frac{h_{ij}(x_{ij}^{max}(\theta)) - h_{ij}(x_{ij}^{max}(\theta) + \epsilon_0)}{\epsilon_0} =$$

$$\frac{(w(i, x_{ij}^{max}) - w(j, x_{ij}^{max}))^2 - (w(i, x_{ij}^{max}(\theta) + \epsilon_0) - w(j, x_{ij}^{max}(\theta) + \epsilon_0))^2}{\epsilon_0} \leq$$

$$\frac{(w(i, x_{ij}^{max}) - w(j, x_{ij}^{max}))^2 - (w(i, x_{ij}^{max}) + L\epsilon_0 - w(j, x_{ij}^{max}) + L\epsilon_0)^2}{\epsilon_0} \leq$$

$$4L(w(j, x_{ij}^{max}) - w(i, x_{ij}^{max})) \leq 4L$$

A rearrangement of

$$\frac{h_{ij}(x_{ij}^{max}(\theta)) - h_{ij}(x_{ij}^{max}(\theta) + \epsilon_0)}{\epsilon_0} \leq 4L$$

gives us

$$h_{ij}(x_{ij}^{max}(\theta)) - 4L\epsilon_0 \leq h_{ij}(x_{ij}^{max}(\theta) + \epsilon_0)$$

Integrating ϵ_0 in the interval $(0, \frac{\epsilon}{4L} - \theta)$

$$h_{ij}(x_{ij}^{max}(\theta))(\frac{\epsilon}{4L} - \theta) - \frac{4L}{2}(\frac{\epsilon}{4L} - \theta)^2 \leq$$

$$\int_0^{\frac{\epsilon}{4L}-\theta} h_{ij}(x_{ij}^{max}(\theta) + x)dx \leq \int_0^1 h_{ij}(x)dx = d_{ij}$$

Thus,

$$h_{ij}(x_{ij}^{max}(\theta)) \leq \frac{d_{ij}}{\frac{\epsilon}{4L} - \theta} + 2L(\frac{\epsilon}{4L} - \theta)$$

Therefore,

$$h_{ij}^{sup} = \lim_{\theta \to 0} h_{ij}^{sup}(\theta) = \lim_{\theta \to 0} h_{ij}(x_{ij}^{max}(\theta)) \leq \frac{4Ld_{ij}}{\epsilon} + \frac{\epsilon}{2}$$

If $d_{ij} \leq \frac{\epsilon^2}{8L}$,

$$h_{ij}^{sup} \leq \epsilon.$$

Obs: λ_ϵ is a function of w and ϵ, and doesn't depend on the choice of i and j.

□

Proposition 18. *Let $\hat{B}_i = \{i_1, i_2, \ldots, i_{K_i}\}$ and $\hat{B}_j = \{j_1, j_2, \ldots, j_{K_j}\}$ be two clusters given by the algorithm, and suppose that $\{u_{i_1}, u_{i_2}, \ldots, u_{i_{K_i}}\}$ and $\{u_{j_1}, u_{j_2}, \ldots, u_{j_{K_j}}\}$ are the ground truth labels of the vertices in $\hat{B}_i$ and $\hat{B}_j$, respectively. Let $\bar{w}_{ij} = \frac{1}{K_i K_j} \sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} w(u_{x_i}, u_{y_j})$. If the accuracy parameter of the algorithm is such that $\Delta^2 < \lambda^2 L^{1.5}$, then for each $v_i \in B_i$ and $v_j \in B_j$,*

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - \bar{w}_{ij}| > 8\sqrt{\Delta\sqrt{L}}) \leq 32 K_i K_j e^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}}.$$

Proof. By the definition of the algorithm, we know that there are $b_i \in \hat{B}_i$ and $b_j \in \hat{B}_j$, such that, for any other vertices $v_i \in B_i$ and $v_j \in B_j$, $|\hat{d}_{b_i v_i}| \leq \Delta^2$ and $|\hat{d}_{b_j v_j}| \leq \Delta^2$.

$$|\hat{d}_{b_i v_i}| \leq \Delta^2$$

Thus,

$$d_{b_i v_i} \leq d_{b_i v_i} - \hat{d}_{b_i v_i} + \Delta^2 \leq |d_{b_i v_i} - \hat{d}_{b_i v_i}| + \Delta^2$$

Therefore,

$$\mathbb{P}(d_{b_i v_i} > 2\Delta^2) \leq \mathbb{P}(|d_{b_i v_i} - \hat{d}_{b_i v_i}| + \Delta^2 > 2\Delta^2) = \mathbb{P}(|d_{b_i v_i} - \hat{d}_{b_i v_i}| > \Delta^2)$$

From Theorem 3, $\mathbb{P}(|d_{ij} - \hat{d}_{ij}| > \Delta^2) \leq 8e^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}}$. Thus,

$$\mathbb{P}(d_{b_i v_i} > 2\Delta^2) \leq 8e^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}}.$$

Analogously,

$$\mathbb{P}(d_{b_j v_j} > 2\Delta^2) \leq 8e^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}}$$

From the last two equations:

$$\mathbb{P}(d_{b_iv_i} > 2\Delta^2 \text{ or } d_{b_jv_j} > 2\Delta^2) \leq \mathbb{P}(d_{b_jv_j} > 2\Delta^2) + \mathbb{P}(d_{b_iv_i} > 2\Delta^2) \leq 16e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} \tag{A.4}$$

From lemma 17, for any $0 < \left(\frac{\epsilon}{2}\right)^2 < 2\lambda L$, if $d_{b_iv_i} < \frac{\epsilon^4}{128L}$,

$$(w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{v_j}))^2 \leq sup_{x\in[0,1]}((w(u_{v_i}, x) - w(u_{b_i}, x)^2) < \left(\frac{\epsilon}{2}\right)^2$$

Analogously, if $d_{b_jv_j} < \frac{\epsilon^4}{128L}$

$$(w(u_{v_j}, u_{b_i}) - w(u_{b_j}, u_{b_i}))^2 \leq sup_{x\in[0,1]}((w(u_{v_j}, x) - w(u_{b_j}, x))^2) < \left(\frac{\epsilon}{2}\right)^2$$

Thus, if $d_{b_iv_i} < \frac{\epsilon^4}{128L}$ and $d_{b_jv_j} < \frac{\epsilon^4}{128L}$

$$\begin{aligned} |w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j})| \leq |w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{v_j})| + \\ |w(u_{v_j}, u_{b_i}) - w(u_{b_j}, u_{b_i})| < \epsilon. \end{aligned} \tag{A.5}$$

Assuming $\Delta^2 < \lambda^2 L^{1.5}$, making $\epsilon = 4\sqrt{\Delta\sqrt{L}}$ leads to $0 < \epsilon < 4\lambda L$. In that case, by equation (A.5),

$$\left(d_{b_iv_i} < \frac{\epsilon^4}{128L} = 2\Delta^2 \text{ and } d_{b_jv_j} < \frac{\epsilon^4}{128L} = 2\Delta^2\right)$$

Thus,

$$\left(|w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j})| < 4\sqrt{\Delta\sqrt{L}}\right).$$

Therefore,

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j})| > 4\sqrt{\Delta\sqrt{L}}) \leq \mathbb{P}(d_{b_i v_i} > 2\Delta^2 \text{ or } d_{b_j v_j} > 2\Delta^2) \quad \text{(A.6)}$$

From equations (A.4) and (A.6)

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j})| > 4\sqrt{\Delta\sqrt{L}}) \leq 16e^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}} \quad \text{(A.7)}$$

So, for any $x_i \in \hat{B}_i$ and $y_j \in \hat{B}_j$,

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j}) + w(u_{b_i}, u_{b_j}) - w(u_{x_i}, u_{y_j})| > 8\sqrt{\Delta\sqrt{L}}) \leq$$

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j})| > 4\sqrt{\Delta\sqrt{L}}) +$$

$$\mathbb{P}(|w(u_{x_i}, u_{y_j}) - w(u_{b_i}, u_{b_j})| > 4\sqrt{\Delta\sqrt{L}})$$

$$\leq 32e^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}}$$

Therefore,

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{x_i}, u_{x_j})| > 8\sqrt{\Delta\sqrt{L}}) \leq 32e^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}}.$$

Averaging over $x_i \in \hat{B}_i$ and $x_j \in \hat{B}_j$

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - \bar{w}_{ij}| > 8\sqrt{\Delta\sqrt{L}}) =$$

$$\mathbb{P}(|\sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} (w(u_{v_i}, u_{v_j}) - w(u_{x_i}, u_{x_j}))| > K_i K_j 8\sqrt{\Delta\sqrt{L}}) \leq$$

$$\sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} \mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{x_i}, u_{x_j})| > 8\sqrt{\Delta\sqrt{L}}) \leq \sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} 32e^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}}$$

So we finally have

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - \bar{w}_{ij}| > 8\sqrt{\Delta\sqrt{L}}) \leq$$

$$32K_iK_je^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}}.$$

□

Proposition 19. *Let* $\hat{w}_{ij} = \frac{1}{K_iK_j} \sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} \frac{G_1[x_i,y_j]+G_2[x_i,y_j]+\ldots+G_T[x_i,y_j]}{T}$. *Under the conditions of Proposition 18,*

$$\mathbb{P}(|\hat{w}_{ij} - \bar{w}_{ij}| > 16\sqrt{\Delta L}) \leq 4e^{-32TK_iK_j\sqrt{L}\Delta} + 32K_i^2K_j^2e^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}}.$$

Proof. From proposition 18, for any $v_i \in \hat{B}_i$ and $v_j \in \hat{B}_j$

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - \bar{w}_{ij}| > 8\sqrt{\Delta\sqrt{L}}) \leq 32K_iK_je^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}}.$$

Considering all K_iK_j possible values of (v_i, v_j), let E_{ij} be the event that all these values satisfy $|w(u_{v_i}, u_{v_j}) - \bar{w}_{ij}| \leq 8\sqrt{\Delta\sqrt{L}}$, and name $\bar{E}_{ij}$ the complement of E_{ij} . Clearly,

$$\mathbb{P}(\bar{E}_{ij}) \leq \sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} \mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{x_i}, u_{x_j})| > 8\sqrt{\Delta\sqrt{L}}) \leq$$

$$32K_i^2K_j^2e^{-\frac{S\Delta^4}{\frac{256}{(T-1)} + \frac{8\Delta^2}{3}}}. \tag{A.8}$$

Now, assume that E_{ij} has happened and let $\epsilon = 4\sqrt{\Delta\sqrt{L}}$. Then,

$$\bar{w}_{ij}^{-\epsilon} = \bar{w}_{ij} - \epsilon \leq w(u_{v_i}, u_{v_j}) \leq \bar{w}_{ij} + \epsilon = \bar{w}_{ij}^{+\epsilon}$$

Each $G_1[v_i, v_j], G_2[v_i, v_j], \ldots, G_T[v_i, v_j]$ comes from the realization of a Bernoulli variable with mean $w(u_{v_i}, u_{v_j})$. Thus, from the Hoeffding inequality, if we average them over all values of $v_i \in \hat{B}_i$ and $v_j \in \hat{B}_j$,

$$\mathbb{P}(\frac{1}{K_i K_j} \sum_{v_i \in \hat{B}_i, v_j \in \hat{B}_j} \frac{G_1[v_i, v_j] + \ldots + G_T[v_i, v_j]}{T} > \bar{w}_{ij}^{+\epsilon} + \epsilon = \bar{w}_{ij} + 2\epsilon | E_{ij}) \leq$$

$$2e^{-2TK_iK_j\epsilon^2}$$

and,

$$\mathbb{P}(\frac{1}{K_i K_j} \sum_{v_i \in \hat{B}_i, v_j \in \hat{B}_j} \frac{G_1[v_i, v_j] + \ldots + G_T[v_i, v_j]}{T} < \bar{w}_{ij}^{-\epsilon} - \epsilon = \bar{w}_{ij} - 2\epsilon | E_{ij}) \leq$$

$$2e^{-2TK_iK_j\epsilon^2}$$

Thus,

$$\mathbb{P}(|\hat{w}_{ij} - \bar{w}_{ij}| > 2\epsilon | E_{ij}) \leq 4e^{-2TK_iK_j\epsilon^2}$$

Since $\epsilon = 4\sqrt{\Delta\sqrt{L}}$,

$$\mathbb{P}(|\hat{w}_{ij} - \bar{w}_{ij}| > 8\sqrt{\Delta\sqrt{L}} | E_{ij}) \leq 4e^{-32TK_iK_j\sqrt{L}\Delta}$$

From equation (A.8), $\mathbb{P}(\bar{E}_{ij}) < 32K_i^2K_j^2e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}}$. Therefore,

$$\mathbb{P}(|\hat{w}_{ij} - \bar{w}_{ij}| > 8\sqrt{\Delta\sqrt{L}}) \leq \mathbb{P}(|\hat{w}_{ij} - \bar{w}_{ij}| > 8\sqrt{\Delta\sqrt{L}}|E_{ij}) + \mathbb{P}(\bar{E}_{ij}) \leq$$

$$4e^{-32TK_iK_j\sqrt{L}\Delta} + 32K_i^2K_j^2e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}}.$$

□

Proposition 20. *Let $v_i \in \hat{B}_i$ and $v_j \in \hat{B}_j$ be two vertices, and let u_{v_i} and u_{v_j} be their respective ground truth positions in the $[0,1]$ interval. If $\hat{w}_{ij}$ is the estimation for $w(u_{v_i}, u_{v_j})$ provided by the algorithm, then*

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - \hat{w}_{ij}| > 16\sqrt{\Delta\sqrt{L}}) \leq$$

$$4e^{-32TK_iK_j\sqrt{L}\Delta} + 32K_i^2K_j^2e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} + 32K_iK_je^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}}.$$

Proof. From the proposition 18,

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - \bar{w}_{ij}| > 8\sqrt{\Delta\sqrt{L}}) \leq 32K_iK_je^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}}.$$

and from proposition 19

$$\mathbb{P}(|\hat{w}_{ij} - \bar{w}_{ij}| > 8\sqrt{\Delta\sqrt{L}}) \leq 4e^{-32TK_iK_j\sqrt{L}\Delta} + 32K_i^2K_j^2e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}}.$$

Thus,

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - \hat{w}_{ij}| > 16\sqrt{\Delta\sqrt{L}}) \leq$$

$$4e^{-32TK_iK_j\sqrt{L}\Delta} + 32K_i^2K_j^2e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} + 32K_iK_je^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}}.$$

□

Proposition 21. *Let E be a subset of edges (v_i, v_j). Under the above setup, there exists constants c_0 and c_1, that depend only on w, such that*

$$\begin{aligned}\mathbb{P}\left(\frac{1}{|E|}\sum_{v_i,v_j\in E}|w(u_{v_i},u_{v_j})-\hat{w}_{ij}|>c_0\sqrt{\Delta}\right)\leq \\ 64|E|n^4e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}}+\sum_{v_i,v_j\in E}4e^{-c_1TK_iK_j\Delta}\end{aligned} \tag{A.9}$$

Proof. From proposition 20, for any two vertices v_i and v_j,

$$\mathbb{P}(|w(u_{v_i},u_{v_j})-\hat{w}_{ij}|>16\sqrt{\Delta\sqrt{L}})\leq$$

$$4e^{-32TK_iK_j\sqrt{L}\Delta} + 32K_i^2K_j^2e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} + 32K_iK_je^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} \leq$$

$$64n^4e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} + 4e^{-32TK_iK_j\sqrt{L}\Delta}.$$

Averaging the above expression over all pairs $(v_i, v_j) \in E$,

$$\mathbb{P}(\frac{1}{|E|}\sum_{v_i,v_j\in E}|w(u_{v_i},u_{v_j})-\hat{w}_{ij}|>16\sqrt{\Delta\sqrt{L}})\leq$$

$$\sum_{v_i,v_j\in E}\mathbb{P}(|w(u_{v_i},u_{v_j})-\hat{w}_{ij})|>16\sqrt{\Delta\sqrt{L}})\leq$$

$$64|E|n^4e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}}+\sum_{v_i,v_j\in E}4e^{-32TK_iK_j\sqrt{L}\Delta}.$$

Choosing $c_0 = 16L^{1/4}$ and $c_1 = 32\sqrt{L}$,

$$64|E|n^4 e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} + \sum_{v_i,v_j\in E} 4e^{-32TK_iK_j\sqrt{L}\Delta} \leq$$

$$64|E|n^4 e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} + \sum_{v_i,v_j\in E} 4e^{-c_1TK_iK_j\Delta},$$

we finally have

$$\mathbb{P}\left(\frac{1}{|E|}\sum_{v_i,v_j\in E}|w(u_{v_i},u_{v_j})-\hat{w}_{ij}| > c_0\sqrt{\Delta}\right) \leq 64|E|n^4 e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} + \sum_{v_i,v_j\in E} 4e^{-c_1TK_iK_j\Delta}. \tag{A.10}$$

□

Now we are ready to prove Theorem 5.

Proof. Suppose we execute the algorithm in a set of observed graphs with n vertices using parameters Δ and S_n, and call K'_n be the number of blocks generated in the clustering step. Assume that, as n grows, we use a sequence of accuracy and precision parameters satisfying $S_n \in \Theta(n)$ and $\Delta \in \omega\left(\left(\frac{\log(n)}{n}\right)^{\frac{1}{4}}\right) \cap o(1)$.

Our proof is based on proposition 21. The intuition is to show that the three terms $c_0\sqrt{\Delta}$, $64|V|n^4 e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}}$ and $\sum_{v_i,v_j\in V} 4e^{-c_1TK_iK_j\Delta}$ vanish as $n \to \infty$. This is clearly true for the first two terms if we choose $S \in \Theta(n)$ and $\Delta \in \omega\left(\left(\frac{\log(n)}{n}\right)^{\frac{1}{4}}\right) \cap o(1)$. For the third term, it is necessary to consider the size of the clusters the algorithm generates. We prove that the number of small clusters

is asymptotically irrelevant, and that indeed most of the error come from vertices whose cluster is large enough to make $4e^{-c_1 T K_i K_j \Delta}$ vanish.

From the proof of Theorem 4:

$$\mathbb{P}(K'_n > \frac{QL\sqrt{2}}{\Delta}) \leq 8n^2 e^{-\frac{S_n \Delta^4}{\frac{1024}{T^{-1}} + \frac{16\Delta^2}{3}}} \tag{A.11}$$

Let E_n be the event $K'_n \leq \frac{QL\sqrt{2}}{\Delta}$. Note that $S \in \Theta(n)$ and $\Delta \in \omega\left(\left(\frac{\log(n)}{n}\right)^{\frac{1}{4}}\right)$ implies

$$\lim_{n\to\infty} \mathbb{P}(E_n) = 1.$$

Now suppose E_n happens, and define r_n as the number of blocks with less than $\frac{n\Delta^2}{QL\sqrt{2}}$ elements. Let V_n be the union of these blocks, and call $\bar{V}_n$ the complement of V_n, i.e, $\bar{V}_n = V \backslash V_n$. Then

$$|V_n| \leq r_n \frac{n\Delta^2}{QL\sqrt{2}} \leq K'_n \frac{n\Delta^2}{QL\sqrt{2}} \leq n\Delta$$

Thus,

$$\frac{|V_n|}{n} \leq \Delta \tag{A.12}$$

Letting $n \to \infty$

$$\lim_{n\to\infty} \left(\frac{|V_n|}{n}\right) = 0.$$

This limit shows that, as n increases, the set of vertices belonging to small blocks becomes irrelevant in comparison with the total number of vertices.

Looking at the error of estimation,

$$\mathrm{Err}(\hat{w}) = \frac{1}{n^2}\sum_{v_i,v_j\in V}|w(u_i,u_j)-\hat{w}_{ij}| = \frac{|V_n|^2}{n^2}\frac{\sum_{v_i,v_j\in V_n}|w(u_i,u_j)-\hat{w}_{ij}|}{|V_n|^2}+$$

$$+2\frac{|V_n||\bar{V}_n|}{n^2}\frac{\sum_{v_i\in V_n,v_j\in\bar{V}_n}|w(u_i,u_j)-\hat{w}_{ij}|}{|V_n||\bar{V}_n|}+\frac{|\bar{V}_n|^2}{n^2}\frac{\sum_{v_i,v_j\in\bar{V}_n}|w(u_i,u_j)-\hat{w}_{ij}|}{|\bar{V}_n|^2}$$

Using equation A.12 and the fact that $|w(u_i,u_j)-\hat{w}_{ij}|\leq 1$ and $\frac{\bar{V}_n}{n}\leq 1$,

$$\mathrm{Err}(\hat{w}) \leq \frac{|\bar{V}_n|^2}{n^2}\frac{\sum_{v_i,v_j\in\bar{V}_n}|w(u_i,u_j)-\hat{w}_{ij}|}{|\bar{V}_n|^2}+2\Delta+|\Delta^2|$$

Since $\Delta\in o(1)$, for n large enough,

$$\mathrm{Err}(\hat{w}) \leq \frac{|\bar{V}_n|^2}{n^2}\frac{\sum_{v_i,v_j\in\bar{V}_n}|w(u_i,u_j)-\hat{w}_{ij}|}{|\bar{V}_n|^2}+3\Delta \tag{A.13}$$

Therefore, using proposition 21 with $E=\bar{V}_n$:

$$\mathbb{P}(\mathrm{Err}(\hat{w})>c_0\sqrt{\Delta}+3\Delta|E_n)\leq$$

$$\mathbb{P}\left(\frac{|\bar{V}_n|^2}{n^2}\frac{\sum_{v_i,v_j\in\bar{V}_n}|w(u_i,u_j)-\hat{w}_{ij}|}{|\bar{V}_n|^2}+3\Delta>c_0\sqrt{\Delta}+3\Delta\mid E_n\right)\leq$$

$$\frac{1}{\mathbb{P}(E_n)}\mathbb{P}\left(\frac{|\bar{V}_n|^2}{n^2}\frac{\sum_{v_i,v_j\in\bar{V}_n}|w(u_i,u_j)-\hat{w}_{ij}|}{|\bar{V}_n|^2}>c_0\sqrt{\Delta}\right)\leq$$

$$\leq\frac{1}{\mathbb{P}(E_n)}\left(64|\bar{V}_n|n^4e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}}+\sum_{v_i,v_j\in\bar{V}_n}4e^{-c_1TK_iK_j\Delta}\right)$$

Thus

$$\mathbb{P}(\text{Err}(\hat{w}) > \alpha\sqrt{\Delta}|E_n)\mathbb{P}(E_n) \leq \left(64|\bar{V}_n|n^4 e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} + \sum_{v_i,v_j \in \bar{V}_n} 4e^{-c_1 T K_i K_j \Delta} \right)$$

where $\alpha = c_0 + 3 > c_0 + 3\Delta$.

Since $S \in \Theta(n)$ and $\Delta \in \omega\left(\left(\frac{\log(n)}{n}\right)^{\frac{1}{4}}\right) \cap o(1)$, and since each vertex is $\bar{V}_n$ is in a block with at least $\frac{n\Delta^2}{QL\sqrt{2}}$ elements,

$$\lim_{n\to\infty} \mathbb{P}(\text{Err}(\hat{w}) > \alpha\sqrt{\Delta}|E_n)\mathbb{P}(E_n) = 0$$

For any $\epsilon > 0$,

$$\lim_{n\to\infty} \mathbb{P}(\text{Err}(\hat{w}) > \epsilon) = 0, \tag{A.14}$$

because $\lim \Delta = 0$ and $\lim \mathbb{P}(E_n) = 1$.

From the fact that $\hat{w}$ is bounded in $[0, 1]$,

$$\mathbb{E}(\text{Err}(\hat{w})) \leq \epsilon\mathbb{P}(\text{Err}(\hat{w}) \leq \epsilon) + \mathbb{P}(\text{Err}(\hat{w}) > \epsilon)$$

Making $\epsilon \to 0$ and using equation A.14,

$$\lim_{n\to\infty} \mathbb{E}(\text{Err}(\hat{w})) \leq \lim_{n\to\infty} \mathbb{P}(\text{Err}(\hat{w}) > \epsilon) = 0$$

That finishes the proof of part a).

For part b), assuming that $S \in \Theta(n)$ and that Δ is constant, we can use the same arguments as above to prove that the terms $64|V|n^4 e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}}$ and $\sum_{v_i,v_j \in V} 4e^{-c_1 T K_i K_j \Delta}$ of equation A.9 vanish as $n \to \infty$. However, term $c_0\sqrt{\Delta}$

doesn't vanish, and equation (A.14) becomes

$$\lim_{n\to\infty} \mathbb{P}(\mathrm{Err}(\hat{w}) > c_0\sqrt{\Delta}) = 0. \tag{A.15}$$

And the result comes as a direct consequence of this equation.

□

A.4 Proof of theorem 6

Proof.

$$\left|\frac{w_n(u_i,u_j)}{\rho_n} - \frac{\hat{w}_n(i,j)}{\hat{\rho_n}}\right| = \left|\frac{w_n(u_i,u_j)}{\rho_n} - \frac{\hat{w}_n(i,j)}{\rho_n}\frac{\rho_n}{\hat{\rho}_n}\right| =$$

$$\left|\frac{w_n(u_i,u_j)}{\rho_n} - \frac{\hat{w}_n(i,j)}{\rho_n}\left(1 - \frac{\hat{\rho}_n - \rho_n}{\hat{\rho_n}}\right)\right| =$$

$$\left|\frac{w_n(u_i,u_j)}{\rho_n} - \frac{\hat{w}_n(i,j)}{\rho_n} + \frac{\hat{w}_n(i,j)}{\rho_n}\left(\frac{\hat{\rho}_n - \rho_n}{\hat{\rho}_n}\right)\right| \leq$$

$$\left|\frac{w_n(u_i,u_j)}{\rho_n} - \frac{\hat{w}_n(i,j)}{\rho_n}\right| + \left|\frac{\hat{w}_n(i,j)}{\hat{\rho}_n}\left(\frac{\hat{\rho}_n - \rho_n}{\rho_n}\right)\right|$$

Therefore

$$\frac{1}{n^2}\sum_{i,j}\left|\frac{w_n(u_i,u_j)}{\rho_n} - \frac{\hat{w}_n(i,j)}{\hat{\rho}_n}\right| \leq$$

$$\frac{1}{n^2}\left(\sum_{i,j}\left|\frac{w_n(u_i,u_j)}{\rho_n} - \frac{\hat{w}_n(i,j)}{\rho_n}\right| + \left|\frac{\hat{w}_n(i,j)}{\hat{\rho}_n}\left(\frac{\hat{\rho}_n - \rho_n}{\rho_n}\right)\right|\right)$$

Which then becomes:

$$Err_\nu(\hat{\nu}) \leq \frac{Err(\hat{w})}{\rho} + \left(\frac{\hat{\rho} - \rho}{\rho}\right)$$

Since $\rho \in \omega(\sqrt{\Delta})$, we can assume that $\rho > \sqrt{\Delta}$. Thus,

$$Err_{\nu}(\hat{\nu}) \leq \frac{Err(\hat{w}_n)}{\rho_n} + \left(\frac{\hat{\rho}_n - \rho_n}{\sqrt{\Delta}}\right) \tag{A.16}$$

From the proof of Theorem 5,

$$\mathbb{P}(\frac{\text{Err}(\hat{w}_n)}{\rho_n} > \alpha\frac{\sqrt{\Delta}}{\rho_n}) = \mathbb{P}(\text{Err}(\hat{w}) > \alpha\sqrt{\Delta}) \leq$$

$$64|\bar{V}_n|n^4 e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} + \sum_{v_i,v_j\in\bar{V}_n} 4e^{-c_1 TK_iK_j\Delta} \tag{A.17}$$

To estimate $\frac{\hat{\rho}_n-\rho_n}{\sqrt{\Delta}}$, let $deg(i) = \int_0^1 (w_n(u_i, x)dx$ be the expected normalized degree of i and consider $\widehat{deg}(i) = \frac{1}{nT}\sum_j \sum_t G_t[i,j]$ an estimator for $deg(i)$.

For each j, we know that $\mathbb{E}(\sum_T G_t[i,j]) = w_n(u_i,u_j)$ and $Var(\sum_T G_t[i,j]) = \frac{w_n(u_i,u_j)(1-w_n(u_i,u_j))}{T} \leq \frac{1}{T}$. Since $u_j \sim \text{Uniform}(0,1)$, we can now use Bernstein inequality to see that

$$\mathbb{P}\left(|deg(i) - \widehat{deg}(i)| > \epsilon\right) \leq 2e^{-\frac{n\epsilon^2}{\frac{2}{T}+\frac{2\epsilon}{3}}}$$

Summing over the $i's$

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}|deg(i) - \widehat{deg}(i)| > \epsilon\right) \leq 2ne^{-\frac{n\epsilon^2}{\frac{2}{T}+\frac{2\epsilon}{3}}}$$

Which then becomes

$$\mathbb{P}\left(|\frac{1}{n}\sum_{i=1}^{n}deg(i) - \frac{1}{n}\sum_{i=1}^{n}\widehat{deg}(i)| > \epsilon\right) \leq 2ne^{-\frac{n\epsilon^2}{\frac{2}{T}+\frac{2\epsilon}{3}}}$$

And finally,

$$\mathbb{P}\left(|\frac{1}{n}\sum_{i=1}^{n} deg(i) - \hat{\rho}| > \epsilon\right) \leq 2ne^{-\frac{n\epsilon^2}{\frac{2}{T}+\frac{2\epsilon}{3}}} \tag{A.18}$$

By definition, $\rho_n = \int_0^1 deg(x)dx$. Thus, if we make $u_i \sim \text{Uniform}(0,1)$, then $\mathbb{E}(deg(u_i)) = \rho_n$. From Hoeffding inequality, we then have

$$\mathbb{P}\left(|\rho_n - \frac{1}{n}\sum_{i} deg(i)| > \epsilon\right) \leq 2e^{-2n\epsilon^2} \tag{A.19}$$

From (A.18) and (A.19)

$$\mathbb{P}\left(|\rho_n - \frac{1}{n}\sum_{i} deg(i)| + |\frac{1}{n}\sum_{i=1}^{n} deg(i) - \hat{\rho}_n| > 2\epsilon\right) \leq 2e^{-2n\epsilon^2} + 2ne^{-\frac{n\epsilon^2}{\frac{2}{T}+\frac{2\epsilon}{3}}}$$

Thus,

$$\mathbb{P}\left(|\rho_n - \hat{\rho}_n| > 2\epsilon\right) \leq 2e^{-2n\epsilon^2} + 2ne^{-\frac{n\epsilon^2}{\frac{2}{T}+\frac{2\epsilon}{3}}}$$

Which is equivalent to

$$\mathbb{P}\left(\frac{|\rho_n - \hat{\rho}_n|}{\sqrt{\Delta}} > \frac{2\epsilon}{\sqrt{\Delta}}\right) \leq 2e^{-2n\epsilon^2} + 2ne^{-\frac{n\epsilon^2}{\frac{2}{T}+\frac{2\epsilon}{3}}}$$

Making $\epsilon = \Delta$,

$$\mathbb{P}\left(\frac{|\rho_n - \hat{\rho}_n|}{\sqrt{\Delta}} > \sqrt{\Delta}\right) \leq 2e^{-2n\Delta^2} + 2ne^{-\frac{n\Delta^2}{\frac{2}{T}+\frac{2\Delta}{3}}} \tag{A.20}$$

Summing equations (A.17) and (A.20)

$$\mathbb{P}(\frac{\text{Err}(\hat{w}_n)}{\rho_n} > \alpha\frac{\sqrt{\Delta}}{\rho_n}) + \mathbb{P}\left(\frac{|\rho_n - \hat{\rho}_n|}{\sqrt{\Delta}} > \sqrt{\Delta}\right) \leq$$

$$64|\bar{V}_n|n^4 e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} + \sum_{v_i,v_j\in\bar{V}_n} 4e^{-c_1 TK_iK_j\Delta} + 2e^{-2n\Delta^2} + 2ne^{-\frac{n\Delta^2}{\frac{2}{T}+\frac{2\Delta}{3}}}$$

From equation (A.16), we finally have

$$\mathbb{P}(Err_\nu(\hat{\nu}) > \alpha\frac{\sqrt{\Delta}}{\rho_n} + \sqrt{\Delta}) \leq$$

$$64|\bar{V}_n|n^4 e^{-\frac{S\Delta^4}{\frac{256}{(T-1)}+\frac{8\Delta^2}{3}}} + \sum_{v_i,v_j\in\bar{V}_n} 4e^{-c_1 TK_iK_j\Delta} + 2e^{-2n\Delta^2} + 2ne^{-\frac{n\Delta^2}{\frac{2}{T}+\frac{2\Delta}{3}}}$$

The result is a consequence of the facts that $\rho_n \in \omega(\sqrt{\Delta})$ and

$$\Delta \in \omega\left(\left(\frac{\log(n)}{n}\right)^{\frac{1}{4}}\right) \cap o(1)$$

.

□

A.5 Proof of proposition 8

Proof. From theorem 7 and equation (2.23), it is enough to prove that

$$\lambda = \sup\left\{\left(\int_0^1\int_0^1\int_0^1 f(y_1)w(y_1,x)w(x,y_2)f(y_2)dxdy_1dy_2\right)^{\frac{1}{2}} : f \geq 0, ||f||_2 \leq 1\right\} \tag{A.21}$$

Since w is a blockmodel with B blocks, the unit interval $[0,1]$ can be split in

subintervals $U_1, \ldots, U_B$ for which $w(u_i, u_j) = M_{I,J}, \forall u_i \in U_I, u_j \in U_J$. Thus

$$\int_0^1 \int_0^1 \int_0^1 f(y_1)w(y_1, x)w(x, y_2)f(y_2)dxdy_1dy_2 =$$

$$\sum_{I=1}^{B}\sum_{J=1}^{B}\sum_{K=1}^{B} \int_{y_1 \in I} \int_{y_2 \in J} f(y_1)M_{IK}.M_{KJ}.p_K f(y_2)dy_1dy_2 =$$

$$\sum_{I=1}^{B}\sum_{J=1}^{B}\sum_{K=1}^{B} \left(\int_{y_1 \in I} f(y_1)dy_1 \right) M_{IK}.M_{KJ}.p_K \left(\int_{y_2 \in J} f(y_2)dy_2 \right) =$$

$$\sum_{I=1}^{B}\sum_{J=1}^{B}\sum_{K=1}^{B} F_I.M_{IK}.M_{KJ}.F_J.p_I p_J p_K$$

where $F_I = \int_{y \in I} f(y)dy$ is the I-th position of a vector F.

$$\sum_{I=1}^{B}\sum_{J=1}^{B}\sum_{K=1}^{B} F_I.M_{IK}.M_{KJ}.F_J.p_I p_J p_K =$$

$$\sum_{I=1}^{B}\sum_{J=1}^{B}\sum_{K=1}^{B} (p_I^{\frac{1}{2}}.F_I).(p_I^{\frac{1}{2}}.M_{IK}.p_K^{\frac{1}{2}}).(p_K^{\frac{1}{2}}.M_{KJ}.p_J^{\frac{1}{2}}).(p_J^{\frac{1}{2}}.F_J) =$$

$$\hat{F}M'^2\hat{F}^T.$$

where $\hat{F}_I = p_I^{\frac{1}{2}}.F_I$. Also note that

$$||f||_2 = \int_0^1 f(x)^2 dx = \sum_{I=1}^{B} \hat{F}_I^2$$

So, our problem of verifying (A.21) is reduced to solving the optimization problem:

1. Maximize $\hat{F}M'^2\hat{F}^T$.
2. Subject to $\sum_{I=1}^{B} \hat{F}_I^2 = 1$

Clearly, $\hat{F}M'^2\hat{F}^T$ reaches its maximum when $\hat{F}$ is an eigenvector of λ, the maximum eigenvalue (in absolute value) of M'. In that case,

$$\sup\left\{\left(\int_0^1\int_0^1\int_0^1 f(y_1)w(y_1,x)w(x,y_2)f(y_2)dxdy_1dy_2\right)^{\frac{1}{2}} : f \geq 0, ||f||_2 \leq 1\right\} = \lambda$$

And we finish the proof. □

B

Appendix to Chapter 3

B.1 Proof of theorem 9

Proof. From the definition of ξ, we know that

$$w(u_{v_i}, u_{v_k}) - \xi \leq w(u_{v_i'}, u_{v_k}) \leq w(u_{v_i}, u_{v_k}) + \xi$$

Using this fact, observe that

$$w(u_{v_i}, u_{v_k})(w(u_{v_j}, u_{v_k}) - \xi) \leq w(u_{v_i}, u_{v_k})w(u_{v'_j}, u_{v_k}) \leq$$

$$w(u_{v_i}, u_{v_k})(w(u_{v_j}, u_{v_k}) + \xi)$$

Since $w \leq 1$,

$$w(u_{v_i}, u_{v_k})w(u_{v_j}, u_{v_k}) - \xi \leq w(u_{v_i}, u_{v_k})w(u_{v'_j}, u_{v_k}) \leq$$

$$w(u_{v_i}, u_{v_k})w(u_{v_j}, u_{v_k}) + \xi$$

Thus,

$$r_{ij} - \xi = \int_0^1 w(u_{v_i}, u_{v_k})w(u_{v_j}, u_{v_k})du_{v_k} - \xi \leq$$

$$r^*_{ij} \leq \int_0^1 w(u_{v_i}, u_{v_k})w(u_{v_j}, u_{v_k})du_{v_k} + \xi = r_{ij} + \xi$$

Therefore,

$$|r^*_{ij} - r_{ij}| \leq \xi \tag{B.1}$$

But from equation (3.7) we know that

$$\mathbb{P}\left(\max_{ij} |\hat{r}_{ij} - r_{ij}| > \epsilon\right) \leq 2n^2 e^{-2S\epsilon^2}$$

Then,

$$\mathbb{P}\left(|\hat{r}_{ij'} - r_{ij'}| > \epsilon\right) \leq 2n^2 e^{-2S\epsilon^2}$$

Substituting ϵ with $\frac{\epsilon}{4}$,

$$\mathbb{P}\left(|\hat{r}^*_{ij} - r^*_{ij}| > \frac{\epsilon}{4}\right) \leq 2n^2 e^{-\frac{S\epsilon^2}{8}}$$

Thus, using this result and equation (B.1) we finally have:

$$\mathbb{P}\left(|\hat{r}^*_{ij} - r_{ij}| > \frac{\epsilon}{4} + \xi\right) \leq 2n^2 e^{-\frac{S\epsilon^2}{8}}$$

The bias of $\hat{r}^*_{ij}$ is transmitted to the estimator $\hat{d}^*_{ij} = \hat{r}^*_{ii} + \hat{r}^*_{jj} - \hat{r}^*_{ij} - \hat{r}^*_{ji}$ as:

$$\mathbb{P}(|d_{ij} - \hat{d}^*_{ij}| > \epsilon + 4\xi) \leq 8n^2 e^{-\frac{S\epsilon^2}{8}}, \tag{B.2}$$

□

B.2 Proof of theorem 10

Proof. This proof is an adaptation of the proof for theorem 4.

Let $B_1, B_2, \ldots, B_{K^*}$ be the blocks obtained in the clustering step of SBA, and let $b_1, b_2, \ldots, b_{K^*}$ be their respective pivots. Divide each of the intervals $I_1 = (\alpha_0, \alpha_1), \ldots, I_Q = (\alpha_{Q-1}, \alpha_Q)$ that define the pieces in which w is Lipschitz in $R = L\sqrt{\frac{2}{\Delta^2 - 8\xi}}$ subintervals of equal size. Clearly, the size of each subinterval is at most $\frac{1}{R}$, because the union of these disjoint subintervals is in $[0, 1]$. Thus, two points i and j in the same subinterval must satisfy $d_{ij} < \left(\frac{L}{R}\right) = \frac{\Delta^2}{2} - 4\xi$. Supposing that $K^* > Q.R = \frac{QL\sqrt{2}}{\sqrt{\Delta^2 - 8\xi}}$, by the pigeonhole principle, there should be at least two pivots b_i and b_j in the same subinterval, for which $d_{b_i b_j} < \frac{\Delta^2}{2} - 4\xi$. But

we know, by the algorithm, that the estimated distance between two pivots is at least Δ^2. So $\hat{d}_{b_i b_j} \geq \Delta^2$, and therefore $\hat{d}_{b_i b_j} - d_{b_i b_j} > \frac{\Delta^2}{2} + 4\xi$. Let E be the event that there exists two pivots b'_i and b'_j for which $\hat{d}_{b'_i b'_j} - d_{b'_i b'_j} > \frac{\Delta^2}{2} + 4\xi$. Clearly, $\mathbb{P}(K^* > \frac{QL\sqrt{2}}{\sqrt{\Delta^2 - 8\xi}}) \leq \mathbb{P}(E)$, because, as we have just seen, E is a consequence of $K_n^* > \frac{QL\sqrt{2}}{\sqrt{\Delta^2 - 8\xi}}$. To compute $\mathbb{P}(E)$, remember from Theorem 9 that given, b_i and b_j,

$$\mathbb{P}(|d_{b_i b_j} - \hat{d}_{b_i b_j}| > \frac{\Delta^2}{2} + 4\xi) \leq 8n^2 e^{-\frac{S\Delta^4}{32}}, \tag{B.3}$$

So, given $b_1, b_2, \ldots, b_{K'}$,

$$\mathbb{P}(E|b_1, b_2, \ldots, b_{K^*}) \leq \sum_{b_i b_j} \mathbb{P}(|d_{b_i b_j} - \hat{d}_{b_i b_j}| > \frac{\Delta^2}{2} + 4\xi) \leq 2n^4 e^{-\frac{S\Delta^4}{32}}, \tag{B.4}$$

Thus,

$$\mathbb{P}(E) = \sum_{b_1, b_2, \ldots, b_{K'}} \mathbb{P}(E|b_1, b_2, \ldots, b_{K'}) \mathbb{P}(b_1, b_2, \ldots, b_{K'})$$

Using equation (B.4)

$$\mathbb{P}(E) \leq \sum_{b_1, b_2, \ldots, b_{K'}} 2n^2 e^{-\frac{S\Delta^4}{32}} \mathbb{P}(b_1, b_2, \ldots, b_{K^*}) = 2n^4 e^{-\frac{S\Delta^4}{32}}$$

And we finally have,

$$\mathbb{P}(K^* > \frac{QL\sqrt{2}}{\sqrt{\Delta^2 - 8\xi}}) \leq 2n^4 e^{-\frac{S\Delta^4}{32}}$$

□

B.3 Proof of theorem 11

Apply the matching algorithm in a graph G with n vertices, and for any vertex $i \in \{1, \dots, n\}$ call i' the twin of i (remember that the twin of i is defined by $i' \in \operatorname{argmin}_j(\hat{m}_{ij})$). We want to prove that, as n increases, $\xi = \max_{i,k \in \{1,2,\dots,n\}} |w(u_i, u_k) - w(u_{i'}, u_k)|$ vanishes. The proof is based in four observations:

1. $\hat{m}_{ii'}$ is an unbiased estimator for $m_{ii'}$.
2. As n increases, the matching distance $m_{ii'}$ between any vertex i and its twin i' decreases.
3. Small matching distance $m_{ii'}$ implies small similarity distance $d_{ii'}$.
4. Small similarity distance between vertices and its twins implies small ξ.

These steps are approached in the following lemmas.

Lemma 22. *The estimator $\hat{m}_{ij}$ for m_{ij} satisfies*

$$\mathbb{P}\left(|m_{ii'} - \hat{m}_{ii'}| > \epsilon\right) \leq 2n^3 e^{-2(n-2)\epsilon^2}$$

Proof. By definition,

$$\begin{aligned} m_{ij} &= \frac{1}{n-2} \sum_{k \neq i,j} |r_{ik} - r_{jk}| \\ &= \frac{1}{n-2} \sum_{k \neq i,j} \left| \int \left(w(u_i, x) - w(u_j, x)\right) w(u_k, x) dx \right|, \end{aligned} \tag{B.5}$$

and

$$\hat{m}_{ij} = \frac{1}{n-2} \sum_{k \neq i,j} |\hat{r}_{ik} - \hat{r}_{jk}|. \tag{B.6}$$

where

$$\hat{r}_{ij} = \frac{1}{n-2} \sum_{h \neq i,j} \hat{r}^h_{ij} = \frac{1}{n-2} \sum_{h \neq i,j} G[i,h]G[j,h]. \tag{B.7}$$

Clearly,

$$\mathbb{E}(\hat{r}^h_{ik} - \hat{r}^h_{jk} | u_i, u_j, u_h) = \mathbb{E}(G[i,h]G[k,h] - G[j,h]G[k,h]) =$$

$$\big(w(u_i, u_h) - w(u_j, u_h)\big)\, w(u_k, u_h)$$

Since $u_h \sim \text{uniform}(0,1)$,

$$\mathbb{E}(\hat{r}_{ik} - \hat{r}_{jk}) = \int \big(w(u_i, x) - w(u_j, x)\big)\, w(u_k, x) dx = r_{ik} - r_{jk}$$

From the Hoeffding inequality,

$$\mathbb{P}\left(|(\hat{r}_{ik} - \hat{r}_{jk}) - (r_{ik} - r_{jk})| > \epsilon\right) \leq 2e^{-2(n-2)\epsilon^2}$$

Then

$$\mathbb{P}\left(\big||\hat{r}_{ik} - \hat{r}_{jk}| - |r_{ik} - r_{jk}|\big| > \epsilon\right) \leq 2e^{-2(n-2)\epsilon^2}$$

From the definition of m_{ij} and $\hat{m}_{ij}$,

$$\mathbb{P}\left(|m_{ij} - \hat{m}_{ij}| > \epsilon\right) \leq 2ne^{-2(n-2)\epsilon^2}$$

Taking the union over all i, j

$$\mathbb{P}\left(\max_{ij} |m_{ij} - \hat{m}_{ij}| > \epsilon\right) \leq 2n^3 e^{-2(n-2)\epsilon^2}$$

So finally

$$\mathbb{P}\left(|m_{ii'} - \hat{m}_{ii'}| > \epsilon\right) \leq 2n^3 e^{-2(n-2)\epsilon^2}.$$

□

Lemma 23. *For any vertex i and any constant* $0 < \zeta < 1$

$$\mathbb{P}\left(\hat{m}_{ii'} > \epsilon + \frac{L}{n^{1-\zeta}}\right) \leq 2n^3 e^{-2(n-2)\epsilon^2} + e^{-(n-1)^\zeta}$$

Where L is the Lipschitz constant.

Proof. Consider a vertex i and let E_i^1 be the event that there exists no other vertex j such that $|u_i - u_j| < \frac{1}{n^{1-\zeta}}$, for some $0 < \zeta < 1$. Then

$$\mathbb{P}(E_i^1) < (1 - \frac{1}{n^{1-\zeta}})^{n-1} < e^{-\frac{n-1}{n^{1-\zeta}}} < e^{-(n-1)^\zeta} \tag{B.8}$$

So, suppose that E_i^1 didn't happen, and let j be such that $|u_i - u_j| < \frac{1}{n^{1-\zeta}}$. Since w is Lipschitz, $|w(u_i, x) - w(u_j, x)| < \frac{L}{n^{1-\zeta}}$ for any $0 \leq x \leq 1$. Then,

$$|r_{ik} - r_{jk}| = |\int_0^1 \left(w(u_i, x) - w(u_j, x)\right) w(u_k, x) dx| < \frac{L}{n^{1-\zeta}}$$

Thus,

$$m_{ij} < \frac{L}{n^{1-\zeta}}$$

From lemma 22 and equation (B.8), we then know that

$$\mathbb{P}\left(\hat{m}_{ij} > \epsilon + \frac{L}{n^{1-\zeta}} | \overline{E_i^1}\right) \leq 2n^3 e^{-2(n-2)\epsilon^2}$$

where $\overline{E_i^1}$ is the complement of event E_i^1. Therefore

$$\mathbb{P}\left(\hat{m}_{ij} > \epsilon + \frac{L}{n^{1-\zeta}}\right) =$$

$$\mathbb{P}\left(\hat{m}_{ij} > \epsilon + \frac{L}{n^{1-\zeta}} |\overline{E_i^1}|\right) \mathbb{P}(\overline{E_i^1}) + \mathbb{P}\left(\hat{m}_{ij} > \epsilon + \frac{L}{n^{1-\zeta}} |E_i^1|\right) \mathbb{P}(E_i^1) \leq$$

$$2n^3 e^{-2(n-2)\epsilon^2} + e^{-(n-1)^\zeta}$$

By definition $i' \in \text{argmin}_j(\hat{m}_{ij})$, thus

$$\mathbb{P}\left(\hat{m}_{ii'} > \epsilon + \frac{L}{n^{1-\zeta}}\right) \leq 2n^3 e^{-2(n-2)\epsilon^2} + e^{-(n-1)^\zeta}$$

□

Lemma 24. *Let η be a small positive constant is the size of the smallest Lipschitz component of w. Then*

$$\mathbb{P}(m_{ij} < \frac{\eta d_{ij}}{2} - 2L\eta^2) \leq \frac{8}{n^3\eta}$$

Proof. Let Ψ_i be the set of vertices j in the same Lipschitz component as i satisfying $|u_i - u_j| < \eta$. Since the u'_k are chosen Uniform$[0, 1]$,

$$\mathbb{E}\left(\frac{|\Psi_i|}{n}\right) \geq \eta, \text{ var}\left(\frac{|\Psi_i|}{n}\right) = \frac{\eta(1-\eta)}{n}$$

From Chebyshev's inequality,

$$\mathbb{P}\left(\left|\frac{|\Psi_i|}{n} - \eta\right| > \epsilon\right) \leq \left(\frac{\eta(1-\eta)}{n\epsilon^2}\right) \leq \frac{\eta}{n\epsilon^2}$$

using $\epsilon = \frac{\eta n}{2}$

$$\mathbb{P}\left(|\Psi_i| < \frac{\eta n}{2}\right) \leq \frac{4}{n^3\eta}$$

Let E_i^2 be the event that $|\Psi_i| > \frac{\eta n}{2}$ and assume that it has happened.

For each $k \in \Psi_i$, by the Lipschitz property of w, we know that

$$|w(u_j, u_i) - w(u_j, u_k)| \leq L(u_i - u_k) \leq L\eta$$

This implies

$$|r_{ji} - r_{jk}| = |\int_0^1 (w(u_i, x) - w(u_k, x))\, w(u_j, x)dx| < L\eta$$

Therefore,

$$m_{ij} = \frac{1}{n-2}\sum_{k\neq i,j} |r_{ik} - r_{jk}| > \frac{1}{n-2}\sum_{k\in\Psi_i} |r_{ik} - r_{jk}| =$$

$$\frac{1}{n-2}\sum_{k\in\Psi_i} |r_{ik} - r_{ii} + r_{ii} - r_{ij} + r_{ij} - r_{jk}| \geq$$

$$\frac{1}{n-2}\sum_{k\in\Psi_i} \left(|r_{ii} - r_{ij}| - |r_{ik} - r_{ii}| - |r_{ij} - r_{jk}|\right) >$$

$$\frac{1}{n-2}\sum_{k\in\Psi_i} \left(|r_{ii} - r_{ij}| - 2L\eta\right)$$

Thus,

$$m_{ij} > \frac{|\Psi_i|}{n-2}\left(|r_{ii} - r_{ij}| - 2L\eta\right)$$

From our assumption that E_i^2 has happened, this becomes

$$m_{ij} > \frac{\eta n}{2(n-2)} \left(|r_{ii} - r_{ij}| - 2L\eta \right) > \eta |r_{ii} - r_{ij}| - 2L\eta^2$$

Since

$$\mathbb{P}\left(\overline{E_1^2}\right) \leq \frac{4}{n^3 \eta}$$

we have

$$\mathbb{P}(m_{ij} < \eta |r_{ii} - r_{ij}| - 2L\eta^2) \leq \mathbb{P}(\overline{E_i^2}) \leq \frac{4}{n^3 \eta} \tag{B.9}$$

If we apply analogous procedure to j instead of to i, we have

$$\mathbb{P}(m_{ij} < \eta |r_{jj} - r_{ij}| - 2L\eta^2) \leq \frac{4}{n^3 \eta} \tag{B.10}$$

From B.9 and B.10

$$\mathbb{P}(2m_{ij} < \eta |r_{ii} - r_{ij} - r_{ji} + r_{jj}| - 4L\eta^2) \leq$$

$$\mathbb{P}(2m_{ij} < \eta |r_{ii} - r_{ij}| + |r_{jj} + r_{ji}| - 4L\eta^2) \leq$$

$$\mathbb{P}(m_{ij} < \eta |r_{ii} - r_{ij}| - 2L\eta^2) + \mathbb{P}(m_{ij} < |r_{jj} + r_{ji}| - 2L\eta^2) \leq \frac{8}{n^3 \eta}$$

Therefore,

$$\mathbb{P}(m_{ij} < \frac{\eta d_{ij}}{2} - 2L\eta^2) \leq \frac{8}{n^3 \eta}$$

□

Now, let's finish the proof of Theorem 11.

From lemma 24,

$$\mathbb{P}(d_{ij} > \frac{2m_{ij}}{\eta} + 4L\eta) \leq \frac{8}{n^3\eta}$$

From lemma 22

$$\mathbb{P}\left(m_{ij} > \hat{m}_{ij} + \epsilon\right) \leq 2n^3 e^{-2(n-2)\epsilon^2}$$

Putting the last two equations together

$$\mathbb{P}(d_{ij} > \frac{2(\hat{m}_{ij} + \epsilon)}{\eta} + 4L\eta) \leq$$

$$\mathbb{P}(d_{ij} > \frac{2(\hat{m}_{ij} + \epsilon)}{\eta} + 4L\eta, m_{ij} \leq \hat{m}_{ij} + \epsilon) + \mathbb{P}(m_{ij} > \hat{m}_{ij} + \epsilon) \leq$$

$$\mathbb{P}(d_{ij} > \frac{2m_{ij}}{\eta} + 4L\eta, m_{ij} \leq \hat{m}_{ij} + \epsilon) + \mathbb{P}(m_{ij} > \hat{m}_{ij} + \epsilon) \leq$$

$$\mathbb{P}(d_{ij} > \frac{2m_{ij}}{\eta} + 4L\eta) + \mathbb{P}(m_{ij} > \hat{m}_{ij} + \epsilon) \leq$$

$$2n^3 e^{-2(n-2)\epsilon^2} + \frac{8}{n^3\eta}$$

On the other hand,

$$\mathbb{P}\left(d_{ii'} > \frac{2(\hat{m}_{ij} + \epsilon)}{\eta} + 4L\eta\right) \geq$$

$$\mathbb{P}\left(d_{ii'} > \frac{2(\hat{m}_{ij} + \epsilon)}{\eta} + 4L\eta, \hat{m}_{ii'} \leq \epsilon + \frac{L}{n^{1-\zeta}}\right) \geq$$

$$\mathbb{P}\left(d_{ii'} > \frac{2}{\eta}\left(\epsilon + \frac{L}{n^{1-\zeta}}\right) + 4L\eta, \hat{m}_{ii'} \leq 2\epsilon + \frac{L}{n^{1-\zeta}}\right) =$$

$$\mathbb{P}\left(d_{ii'} > \frac{2}{\eta}\left(\epsilon + \frac{L}{n^{1-\zeta}}\right) + 4L\eta\right) -$$

$$\mathbb{P}\left(d_{ii'} > \frac{2}{\eta}\left(\epsilon + \frac{L}{n^{1-\zeta}}\right) + 4L\eta, \hat{m}_{ii'} > 2\epsilon + \frac{L}{n^{1-\zeta}}\right) \geq$$

$$\mathbb{P}\left(d_{ii'} > \frac{2}{\eta}\left(\epsilon + \frac{L}{n^{1-\zeta}}\right) + 4L\eta\right) - \mathbb{P}\left(\hat{m}_{ii'} > 2\epsilon + \frac{L}{n^{1-\zeta}}\right)$$

From lemma 23

$$\mathbb{P}\left(\hat{m}_{ii'} > \epsilon + \frac{L}{n^{1-\zeta}}\right) \leq 2n^3 e^{-2(n-2)\epsilon^2} + e^{-(n-1)^\zeta}$$

Thus the above equation becomes

$$\mathbb{P}\left(d_{ii'} > \frac{2(\hat{m}_{ij} + \epsilon)}{\eta} + 4L\eta\right) \geq$$

$$\mathbb{P}\left(d_{ii'} > \frac{2}{\eta}\left(\epsilon + \frac{L}{n^{1-\zeta}}\right) + 4L\eta\right) - 2n^3 e^{-8(n-2)\epsilon^2} - e^{-(n-1)^\zeta}$$

From previous considerations, we know that

$$\mathbb{P}(d_{ij} > \frac{2(\hat{m}_{ij} + \epsilon)}{\eta} + 4L\eta) \leq$$

$$2n^3 e^{-2(n-2)\epsilon^2} + \frac{8}{n^3\eta}$$

Therefore we finally have

$$\mathbb{P}\left(d_{ii'} > \frac{2}{\eta}\left(\epsilon + \frac{L}{n^{1-\zeta}}\right) + 4L\eta\right) - 2n^3 e^{-8(n-2)\epsilon^2} - e^{-(n-1)^\zeta} \leq$$

$$2n^3 e^{-2(n-2)\epsilon^2} + \frac{8}{n^3\eta}$$

Which then becomes,

$$\mathbb{P}\left(d_{ii'} > \frac{2}{\eta}\left(\epsilon + \frac{L}{n^{1-\zeta}}\right) + 4L\eta\right) \leq$$

$$4n^3 e^{-8(n-2)\epsilon^2} + e^{-(n-1)^\zeta} + \frac{8}{n^3\eta}$$

For this probability to vanish as $n \to \infty$, we need ϵ^2 to decrease slower than $\sqrt{\frac{1}{n-2}}$. By making $\epsilon = \frac{1}{n^{(1-\zeta)/2}}$ the above equation becomes

$$\mathbb{P}\left(d_{ii'} > \frac{2}{\eta}\left(\frac{1}{n^{(1-\zeta)/2}} + \frac{L}{n^{1-\zeta}}\right) + 4L\eta\right) \leq$$

$$4n^3 e^{-8\frac{n-2}{n}n^\zeta} + e^{-(n-1)^\zeta} + \frac{8}{n^3\eta}$$

Since $\frac{1}{n^{(1-\zeta)/2}}$ dominates $\frac{L}{n^{1-\zeta}}$ we simplify this to

$$\mathbb{P}\left(d_{ii'} > \frac{4}{\eta n^{(1-\zeta)/2}} + 4L\eta\right) \leq$$

$$4n^3 e^{-8\frac{n-2}{n}n^\zeta} + e^{-(n-1)^\zeta} + \frac{8}{n^3\eta}$$

For choosing η, notice that the right hand side of the expression inside the probability vanishes with best asymptotics rates when $4n^3 e^{-8\frac{n-2}{n}n^\zeta}$ and $e^{-(n-1)^\zeta} + \frac{8}{n^3\eta}$ have similar asymptotics, since the product of these two expressions don't depend on ζ. Let's choose $\eta = \frac{1}{L^{1/2} n^{\frac{1-\zeta}{4}}}$ and so we have

$$\mathbb{P}\left(d_{ii'} > \frac{8L^{1/2}}{n^{\frac{1-\zeta}{4}}}\right) \leq$$

$$4n^3e^{-8\frac{n-2}{n}n^\zeta} + e^{-(n-1)^\zeta} + 8L^{1/2}n^{-\frac{11-\zeta}{4}}$$

Now the result come as a consequence of lemma 17. The lemma says that for any $0 < \epsilon < c_w$, where c_w is a constant that depends only on the graphon, if $d_{ij} \leq \frac{\epsilon^2}{8L}$ then $\sup_{x\in[0,1]} (w(i,\cdot) - w(j,\cdot))^2 \leq \epsilon$. This essentially means that if d_{ij} is small enough then $\sup_{x\in[0,1]} |w(i,\cdot) - w(j,\cdot)| \leq \left(8Ld_{ij}\right)^{1/4}$. In that case,

$$\mathbb{P}\left(\sup_{x\in[0,1]} |w(i,\cdot) - w(j,\cdot)|^4 \frac{1}{8L^{1/2}} > \frac{8L^{1/2}}{n^{\frac{1-\zeta}{4}}}\right) \leq$$

$$4n^3e^{-8\frac{n-2}{n}n^\zeta} + e^{-(n-1)^\zeta} + 8L^{1/2}n^{-\frac{11-\zeta}{4}}$$

Thus,

$$\mathbb{P}\left(\sup_{x\in[0,1]} |w(i,\cdot) - w(j,\cdot)| > (64L)^{\frac{1}{4}}n^{-\frac{1-\zeta}{8}}\right) \leq$$

$$4n^3e^{-8\frac{n-2}{n}n^\zeta} + e^{-(n-1)^\zeta} + 8L^{1/2}n^{-\frac{11-\zeta}{4}}$$

Since $\xi = \max_{i,k\in\{1,2,\ldots,n\}} |w(u_i,u_k) - w(u_{i'},u_k)|$,

$$\mathbb{P}\left(\xi > (64L)^{\frac{1}{4}}n^{-\frac{1-\zeta}{8}}\right) \leq$$

$$4n^5e^{-8\frac{n-2}{n}n^\zeta} + n^2e^{-(n-1)^\zeta} + 8L^{1/2}n^{-\frac{3+\zeta}{4}}$$

B.4 Proof of theorem 12

Proof of Theorem 12 is based on the following propositions 25, 26, 27 and 28 , which are respective analogous of propositions 18, 19, 20 and 21. We provide a proof for proposition 25, but skip the details for 26, 27 and 28 because, given

proposition 25, they are straightforward adaptations of propositions 19, 20 and 21.

Proposition 25. *Let $\hat{B}_i = \{i_1, i_2, \ldots, i_{K_i}\}$ and $\hat{B}_j = \{j_1, j_2, \ldots, j_{K_j}\}$ be two clusters given by the algorithm, and suppose that $\{u_{i_1}, u_{i_2}, \ldots, u_{i_{K_i}}\}$ and $\{u_{j_1}, u_{j_2}, \ldots, u_{j_{K_j}}\}$ are the ground truth labels of the vertices in $\hat{B}_i$ and $\hat{B}_j$, respectively. Let $\bar{w}_{ij} = \frac{1}{K_i K_j} \sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} w(u_{x_i}, u_{y_j})$. If the accuracy parameter of the algorithm is such that $\Delta^2 < \lambda^2 L^{1.5}$, then for each $v_i \in B_i$ and $v_j \in B_j$,*

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - \bar{w}_{ij}| > 8((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq 32n^2 K_i K_j e^{-\frac{S\Delta^4}{32}}.$$

Proof. By the definition of the algorithm, we know that there are $b_i \in \hat{B}_i$ and $b_j \in \hat{B}_j$, such that, for any other vertices $v_i \in B_i$ and $v_j \in B_j$, $|\hat{d}_{b_i v_i}| \leq \Delta^2$ and $|\hat{d}_{b_j v_j}| \leq \Delta^2$. Then

$$d_{b_i v_i} \leq d_{b_i v_i} - \hat{d}_{b_i v_i} + \Delta^2 \leq |d_{b_i v_i} - \hat{d}_{b_i v_i}| + \Delta^2$$

Which implies

$$\mathbb{P}(d_{b_i v_i} > 2\Delta^2 + 4\xi) \leq \mathbb{P}(|d_{b_i v_i} - \hat{d}_{b_i v_i}| + \Delta^2 > 2\Delta^2 + 4\xi) =$$

$$\mathbb{P}(|d_{b_i v_i} - \hat{d}_{b_i v_i}| > \Delta^2 + 4\xi)$$

From Theorem 9, $\mathbb{P}(|d_{ij} - \hat{d}_{ij}| > \Delta^2 + 4\xi) \leq 8n^2 e^{-\frac{S\Delta^4}{8}}$. Thus,

$$\mathbb{P}(d_{b_i v_i} > 2\Delta^2 + 4\xi) \leq 8n^2 e^{-\frac{S\Delta^4}{8}}.$$

Analogously,

$$\mathbb{P}(d_{b_j v_j} > 2\Delta^2 + 4\xi) \leq 8n^2 e^{-\frac{S\Delta^4}{8}}$$

$$\begin{aligned} \mathbb{P}(d_{b_i v_i} > 2\Delta^2 + 4\xi \; or \; d_{b_j v_j} > 2\Delta^2 + 4\xi) \leq \\ \mathbb{P}(d_{b_j v_j} > 2\Delta^2 + 4\xi) + \mathbb{P}(d_{b_i v_i} > 2\Delta^2 + 4\xi) \leq 16n^2 e^{-\frac{S\Delta^4}{8}} \end{aligned} \tag{B.11}$$

From lemma 17, for any $0 < \left(\frac{\epsilon}{2}\right)^2 < 2\lambda L$, if $d_{b_i v_i} < \frac{\epsilon^4}{128L}$,

$$(w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{v_j}))^2 \leq sup_{x \in [0,1]}((w(u_{v_i}, x) - w(u_{b_i}, x))^2) < \left(\frac{\epsilon}{2}\right)^2$$

Analogously, if $d_{b_j v_j} < \frac{\epsilon^4}{128L}$

$$(w(u_{v_j}, u_{b_i}) - w(u_{b_j}, u_{b_i}))^2 \leq sup_{x \in [0,1]}((w(u_{v_j}, x) - w(u_{b_j}, x))^2) < \left(\frac{\epsilon}{2}\right)^2$$

Thus, if $d_{b_i v_i} < \frac{\epsilon^4}{128L}$ and $d_{b_j v_j} < \frac{\epsilon^4}{128L}$

$$\begin{aligned} |w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j})| \leq \\ |w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{v_j})| + |w(u_{v_j}, u_{b_i}) - w(u_{b_j}, u_{b_i})| < \epsilon. \end{aligned} \tag{B.12}$$

Assuming $\Delta^2 + 2\xi < \lambda^2 L^{1.5}$, making $\epsilon = 4((\Delta^2 + 2\xi)L)^{\frac{1}{4}}$ leads to $0 < \epsilon < 4\lambda L$. In that case, by equation (B.12), the following event

$$\left(d_{b_i v_i} < \frac{\epsilon^4}{128L} = 2\Delta^2 + 4\xi \text{ and } d_{b_j v_j} < \frac{\epsilon^4}{128L} = 2\Delta^2 + 4\xi\right)$$

implies

$$\left(|w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j})| < 4((\Delta^2 + 2\xi)L)^{\frac{1}{4}}\right).$$

Therefore,

$$\begin{aligned}\mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j})| > 4((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq \\ \mathbb{P}(d_{b_i v_i} > 2\Delta^2 + 4\xi \text{ or } d_{b_j v_j} > 2\Delta^2 + 4\xi)\end{aligned} \tag{B.13}$$

From equations (B.11) and (B.13)

$$\mathbb{P}(d_{b_i v_i} > 2\Delta^2 + 2\xi \text{ or } d_{b_j v_j} > 2\Delta^2 + 4\xi) \leq 16n^2 e^{-\frac{S\Delta^4}{8}}$$

Thus

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j})| > 4((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq 16n^2 e^{-\frac{S\Delta^4}{8}} \tag{B.14}$$

So, for any $x_i \in \hat{B}_i$ and $y_j \in \hat{B}_j$,

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j}) + w(u_{b_i}, u_{b_j}) - w(u_{x_i}, u_{y_j})| > 8((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq$$

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{b_i}, u_{b_j})| > 4((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) +$$

$$+\mathbb{P}(|w(u_{x_i}, u_{y_j}) - w(u_{b_i}, u_{b_j})| > 4((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq 32e^{-\frac{S\Delta^4}{8}}$$

Thus

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{x_i}, u_{x_j})| > 8((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq 32e^{-\frac{S\Delta^4}{8}}.$$

Averaging over $x_i \in \hat{B}_i$ and $x_j \in \hat{B}_j$

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - \bar{w}_{ij}| > 8((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) =$$

$$= \mathbb{P}(|\sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} (w(u_{v_i}, u_{v_j}) - w(u_{x_i}, u_{x_j}))| > K_i K_j 8((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq$$

$$\mathbb{P}(\sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} |(w(u_{v_i}, u_{v_j}) - w(u_{x_i}, u_{x_j})| > K_i K_j 8((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq$$

$$\sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} \mathbb{P}(|w(u_{v_i}, u_{v_j}) - w(u_{x_i}, u_{x_j})| > 8((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq \sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} 32n^2 e^{-\frac{S\Delta^4}{8}}$$

Therefore

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - \bar{w}_{ij}| > 8((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq 32n^2 K_i K_j e^{-\frac{S\Delta^4}{8}}.$$

□

Proposition 26. *Let* $\hat{w}_{ij} = \frac{1}{K_i K_j} \sum_{x_i \in \hat{B}_i, y_j \in \hat{B}_j} G[x_i, y_j]$. *Under the conditions of Proposition 25,*

$$\mathbb{P}(|\hat{w}_{ij} - \bar{w}_{ij}| > 8((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq 4n^2 e^{-128K_i K_j \sqrt{(\Delta^2 + 2\xi)L}} + 32n^2 K_i^2 K_j^2 e^{-\frac{S\Delta^4}{8}}.$$

Proposition 27. *Let* $v_i \in \hat{B}_i$ *and* $v_j \in \hat{B}_j$ *be two vertices, and let* u_{v_i} *and* u_{v_j} *be their respective ground truth positions in the* $[0, 1]$ *interval. If* $\hat{w}_{ij}$ *is the estimation for* $w(u_{v_i}, u_{v_j})$ *provided by the algorithm, then*

$$\mathbb{P}(|w(u_{v_i}, u_{v_j}) - \hat{w}_{ij}| > 16((\Delta^2 + 2\xi)L)^{\frac{1}{4}}) \leq$$

$$4n^2 e^{-128K_i K_j \sqrt{(\Delta^2 + 2\xi)L}} + 32n^2 K_i^2 K_j^2 e^{-\frac{S\Delta^4}{8}} + 32n^2 K_i K_j e^{-\frac{S\Delta^4}{8}}.$$

Proposition 28. *Let E be a subset of edges (v_i, v_j). Under the above setup, there exist constants c_0 and c_1, that depends only on w, such that*

$$\mathbb{P}\left(\frac{1}{|E|}\sum_{v_i,v_j\in E}|w(u_{v_i},u_{v_j})-\hat{w}_{ij}|>16((\Delta^2+2\xi)L)^{\frac{1}{4}}\right)\leq 4|E|(n)^6e^{-\frac{S\Delta^4}{8}}+\sum_{v_i,v_j\in E}4n^2e^{-c_1K_iK_j\sqrt{(\Delta^2+2\xi)L}} \tag{B.15}$$

To finish the proof of theorem 12, one has to follow the similar steps of theorem 5, but now using propositions 25, 26, 27, 28 and theorem 10. The only major change is that one needs to attempt to the fact that 10 requires $\Delta > \sqrt{8\xi}$, since, in our theorem 11 ξ approaches 0 with rate $(64L)^{\frac{1}{4}}n^{-\frac{1-\zeta}{8}}$, we now require $\omega\left(n^{-\frac{1}{16}}\right)\cap o(1)$ instead of $\omega\left(\left(\frac{\log(n)}{n}\right)^{\frac{1}{4}}\right)\cap o(1)$ in 5.

C

Appendix to Chapter 4

C.1 Proof of Theorem 16

Proof. Let $z_i = y_i - \alpha - \beta t_i$, and μ_I and ς_I^2 be as described in proposition 14, for DIRG models, or proposition15, for AIRG models. In either case,

$$P(\{\gamma s_I(v_i)\}_{I, v_i \in B_I} | A_1, \ldots, A_q, F, M) \cong \prod_I \prod_{v_i \in B_I} \mathrm{N}\left(\gamma s_I(v_i); \gamma \mu_I, \gamma^2 \varsigma_I^2\right),$$

where s_I is the social component, i.e., $s_i = p_i$, if the model is DIRG, or $s_i = r_i$, if the model is AIRG.

Thus, the probability of z_i given parameters γ and $\{\sigma_I\}_I$, the output of SBA, and the probabilities of treatment f_i (which are used to compute μ_I and ς_I) is:

$$p(z_i|\gamma, \{\sigma_I\}_I, A_1, \ldots, A_q, F, M) =$$

$$\prod_i \int \frac{1}{\sqrt{2\pi\sigma_I^2}} \exp\left(-\frac{(z_i - \gamma s_i)^2}{\sigma_I^2}\right) \frac{1}{\sqrt{2\pi\varsigma_I^2}} \exp\left(-\frac{(\gamma s_i - \gamma\mu_i)^2}{2\gamma^2\varsigma_I^2}\right) \frac{1}{\gamma} d\gamma s_i =$$

$$\prod_i \int \frac{1}{2\pi\varsigma_I\sigma\gamma} \exp\left(-\frac{(z_i - \gamma s_i)^2}{2\sigma_I^2} - \frac{(\gamma s_i - \gamma\mu_i)^2}{2\gamma^2\varsigma_I^2}\right) d\gamma s_i =$$

$$\prod_i \int \frac{1}{2\pi\varsigma_I\sigma\gamma} \exp\left(-\gamma^2 s_i^2 \left(\frac{1}{2\sigma_I^2} + \frac{1}{2\gamma^2\varsigma_I^2}\right) + 2\gamma s_i \left(\frac{z_i}{2\sigma_I^2} + \frac{\gamma\mu_i}{2\gamma^2\varsigma^2}\right)\right.$$

$$\left. - \left(\frac{z_i^2}{2\sigma_I^2} + \frac{\gamma^2\mu_i^2}{2\gamma^2\varsigma_I^2}\right)\right) d\gamma s_i =$$

$$\prod_i \left[\frac{1}{\sqrt{2\pi\varsigma_I^2\sigma^2\gamma^2\left(\frac{1}{\sigma_I^2} + \frac{1}{\gamma^2\varsigma^2}\right)}} \exp\left(\frac{\left(\frac{z_i}{2\sigma_I^2} + \frac{\gamma\mu_i}{2\gamma^2\varsigma_I^2}\right)^2}{\left(\frac{1}{2\sigma_I^2} + \frac{1}{2\gamma^2\varsigma_I^2}\right)} - \left(\frac{z_i^2}{2\sigma_I^2} + \frac{\mu_i^2}{2\varsigma_I^2}\right)\right)\right.$$

$$\left. \int \frac{1}{\sqrt{2\pi\left(\frac{1}{\sigma_I^2} + \frac{1}{\gamma^2\varsigma^2}\right)^{-1}}} \exp\left(-\frac{\left(\gamma s_i - \left(\frac{z_i}{\sigma_I^2} + \frac{\gamma\mu_i}{\gamma^2\varsigma^2}\right)\left(\frac{1}{\sigma_I^2} + \frac{1}{\gamma^2\varsigma^2}\right)^{-1}\right)^2}{2\left(\left(\frac{1}{\sigma_I^2} + \frac{1}{\gamma^2\varsigma^2}\right)^{-1/2}\right)^2}\right) d\gamma s_i \right]$$

Since the last line is the probability function of a normal, its integral is 1. Thus

$$p(z_i|\gamma, \{\sigma_I\}_I, A_1, \ldots, A_q, F, M) =$$

$$\prod_i \left[\frac{1}{\sqrt{2\pi\left(\gamma^2\varsigma_I^2 + \sigma_I^2\right)}} \exp\left(\frac{\left(\frac{z_i\gamma\varsigma_I}{\sigma_I} + \frac{\mu_i\sigma_I}{\varsigma_I}\right)^2}{2\left(\gamma^2\varsigma_I^2 + \sigma_I^2\right)} - \left(\frac{z_i^2}{2\sigma_I^2} + \frac{\mu_i^2}{2\varsigma_I^2}\right)\right)\right]$$

Taking the $\log$

$$\log p(z_i|\gamma, \{\sigma_I\}_I, A_1, \ldots, A_q, F, M) = \sum_i \log\left(\frac{1}{\sqrt{2\pi\left(\gamma^2\varsigma_I^2 + \sigma_I^2\right)}}\right) +$$

$$\sum_i \left(\frac{\left(\frac{z_i\gamma\varsigma_I}{\sigma_I} + \frac{\mu_i\sigma_I}{\varsigma_I}\right)^2}{2\left(\gamma^2\varsigma_I^2 + \sigma_I^2\right)} - \left(\frac{z_i^2}{2\sigma_I^2} + \frac{\mu_i^2}{2\varsigma_I^2}\right)\right) =$$

$$\sum_i \log\left(\frac{1}{\sqrt{2\pi\left(\gamma^2\varsigma_I^2 + \sigma_I^2\right)}}\right) +$$

$$\sum_i \left(\frac{\left(\frac{(y_i-\alpha-\beta t_i)\gamma\varsigma_I}{\sigma_I} + \frac{\mu_i\sigma_I}{\varsigma_I}\right)^2}{2\left(\gamma^2\varsigma_I^2 + \sigma_I^2\right)} - \left(\frac{(y_i-\alpha-\beta t_i)^2}{2\sigma_I^2} + \frac{\mu_i^2}{2\varsigma_I^2}\right)\right)$$

The gradient of the $\log$ with respect to parameters α, β and γ is

$$\nabla \log p(z_i|\gamma, \{\sigma_I\}_I, A_1, \ldots, A_q, F, M) =$$

$$\sum_i \begin{pmatrix} -\gamma\frac{\left(\frac{(y_i-\alpha-\beta t_i)\gamma\varsigma_I^2}{\sigma_I^2}+\mu_I\right)}{\left(\gamma^2\varsigma_I^2+\sigma_I^2\right)} + \frac{(y_i-\alpha-\beta t_i)}{\sigma_I^2} \\ -t_i\gamma\frac{\left(\frac{(y_i-\alpha-\beta t_i)\gamma\varsigma_I^2}{\sigma_I^2}+\mu_I\right)}{\left(\gamma^2\varsigma_I^2+\sigma_I^2\right)} + t_i\frac{(y_i-\alpha-\beta t_i)}{\sigma_I^2} \\ -\frac{\gamma\varsigma_I^2}{\gamma^2\varsigma_I^2+\sigma_I^2} + \frac{\left(\frac{(y_i-\alpha-\beta t_i)^2\gamma\varsigma_I^2}{\sigma_I^2}+\mu(y_i-\alpha-\beta t_i)\right)}{\left(\gamma^2\varsigma_I^2+\sigma_I^2\right)} - \frac{\varsigma_I^2\gamma\left(\frac{(y_i-\alpha-\beta t_i)\gamma\varsigma_I}{\sigma_I}+\frac{\mu_i\sigma_I}{\varsigma_I}\right)^2}{\left(\gamma^2\varsigma_I^2+\sigma_I^2\right)^2} \end{pmatrix}$$

The Herssian matrix is then

$$\frac{1}{n}\sum_{i=1}^{q}\begin{pmatrix} L^* & C^* & D^* \\ C^* & O^* & E^* \\ D^* & E^* & F^* \end{pmatrix},$$

Where

$$L^* = \frac{1}{n}\sum_i \frac{1}{\sigma_I^2}\left(\frac{\gamma^2\varsigma_I^2}{\left(\gamma^2\varsigma_I^2 + \sigma_I^2\right)} - 1\right)$$

$$C^* = \frac{1}{n}\sum_i \frac{t_i}{\sigma_I^2}\left(\frac{\gamma^2\varsigma_I^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)} - 1\right)$$

$$O^* = \frac{1}{n}\sum_i \frac{t_i^2}{\sigma_I^2}\left(\frac{\gamma^2\varsigma_I^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)} - 1\right)$$

$$D^* = \frac{1}{n}\sum_i \left(-\frac{\left(\frac{2(y_i-\alpha-\beta t_i)\gamma\varsigma_I^2}{\sigma_I^2}+\mu_I\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)} + 2\gamma^2\varsigma_I^2\frac{\left(\frac{(y_i-\alpha-\beta t_i)\gamma\varsigma_I^2}{\sigma_I^2}+\mu_I\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2}\right)$$

$$E^* = \frac{1}{n}\sum_i t_i\left(-\frac{\left(\frac{2(y_i-\alpha-\beta t_i)\gamma\varsigma_I^2}{\sigma_I^2}+\mu_I\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)} + 2\gamma^2\varsigma_I^2\frac{\left(\frac{(y_i-\alpha-\beta t_i)\gamma\varsigma_I^2}{\sigma_I^2}+\mu_I\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2}\right)$$

$$-\frac{\gamma\varsigma_I^2}{\gamma^2\varsigma_I^2+\sigma_I^2} + \frac{\left(\frac{(y_i-\alpha-\beta t_i)^2\gamma\varsigma_I^2}{\sigma_I^2}+\mu(y_i-\alpha-\beta t_i)\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)} - \frac{\varsigma_I^2\gamma\left(\frac{(y_i-\alpha-\beta t_i)\gamma\varsigma_I}{\sigma_I}+\frac{\mu_i\sigma_I}{\varsigma_I}\right)^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2}$$

$$F^* = \frac{2\gamma^2\varsigma_I^4}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2} - \frac{\varsigma_I^2}{\gamma^2\varsigma_I^2+\sigma_I^2}$$

$$+\frac{\left(\frac{(y_i-\alpha-\beta t_i)^2\varsigma_I^2}{\sigma_I^2}\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)} - 2\gamma\varsigma_I^2\frac{\left(\frac{(y_i-\alpha-\beta t_i)^2\gamma\varsigma_I^2}{\sigma_I^2}+\mu(y_i-\alpha-\beta t_i)\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2}$$

$$-\frac{2\varsigma_I^3(y_i-\alpha-\beta t_i)\gamma\left(\frac{(y_i-\alpha-\beta t_i)\gamma\varsigma_I}{\sigma_I}+\frac{\mu_i\sigma_I}{\varsigma_I}\right)}{\sigma_I(\gamma^2\varsigma_I^2+\sigma_I^2)^2} - \frac{\varsigma_I^2\left(\frac{(y_i-\alpha-\beta t_i)\gamma\varsigma_I}{\sigma_I}+\frac{\mu_i\sigma_I}{\varsigma_I}\right)^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2}$$

$$+\frac{2\varsigma_I^4\gamma^2\left(\frac{(y_i-\alpha-\beta t_i)\gamma\varsigma_I}{\sigma_I}+\frac{\mu_i\sigma_I}{\varsigma_I}\right)^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)^3} =$$

$$\frac{\gamma^2\varsigma_I^4-\varsigma_I^2\sigma_I^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2} + \frac{(y_i-\alpha-\beta t_i)^2\varsigma_I^2}{\sigma^2(\gamma^2\varsigma_I^2+\sigma_I^2)}$$

$$-4\gamma\varsigma_I^2\frac{\left(\frac{(y_i-\alpha-\beta t_i)^2\gamma\varsigma_I^2}{\sigma_I^2}+\mu(y_i-\alpha-\beta t_i)\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2} + \frac{(\gamma^2\varsigma_I^2-\sigma_I^2)\left(\frac{(y_i-\alpha-\beta t_i)\gamma\varsigma_I^2}{\sigma_I}+\mu_i\sigma_I\right)^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)^3}$$

Taking the expectations and evaluating the sums to find the Fisher information

matrix:

$$L = \frac{1}{n}\sum_I \frac{A_I}{\sigma_I^2}\left(\frac{\gamma^2\varsigma_I^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)} - 1\right)$$

$$C = O = \frac{1}{n}\sum_I \frac{A_I f_I}{\sigma_I^2}\left(\frac{\gamma^2\varsigma_I^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)} - 1\right)$$

$$D = \frac{1}{n}\sum_I \left(-\frac{\left(\frac{2\mu\gamma^2\varsigma_I^2}{\sigma_I^2}+\mu_I\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)} + 2\gamma^2\varsigma_I^2\frac{\left(\frac{\mu\gamma^2\varsigma_I^2}{\sigma_I^2}+\mu_I\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2}\right) =$$

$$\frac{1}{n}\sum_I \left(-\frac{\left(\frac{2\mu\gamma^2\varsigma_I^2}{\sigma_I^2}+\mu_I\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)} + \frac{2\gamma^2\varsigma_I^2\mu_I}{\sigma^2\left(\gamma^2\varsigma_I^2+\sigma_I^2\right)}\right) = \frac{1}{n}\sum_I \frac{\mu_I}{(\gamma^2\varsigma_I^2+\sigma_I^2)} =$$

$$-\frac{1}{n}\sum_I \frac{A_I\mu_I}{(\gamma^2\varsigma_I^2+\sigma_I^2)}$$

$$E = -\frac{1}{n}\sum_I \frac{A_I\mu_I f_I}{(\gamma^2\varsigma_I^2+\sigma_I^2)}$$

$$F = \frac{1}{n}\sum_I \left[\frac{\gamma^2\varsigma_I^4 - \varsigma_I^2\sigma_I^2}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2} + \frac{(\varsigma_I^2+\mu_I^2)^2\gamma^2\varsigma_I^2}{\sigma^2\left(\gamma^2\varsigma_I^2+\sigma_I^2\right)}\right.$$

$$\left. - 4\gamma\varsigma_I^2\frac{\left(\frac{(\varsigma_I^2+\mu_I^2)^2\gamma^3\varsigma_I^2}{\sigma_I^2}+\mu^2\gamma\right)}{(\gamma^2\varsigma_I^2+\sigma_I^2)^2} + \frac{(\varsigma_I^2\gamma^2-\sigma_I^2)\left[(\mu_I\gamma^2\varsigma_I^2+\mu_i\sigma_I^2)^2+\gamma^4\varsigma_I^6\right]}{\sigma_I^2\left(\gamma^2\varsigma_I^2+\sigma_I^2\right)^3}\right]$$

So the Fisher information is finally

$$I = -\begin{pmatrix} L & C & D \\ C & C & E \\ D & E & F \end{pmatrix},$$

where L, C, D, E and F are defined above.

□