The Pennsylvania State University
The Graduate School

PROBLEMS OF DISCRETE INFERENCE WITH NOISY OR INCOMPLETE DATA

A Dissertation in
Statistics
by
Jonathan Hehir

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2023

The dissertation of Jonathan Hehir was reviewed and approved by the following:

Aleksandra Slavković
Professor of Statistics & Public Health Sciences
Dissertation Co-Advisor
Chair of Committee

Xiaoyue Niu
Associate Professor of Statistics
Dissertation Co-Advisor

David Hunter
Professor of Statistics

Daniel Kifer
Professor of Computer Science & Engineering

Bing Li
Professor of Statistics
Graduate Program Chair

Abstract

This dissertation highlights a selection of modern problems involving statistical inference on discrete data. At the heart of each of these problems lies a computational challenge not adequately met by the classical or exact methods, motivating the use of fast, approximate methods. These fast approximations—data sketching for distinct counting and spectral clustering for network community detection—have been widely known and used for decades, yet a number of important gaps remain. In particular, we provide novel adaptations of these methods to three areas: distinct counting and community detection with provable guarantees of privacy, and community detection in networks whose community structure is determined by a mixture of observed and latent factors. The contributions of this work are both theoretical and practical. The proposed solutions to each of these problems is accompanied by rigorous proof of statistical consistency as well as empirical evaluation on both real and simulated data.

Table of Contents

List of Figures

List of Tables

Acknowledgments

If you're reading this—which it seems you are—I owe you a sincere *thank you.* Completing this work was made possible by years of support from family, friends, classmates, faculty, colleagues, some strangers, and of course, my wife. This mostly implicit list of people includes not only my Ph.D. advisors, committee, and collaborators, but also those who were supportive and accommodating in my career, those who provided important guidance and encouragement during undergraduate and community college studies, those who helped me grow at an earlier age, and, crucially, those who have shared their time, energy, joy, sorrow, food, drink, and company with me personally.

Computations for this research were performed on the Pennsylvania State University's Institute for Computational and Data Sciences' Roar supercomputer. This work was supported in part by NIAID/NIH R01-AI136664 and NSF Award No. SES-1853209 to The Pennsylvania State University. A portion of this dissertation is based on research performed while employed by Meta Platforms, Inc. The views, findings, and conclusions stated in this dissertation do not necessarily reflect those of any funding agency or employer.

In memory of my father,
John P. Hehir

Introduction

The history of mathematics is rich with deceptively simple problems in discrete settings. This dissertation highlights a selection of modern problems involving statistical inference on discrete data, ranging from counting to clustering. Simple at first glance, these basic tasks present fundamental computational challenges when approached at scale. Moreover, we pose added challenges by working with either noisy or incomplete data, rather than the complete and accurate data.

Each of the three problems we consider follows a similar pattern. In all cases, the goal is to produce consistent estimators that scale to large data. For this, we rely on approximate methods and forms of dimension reduction, transforming large data into compact representations that retain a high degree of utility in the statistical problem at hand. The proposed solutions to each problem are theoretically proven and empirically demonstrated to produce consistent results.

Counting and Data Sketching

We begin in Chapter 1 with a counting problem. While it may seem that counting is among the easiest of computational tasks—and indeed it can be—the task is markedly more difficult when asked to count the number of *unique* items in an enormous stream of data. Intuitively, to count unique items requires remembering the items that one has already seen. As a result, unique counting over an enormous dataset requires enormous amounts of memory, rendering the task computationally expensive or even infeasible beyond a certain scale. The problem of efficiently approximating unique counts (or *cardinality estimation*) is widely known as the *count-distinct problem* and draws parallels to other problems of statistical population size estimation.

Numerous solutions to the count-distinct problem have been posed over the last several decades that provide approximate unique counts at a fraction of the cost of determining the exact count (e.g., Flajolet and Martin, 1985; Bar-Yossef et al., 2002; Flajolet, Fusy, et al., 2007). These solutions involve the construction of small summaries or *data sketches* that represent the original set of items through a form of sampling typically based on universal hash functions. Estimation of the count of unique items—the population size—proceeds as a statistical parameter estimation task

based on the summarized values.

Thanks to the efficiency and rich aggregation possibilities available through data sketches, modern large-scale data processing frequently involves the storage or transmission of these sketches for later use, raising important privacy concerns about the information that can be gleaned from these compact representations of what may be sensitive original data. In particular, we address the following question: *How can one construct sketches that allow for aggregation and cardinality estimation but prevent inference about individual set membership?* In contrast to prior work in the literature (namely Pagh and Stausholm, 2021), our proposed solutions yield vast improvements in estimation accuracy.

As an example, suppose that a handful of healthcare providers wish to count the number of unique individuals who have received a certain form of care across all the providers. The most naive solution is for each provider to share a complete list of the patients who have received care within their respective systems. This is neither efficient nor privacy-preserving. One step up from here, each provider could provide a compact sketch (e.g., the HyperLogLog sketch of Flajolet, Fusy, et al., 2007) representing their patients, with the collected sketches aggregated to obtain an overall count. This is more efficient and seemingly more privacy-preserving, but as we show in Chapter 1, it is often possible to identify with complete certainty specific individuals who are absent in a given sketch. The solution we propose instead allows each provider to produce *private* sketches that may still be aggregated but come with provable privacy guarantees.

Differential Privacy

The question posed above is an example of *privacy-preserving* inference. The work of Chapters 1 and 2 rely on the intentional addition of noise to disguise individual records in the data while enabling higher-level statistical inference. Specifically, these problems impose and enforce the constraints of *differential privacy* (Dwork et al., 2006; Dwork, 2006), a strong and rigorous mathematical definition of privacy that has entered widespread use. Loosely speaking—and depending on the specific formulation—differential privacy requires that an algorithm applied to any data be approximately invariant to the addition, removal, or substitution of one record with another. This hinders the ability to draw conclusions about individual data records based on the output of the differentially private procedure. Strictly speaking, only a trivial algorithm can be invariant to all such changes, and so the algorithms used in differential privacy are randomized algorithms, and "approximate invariance" is measured in probabilistic terms.

Embracing differential privacy leads us to substitute noisy data (or summaries thereof) for the truth in Chapters 1 and 2. Naturally, working with noisy data leads to challenges in obtaining

estimates that are consistent and have acceptable error. In both of these chapters, we develop fast estimators that explicitly account for the added noise, and we will see that ensuring a higher degree of privacy entails obtaining noisier estimates.

Community Detection and Spectral Clustering

The problem we consider in Chapters 2 and 3 is *community detection*, i.e., clustering in the network setting. From the most primitive clustered network models, we quickly run into computational scaling challenges. In particular, consider the simple stochastic block model (SBM) of Holland et al. (1983). In the SBM, n nodes are partitioned into a set of k communities, and edges are formed independently between nodes with probability depending on the communities to which a pair of nodes belongs. A canonical statistical task is recovering how the network was partitioned, given only the observed edges of the network, i.e., clustering the nodes of the network into the k communities based on the observed edges. This can be approached through classical statistical techniques such as maximum likelihood estimation and expectation–maximization—but only when the network size n is very small, as the computational cost of these methods is exponential in n (Snijders and Nowicki, 1997).

A popular polynomial-time alternative is found in the non-parametric method of *spectral clustering*. Applied to the community detection problem, spectral clustering represents nodes as vectors in a low-dimensional space obtained from a low-rank approximation of a network's adjacency, Laplacian, or similarity matrix. Clustering these vectors has been shown to produce consistent results with appealing theoretical properties across a range of statistical settings (e.g., Von Luxburg, 2007; Lei and Rinaldo, 2015; Abbe, Fan, K. Wang, and Zhong, 2020; Rubin-Delanchy et al., 2022), while also allowing for processing of much larger networks than using classical techniques.

We consider two questions related to spectral clustering. First, a question of privacy preservation: *Can we perform spectral clustering without true knowledge of the relationships (edges) within a network?* To our knowledge, the results in Chapter 2 are the first to address this problem with theoretical guarantees of both privacy and consistency. The methods employed to ensure differential privacy in this problem bear a resemblance to those used in the distinct-counting problem, with both adding noise to binary data through random perturbations. In this case, the random perturbations are applied to the edges of the network, resulting in a *private, synthetic* network that resembles the truth but does not perfectly and faithfully reproduce it. As we show in a number of real-world examples in Section 2.6, a modified spectral clustering algorithm applied to this privatized network can identify clusters of the network in much the same way as

non-private spectral clustering on the true network, though generally with higher error for larger perturbations.

Where the standard community detection problem aims to recover fully latent node labels, the final chapter addresses a network whose structure is determined by a mixture of latent and observed node-level variables. In this setting, network edges are formed through a generalized linear model (GLM) over the observed and unobserved variables, all of which are discrete. In such a network, we ask: *Is it possible to recover the latent factors that contribute to edge formation?* Posing this as a community detection problem reveals an answer. Such networks exhibit communities aligned with the various configurations of the node covariates, of which we have partial knowledge. We leverage spectral clustering as a fast method to detect this community structure, then unravel the communities to separate the latent and observed variables. This separation is made possible by the regularity of the GLM structure. This sort of problem has been raised in a number of prior works (e.g., Hoff, 2007; Handcock et al., 2007; Mele et al., 2022), but ours is the first to rigorously justify the use of a spectral clustering procedure. In Section 3.6, we provide a concrete example of our method applied to a network of Facebook friendships among Harvard students. Where ordinary spectral clustering identifies clusters that align nearly perfectly with students' observed class years, our method identifies a truly latent 2×2 block structure after accounting for the patterns imposed by the observed covariates (see Figure 3.2).

Structure and Provenance

The chapters to follow are adapted from three joint publications: Chapter 1 is adapted from Hehir, Ting, et al. (2023) (under review at the time of this writing), Chapter 2 from Hehir, Slavkovic, et al. (2022) (in *Journal of Privacy and Confidentiality*), and Chapter 3 from Hehir, Niu, et al. (2022) (under review). Each of these chapters begins with a brief abstract as well as a statement on publication and authorship. Reproducible, open-source implementations of the proposed methods are also listed at the beginning of each chapter (when available), in the form of current GitHub repositories and permanent Zenodo archives (Hehir, 2022b; Hehir, 2023). Notation used in a chapter is defined at the end of each chapter's introduction. All simulations, examples, and major theoretical results are provided within the main chapter text. Some proofs—particularly those that are short or provide important mathematical intuition—are also included in the main text, but longer proofs, including some helper lemmas, and additional mathematical background are provided in Appendices A–C, corresponding to Chapters 1–3, respectively.

Chapter 1

Mergeable Data Sketches for Private Distinct Counting

Abstract. Data sketching is a critical tool for distinct counting, enabling multisets to be represented by compact summaries that admit fast cardinality estimates. Because sketches may be merged to summarize multiset unions, they are a basic building block in data warehouses. Although many practical sketches for cardinality estimation exist, none satisfy both differential privacy (DP) and mergeability. We propose the first practical cardinality sketches that are simultaneously mergeable, DP, and have low empirical error. We introduce a novel randomized algorithm for performing logical operations on noisy bits, a tight privacy analysis, and provably optimal estimation. Our sketches dramatically outperform existing theoretical solutions in simulations and on real-world data.

Publication. This chapter and the accompanying Appendix A are adapted from a joint publication with Daniel Ting and Graham Cormode, under review at the time of this dissertation.[1] As first author, I developed the majority of the main results and proofs, wrote the software implementation, and conducted the simulations. The three authors contributed equally to the drafting of the publication. This research was primarily conducted while employed at Meta.

Code. At the time of this writing, an open-source implementation of the proposed method is not available.

1.1 Introduction

Many applications that model large volumes of data are based on tracking cardinalities of events or observations. Consequently, these applications make extensive use of data sketches that support fast, approximate cardinality estimation (Cormode and Yi, 2020). For instance, approximate distinct counting is supported via variants of the HyperLogLog (HLL) sketch (Flajolet, Fusy, et al., 2007; Heule et al., 2013) in popular data management systems including Amazon Redshift, ClickHouse,

[1] Jonathan Hehir, Daniel Ting, and Graham Cormode (2023). "Sketch-Flip-Merge: Mergeable Sketches for Private Distinct Counting". In: *arXiv preprint arXiv:2302.02056*.

Google BigQuery, Splunk, Presto, Redis, and more. At the expense of a small estimation error, these approximate methods drastically reduce the computational cost of distinct counting to run in linear time, using only bounded memory. An additional key feature of distinct-count sketches is the ability to merge two or more sketches to obtain cardinality estimates over their union. This enables not only distributed computation, but also many rich aggregation possibilities from previously computed sketches. As a result, modern data pipelines rely extensively on the performance and functionality of such cardinality sketches.

Increasingly, privacy concerns constrain the operation of data processing. Organizations demonstrating commitments to preserving users' privacy require that data collected from individuals be subject to appropriate mitigations before being passed to downstream processing. Specifically, protections such as differential privacy are used to protect sensitive data while still giving accurate query response.

Although sketching techniques may appear to offer protection by reducing data, it is well-known that sketching alone does not automatically provide a privacy guarantee (Desfontaines et al., 2019). The summaries—or even the estimates calculated from them—can leak considerable information about whether the specific items belong to the underlying set. Recently, it has been shown that the contents of sketches do meet a privacy standard *if* the associated hash functions are not known to the observer (S. G. Choi et al., 2020; A. Smith et al., 2020; Dickens et al., 2022). However, it is not plausible to assume secret hash functions when the computation is shared among multiple entities in a large scale system. In particular, all participants must know the hash when working with sketches that will be merged, and using the same hash in multiple sketches generates correlated randomness that breaks the privacy guarantees. This creates an important gap to make these high-throughput systems private. Previous attempts to construct privacy-preserving sketches (Pagh and Stausholm, 2021) do not offer practical mergeable sketches as the errors are too large (Section 1.6).

In this work, we present the Sketch-Flip-Merge (SFM) summaries, a practical, mergeable, and provably private approach to distinct-count sketching. In particular, we produce summaries that satisfy the strong definition of ε–differential privacy (DP) (Dwork et al., 2006; Dwork, 2008) even when the hash function is known publicly. By attaching the privacy guarantee to the summary itself—not just the cardinality estimate—we may safely release summaries corresponding to sensitive multisets, enabling safe cardinality estimation over any union of such sets using the privacy-preserving summaries in lieu of the original sensitive data.

The key to our approach is to adapt the sketch of Flajolet and Martin (1985), which is often referred to as either *FM85* or *probabilistic counting with stochastic averaging (PCSA)*. Although

subsequent sketches such as HLL (Durand and Flajolet, 2003; Flajolet, Fusy, et al., 2007; Heule et al., 2013) further optimized the space usage, squeezing the space makes them less amenable to privacy protection. In contrast to PCSA where the simple, partitioned binary structure limits the sensitivity to a bit flip, these sketches store extremal hash values where small changes to the input can cause big changes in the summary, requiring more noise and yielding less accurate results. Furthermore, our methods generalize to any bitmap based sketch.

Related Work. Privacy-preserving cardinality sketches have been the subject of several earlier works. While recent efforts provide DP guarantees for HLL-like sketches (A. Smith et al., 2020; Dickens et al., 2022), they rely on random, secret hash functions that preclude the ability to merge sketches. Using a fixed, public hash, S. G. Choi et al. (2020) obtain a DP cardinality estimate from a LogLog sketch by adding noise to the cardinality estimator, but the sketch itself remains sensitive and unsafe for release or sharing. Stanojevic et al. (2017) design a DP algorithm for obtaining cardinality estimation on the union of two multisets using perturbed Bloom filters, but their method does not generalize and scale to the union of more than two multisets.

One line of work extends PCSA with randomized response and subsampling of items to achieve privacy (Tschorsch and Scheuermann, 2013; Nuñez von Voigt and Tschorsch, 2019). However, Tschorsch and Scheuermann (2013) fails to achieve a DP guarantee, and Nuñez von Voigt and Tschorsch (2019) does not address merging sketches. Kreuter et al. (2020) design two sketches, including one based on PCSA. While their DP sketches cannot be merged to form a single sketch, multiple sketches may be used to estimate the union's cardinality if all sketches use the same privacy parameters. The PCSA-based sketch of Pagh and Stausholm (2021, Section 6) achieves DP and supports merging. However, as we observe in our experiments, the sketches and cardinality estimates are very noisy. Finally, Desfontaines et al. (2019) give an impossibility result where both privacy and high accuracy are impossible, but only when many sketches are merged, which is consistent with our results.

Contributions. We propose two practical methods for constructing *mergeable* DP cardinality sketches and obtaining cardinality estimates. The first uses a deterministic bit-merging operation used by Pagh and Stausholm (2021). We prove this merge requires a suboptimal form of randomized response, even after exponential improvement to the prior privacy analysis (Corollary 1.7). Our main methodological contribution is a novel randomized merge allowing for up to a further 75% variance reduction over the optimized deterministic merge. We generalize our randomized merge to perform to arbitrary bitwise operations on binary data that may be of independent interest. We also develop a composite likelihood-based estimator for cardinality and prove this estimator is asymptotically optimal for both private and non-private sketches based on PCSA.

Outline. We give a brief overview of PCSA sketching in Section 1.2, then define privacy and recap randomized response in Section 1.3. Merging sketches is enabled through the careful design of randomized response mechanisms and merge operations over collections of randomized bits in Section 1.4. In Section 1.5, we propose a fast cardinality estimator for the private PCSA sketch and analyze its properties. We compare these methods with private and non-private alternatives in Section 1.6 and state conclusions in Section 1.7.

Notation. We write $[m] = \{1, \ldots, m\}$. $\otimes$ denotes the Kronecker product. Logical operations are denoted $\vee$ (or), $\wedge$ (and), $\veebar$ (xor), and $\neg$ (not). We use the natural logarithm $\log = \log_e$. Equality in distribution is denoted $\overset{D}{=}$. The cardinality of a set D is denoted $|D|$.

1.2 Background and Problem Setup

Let $D \in \mathcal{X}^N$ denote a multiset of N items from some universe $\mathcal{X}$ of objects. The *count-distinct problem* is the task of estimating the number of unique elements in D. That is, if $\texttt{set}(D)$ denotes the support set of items in D, the count-distinct problem aims to approximate $n = |\texttt{set}(D)|$ with a data sketch in bounded memory in a single pass over the data. We consider the *private* count-distinct problem for *mergeable* sketches where the information in a sketch satisfies differential privacy (DP) and sketches can be merged to obtain a sketch of the union of underlying datasets.

We focus on solutions to the count-distinct problem in which sketches form a binary vector, subject to merge operations performed through element-wise logical operations (e.g., *or*). The class of sketches to which our methods apply include PCSA, linear counting (Whang et al., 1990), Bloom filters (Broder and Mitzenmacher, 2004), and Liquid Legions (Kreuter et al., 2020). These are particularly amenable to privacy enhancement through the application of randomized response (Warner, 1965) but require careful design of merge operations for randomized bits. Although this excludes other commonly used sketches such as HyperLogLog and the k-minimum value sketch (Bar-Yossef et al., 2002; Giroire, 2009), the richer set of values stored in these sketches make them less suitable for privatization due to their high sensitivity and, hence, higher noise required for privacy. In the remainder of this paper, we focus on the PCSA sketch of Flajolet and Martin (1985), noting that the results for constructing and merging private sketches in Sections 1.3 and 1.4 apply to related sketches through direct application or simple extensions.

The classical PCSA sketch takes the form of a matrix $S = \mathcal{S}(D) \in \{0, 1\}^{B \times P}$ with B buckets and precision parameter P. Given two independent, universal hash functions, $h_1(x) \sim \text{Uniform}([B]), h_2(x) \sim \text{Geometric}(1/2)$, let $\texttt{bucket}(x) = h_1(x), \texttt{value}(x) = \min\{P, h_2(x)\}$. Then each bit S_{ij} is equal to 1 iff there exists $x \in D$ such that $\texttt{bucket}(x) = i, \texttt{value}(x) = j$. Some

desirable properties of S are immediate. First, S relies only on the set of hashed values $\{h_1(x)\}_{x\in D}$ and $\{h_2(x)\}_{x\in D}$. Hence, it is invariant both to repetitions in D and to the order in which the elements of D are processed. Additionally, two sketches $\mathcal{S}(D_1)$ and $\mathcal{S}(D_2)$ may be merged via a simple bitwise-*or*, $\mathcal{S}(D_1) \vee \mathcal{S}(D_2) = \mathcal{S}(D_1 \cup D_2)$, as each entry S_{ij} in the merged sketch is equal to 1 iff there is an item x in at least one of D_1, D_2 for which $\texttt{bucket}(x) = i, \texttt{value}(x) = j$.

Importantly, when an adversary knows h_1 and h_2, the sketch $\mathcal{S}(D)$ reveals information about elements in D. For example, any $x \in \mathcal{X}$ for which $S_{\texttt{bucket}(x),\texttt{value}(x)} = 0$ cannot belong to D. In what follows, we extend the PCSA sketch to minimize this sort of privacy leakage.

1.3 Private Sketches

Differential privacy (DP) (Dwork et al., 2006; Dwork, 2008) offers a strong and quantifiable notion of privacy. DP mandates that algorithms (*privacy mechanisms*) acting on a dataset D must be randomized—typically through the addition of some carefully tuned noise—so that the distribution of a privacy mechanism's output cannot be significantly influenced by a single input record. As a result, the ability to reverse-engineer information about a single record is limited, and any analysis performed using only the output of the algorithm also satisfies DP. The strength of the DP guarantee is quantified by the parameter $\varepsilon > 0$, often called the privacy budget, with smaller ε offering stronger privacy.

Definition 1.1 (Dwork et al. (2006))**.** *A randomized algorithm $\mathcal{M}$ is said to satisfy ε–**differential privacy (DP)** if for any two neighboring databases D, D' and any set of outputs $E \subseteq \text{Range}(\mathcal{M})$, we have:*

$$\mathbf{P}(\mathcal{M}(D) \in E) \leq e^{\varepsilon}\, \mathbf{P}(\mathcal{M}(D') \in E).$$

In the count-distinct problem, we say two multisets D, D' neighbor if D' can be obtained by adding or removing one unique item to D.

Suppose we have two neighboring multisets D, D', and we consider their PCSA sketches $\mathcal{S}(D), \mathcal{S}(D')$. It follows from the definition of PCSA that these sketches must agree on all but at most one bit. To create DP sketches from $\mathcal{S}(D)$, then, we consider general DP mechanisms applied to vectors of $\{0, 1\}$ bits, where two vectors $x, x' \in \{0, 1\}^d$ neighbor if they differ on at most one bit (i.e., have *sensitivity* 1).

When restricting our attention to mechanisms whose input and output are both a single bit, every DP mechanism can be viewed as an instance of randomized response (RR), dating back to Warner (1965). We describe a generalized form of RR as follows. Let $\mathcal{F}_{p,q}$ denote a general

bit-flipping algorithm, parameterized by two probabilities p and q:

$$\mathcal{F}_{p,q}(x) \sim \begin{cases} \text{Bernoulli}(p), & x = 1 \\ \text{Bernoulli}(q), & x = 0 \end{cases}.$$

Theorem 1.2. *Assume $q \leq 1/2 \leq p$. Applied to vectors with sensitivity 1, the algorithm $\mathcal{M}_{p,q} : \{0,1\}^d \to \{0,1\}^d$ that independently applies $\mathcal{F}_{p,q}$ to each element of its input is ε-DP if and only if:*

$$p, q \in (0,1) \quad \textit{and} \quad \max\left\{\frac{p}{q}, \frac{1-q}{1-p}\right\} \leq e^{\varepsilon}. \tag{1.1}$$

Proof. The proof follows standard techniques and is included in Appendix A. □

Theorem 1.2 provides an entire family of privacy mechanisms $\{\mathcal{M}_{p,q} : p, q \text{ satisfy } (1.1)\}$ that could be applied to a PCSA sketch to make its output ε-DP. Our contribution is then to address several important questions: How can we merge two sketches if their bits have been perturbed via $\mathcal{F}_{p,q}$? How can we estimate cardinality from a sketch with perturbed bits? And how should we choose p and q?

1.4 Merging Perturbed Bit Vectors

We can use a randomized response (RR) mechanism $\mathcal{M}_{p,q}$ to convert PCSA sketches to a private equivalent, but the resulting sketch is no longer mergeable. For ordinary PCSA and multisets D_1, D_2, the bitwise-*or* $\mathcal{S}(D_1) \vee \mathcal{S}(D_2) = \mathcal{S}(D_1 \cup D_2)$ defines a merge operation on sketches that yields the same sketch that would be obtained by first taking the union. However, the same operation on noisy sketches does not satisfy this desirable property. Here, we develop merge operations on noisy sketches and identify under what conditions they exist. In particular, we show (Theorem 1.3) that if a merge operation is deterministic, then *xor* ($\veebar$) is the only possible merge on noisy sketches, and it only works for certain choices of the mechanism $\mathcal{M}_{p,q}$. We show these choices imply that, at best, the variance of such a noisy sketch's cardinality estimator is four times worse than that for regular PCSA on the same sized sketch, even if the privacy budget is near-infinite. Our main contribution is to provide a novel *randomized* merge operation that adds less noise to the sketch. Furthermore, we generalize this operation to perform arbitrary boolean operations on noisy bit vectors.

1.4.1 Deterministic Merging

Applying the standard randomized response mechanism to a PCSA sketch breaks mergeability. PCSA merges sketches using bitwise-*or*, and in the presence of RR noise, the *or* operation biases bits towards 1. Pagh and Stausholm (2021) address this by replacing bitwise-*or* ($\vee$) with bitwise-*xor* ($\veebar$) operations whenever the sketch is updated or merged. However, the *xor* operation alone destroys cardinality information. In particular, the *xor* of a PCSA sketch with itself is the empty sketch. Instead, they subsample items (including duplicates) independently with probability $1/2$. This effectively encodes bits that were 1 in PCSA as Bernoulli$(1/2)$ values, while 0 bits remain 0, and makes the *distribution* of the sketch invariant to duplicates, even though the sketch itself is not. Still, this subsampling operation introduces a lot of noise. Figure 1.4 shows that even for large ε the resulting cardinality estimates have 4 times the variance.

We show that this penalty on the accuracy is inherent for any *deterministic* merge. By considering the broader family of RR mechanisms, Theorem 1.3 shows *xor* is, in fact, the only possible way to merge deterministically, so that randomized merges are the only way to improve merging. Our analysis also improves the Pagh and Stausholm (2021) sketch by significantly reducing the noise required to achieve an ε-DP privacy guarantee and demonstrates how to merge sketches with different privacy budgets.

Theorem 1.3. *Let $f_1 = \mathcal{F}_{p_1,q_1}, f_2 = \mathcal{F}_{p_2,q_2}, f_3 = \mathcal{F}_{p_3,q_3}$, and let $\circ : \{0,1\}^2 \to \{0,1\}$ denote a deterministic and symmetric operation. The following conditions may only be satisfied simultaneously if $\circ = \veebar$ and $p_1 = p_2 = 1/2$:*

1. *f_1, f_2 are ε_1-DP and ε_2-DP for $\varepsilon_1, \varepsilon_2 < \infty$.*
2. *$f_1(x) \circ f_2(y) \overset{D}{=} f_3(x \vee y)$.*
3. *$f_i(0) \overset{D}{\neq} f_i(1)$ for $i = 1, 2, 3$.*

Proof Sketch. The complete proof is included in Appendix A, but we give a brief overview here. First, observe that the possible operations $\circ$ are limited: up to negation, the only possible operations are *and* ($\wedge$), *or* ($\vee$), *xor* ($\veebar$), and a trivial operation that maps to a constant. Assuming the existence of a merge operation other than $\veebar$ leads to a contradiction. Finally, using $\circ = \veebar$ and under the assumption that condition (2) holds, we require:

$$f_1(0) \veebar f_2(1) \overset{D}{=} f_3(1) \overset{D}{=} f_1(1) \veebar f_2(1).$$

Taking expectations of the left– and right-hand side (see Fact A.1), this implies:

$$q_1(1-p_2) + p_2(1-q_1) = p_1(1-p_2) + p_2(1-p_1).$$

Rearranging terms yields:

$$p_2(p_1 - q_1) = (1-p_2)(p_1 - q_1).$$

Since we assumed $p_1 \neq q_1$, we must have $p_2 = \frac{1}{2}$. A similar argument shows $p_1 = \frac{1}{2}$. □

Using our general family of RR mechanisms, we define a mechanism that adds noise to a PCSA sketch to provide privacy (Lemma 1.5) while preserving mergeability (Theorem 1.6). Corollary 1.7 shows our privacy analysis is much tighter than that of Pagh and Stausholm (2021).

Definition 1.4. *For $\varepsilon > 0$, let $\mathcal{M}_\varepsilon^{\text{xor}} : \{0,1\}^d \to \{0,1\}^d$ denote the mechanism that independently applies an asymmetric randomized response $\mathcal{F}_{p,q}$ to each element of its input with $p = 1/2, q = 1/(2e^\varepsilon)$.*

Lemma 1.5. *$\mathcal{M}_\varepsilon^{\text{xor}}$ is ε–differentially private.*

Proof. We need to show that $p/q \leq e^\varepsilon$ and $(1-q)/(1-p) \leq e^\varepsilon$ for $p = 1/2$ and $q = 1/(2e^\varepsilon)$. Clearly, $p/q = e^\varepsilon$, satisfying the first component. Next, consider the expression

$$\begin{aligned} e^\varepsilon - \frac{1-q}{1-p} &= e^\varepsilon - \frac{1 - \frac{1}{2}e^{-\varepsilon}}{\frac{1}{2}} \\ &= e^\varepsilon - (2 - e^{-\varepsilon}) \\ &= e^\varepsilon + e^{-\varepsilon} - 2 \\ &= (e^{\varepsilon/2} - e^{-\varepsilon/2})^2 \\ &\geq 0. \end{aligned}$$

Therefore, $(1-q)/(1-p) \leq e^\varepsilon$ as required. □

Theorem 1.6. *$\mathcal{M}_{\varepsilon_1}^{\text{xor}}(x) \veebar \mathcal{M}_{\varepsilon_2}^{\text{xor}}(y) \overset{D}{=} \mathcal{M}_{\varepsilon^*}^{\text{xor}}(x \vee y)$, where $\varepsilon^* = -\log(e^{-\varepsilon_1} + e^{-\varepsilon_2} - e^{-(\varepsilon_1+\varepsilon_2)})$.*

Proof. Since the entries of $\mathcal{M}^{\text{xor}}(\cdot)$ are independent, it suffices to show this holds for $\mathcal{M}^{\text{xor}}$ applied to arbitrary $x_i, y_i \in \{0,1\}$. Observe that if $x_i = 1$ or $y_i = 1$, then $x_i \vee y_i = 1$, and so $\mathcal{M}_{\varepsilon^*}^{\text{xor}}(x_i \vee y_i) \sim \text{Bernoulli}(\frac{1}{2})$. On the other hand, since we know $x_i = 1$ or $y_i = 1$, then $\mathcal{M}_{\varepsilon_1}^{\text{xor}}(x_i) \sim \text{Bernoulli}(\frac{1}{2})$ or $\mathcal{M}_{\varepsilon_2}^{\text{xor}}(y_i) \sim \text{Bernoulli}(\frac{1}{2})$, and so by Fact A.1, $\mathcal{M}_{\varepsilon_1}^{\text{xor}}(x_i) \veebar \mathcal{M}_{\varepsilon_2}^{\text{xor}}(y_i) \sim \text{Bernoulli}(\frac{1}{2})$.

Thus all that remains to show is that $\mathcal{M}_{\varepsilon^*}^{\text{xor}}(x \vee y) \overset{D}{=} \mathcal{M}_{\varepsilon_1}^{\text{xor}}(x_i) \veebar \mathcal{M}_{\varepsilon_2}^{\text{xor}}(y_i)$ when $x_i = y_i = 0$. In this case, $\mathcal{M}_{\varepsilon_1}^{\text{xor}}(x_i) \sim \text{Bernoulli}(\frac{1}{2}e^{-\varepsilon_1})$, and $\mathcal{M}_{\varepsilon_2}^{\text{xor}}(y_i) \sim \text{Bernoulli}(\frac{1}{2}e^{-\varepsilon_2})$. By Fact A.1, we have

that:

$$\mathcal{M}^{\text{xor}}_{\varepsilon_1}(x_i) \veebar \mathcal{M}^{\text{xor}}_{\varepsilon_2}(y_i) \sim \text{Bernoulli}(q^*),$$

where

$$\begin{aligned} q^* &= \frac{1}{2}e^{-\varepsilon_1}\left(1 - \frac{1}{2}e^{-\varepsilon_2}\right) + \frac{1}{2}e^{-\varepsilon_2}\left(1 - \frac{1}{2}e^{-\varepsilon_1}\right) \\ &= \frac{1}{2}\left(e^{-\varepsilon_1} + e^{-\varepsilon_2} - e^{-(\varepsilon_1+\varepsilon_2)}\right) \\ &= \frac{1}{2}e^{-\varepsilon^*}. \end{aligned}$$

Finally, since $x_i \vee y_i = 0$, we have that $\mathcal{M}^{\text{xor}}_{\varepsilon^*}(x_i \vee y_i) \sim \text{Bernoulli}(\frac{1}{2}e^{-\varepsilon^*})$. □

Corollary 1.7. *Let $\mathcal{M}^{\text{PS}}_{\varepsilon}$ denote the ε-DP privacy mechanism of Pagh and Stausholm (2021, Section 6). Then $\mathcal{M}^{\text{xor}}_{\varepsilon} = \mathcal{M}^{\text{PS}}_{2(\exp(\varepsilon)-1)}$.*

This tighter privacy analysis[2] dramatically reduces noise added to achieve the privacy guarantee, effectively increasing the privacy budget by at least a factor of 2. Pragmatically, Figure 1.4 shows that error increases exponentially as $\varepsilon \to 0$.

1.4.2 Randomized Merging

Theorem 1.3 showed that a deterministic merge is only possible if the 1-bits in a PCSA sketch are randomized to Bernoulli(1/2) values. Thus, even if the privacy budget is nearly infinite, the mergeable DP sketch must add significant noise to the base PCSA sketch. We show that by adding randomness to the merge procedure, we can remove this requirement and achieve lower overall noise while using the standard randomized response mechanism (Definition 1.8).

Definition 1.8. *For $\varepsilon > 0$, we denote by $\mathcal{M}^{\text{sym}}_{\varepsilon} : \{0,1\}^d \to \{0,1\}^d$ the mechanism that independently applies the standard RR mechanism $\mathcal{F}_{p,1-p}$ to each element of its input with $p = e^{\varepsilon}/(e^{\varepsilon}+1)$.*

Lemma 1.9. *$\mathcal{M}^{\text{sym}}_{\varepsilon}$ is ε–differentially private.*

Proof. Since $p = e^{\varepsilon}/(e^{\varepsilon}+1)$ and $q = 1 - p = 1/(e^{\varepsilon}+1)$, we have $p/q = (1-q)/(1-p) = e^{\varepsilon}$, and so the result follows immediately from Theorem 1.2. □

For a merge that works under this standard RR mechanism, we seek a randomized algorithm $g_{\varepsilon_1,\varepsilon_2} : \{0,1\}^2 \to \{0,1\}$ to merge two noisy bits. That is, it must satisfy

$$g_{\varepsilon_1,\varepsilon_2}(\mathcal{M}^{\text{sym}}_{\varepsilon_1}(x), \mathcal{M}^{\text{sym}}_{\varepsilon_2}(y)) \overset{D}{=} \mathcal{M}^{\text{sym}}_{\varepsilon^*}(x \vee y).$$

[2]It is proven in the appendix of Pagh and Stausholm (2021) that $q = 1/(e^{\varepsilon}+1)$ satisfies ε-DP, although the recommendation and main results in the paper rely on the choice of $q = 1/(2+\varepsilon)$. Our recommendation of $q = 1/(2e^{\varepsilon})$ is optimal under DP constraints.

Since $g_{\varepsilon_1,\varepsilon_2}$ is a random mapping from pairs of bits to single bits, we can represent it as a Markov transition matrix. In this manner, we may represent this entire equation as a matrix equality, with ε^* a free parameter. Solving for the largest ε^* that generates a valid solution yields the following optimal randomized merge operation for $\mathcal{M}^{\text{sym}}$.

Theorem 1.10. *Assume $\varepsilon_1, \varepsilon_2 > 0$. Let $q(\varepsilon) = (e^\varepsilon + 1)^{-1}$,*

$$\varepsilon^* = -\log(e^{-\varepsilon_1} + e^{-\varepsilon_2} - e^{-(\varepsilon_1+\varepsilon_2)}),\ q^* = q(\varepsilon^*),$$

$$K_i = \begin{bmatrix} 1-q(\varepsilon_i) & q(\varepsilon_i) \\ q(\varepsilon_i) & 1-q(\varepsilon_i) \end{bmatrix} \text{ for } i \in \{1,2\}, \text{ and}$$

$$v^* = (q^*, 1-q^*, 1-q^*, 1-q^*)^T,$$

Letting $\otimes$ denote the Kronecker product, define:

$$(t_{00}, t_{01}, t_{10}, t_{11})^T = (K_1^{-1} \otimes K_2^{-1})\, v^*, \text{ and}$$

$$g_{\varepsilon_1,\varepsilon_2}(a,b) \sim \text{Bernoulli}(t_{ab}), \quad a,b \in \{0,1\}.$$

Then $g_{\varepsilon_1,\varepsilon_2}(\mathcal{M}^{\text{sym}}_{\varepsilon_1}(x), \mathcal{M}^{\text{sym}}_{\varepsilon_2}(y)) \overset{D}{=} \mathcal{M}^{\text{sym}}_{\varepsilon^*}(x \vee y)$, where g is taken bitwise and independently.

Proof. This theorem is proven in greater generality as Theorem A.3 in Appendix A. □

When the original vectors x and y are released in addition to the merged vector, the ε^* parameter of Theorems 1.6 and 1.10 is best interpreted as a measure of utility in the merged sketch, rather than a privacy budget, since by the post-processing invariance of DP (Dwork et al., 2006), no additional privacy leakage occurs from the release of the merged vector. It is for this reason we seek the maximal ε^* in merging. Noting that Theorems 1.6 and 1.10 produce identical ε^* and that $\mathcal{M}^{\text{sym}}_\varepsilon$ is less noisy than $\mathcal{M}^{\text{xor}}_\varepsilon$, $\mathcal{M}^{\text{sym}}$ remains the preferred mechanism after merging.

Remark 1.11. *By induction, the merges prescribed in Theorems 1.6 and 1.10 allow any k bit vectors of equal length $x_1, \ldots, x_k$ privatized using $\varepsilon_1, \ldots, \varepsilon_k$ to be merged, resulting in a vector equivalent to $v = (x_1 \vee \cdots \vee x_k)$ privatized with*

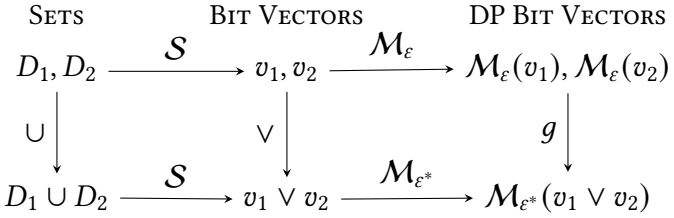
$$\varepsilon^* = -\log\left(1 - \prod_{i=1}^{k}(1 - e^{-\varepsilon_i})\right).$$

A natural question is whether there exists a randomized merge algorithm $g_{\varepsilon_1,\ldots,\varepsilon_k} : \{0,1\}^k \to \{0,1\}$ that satisfies a property like Theorem 1.10 with a larger ε^* than given by induction over the pairwise

merges. In Appendix A, we prove a more general form of Theorem 1.10 (Theorem A.3), which answers this question in the negative.

1.4.3 General Boolean Operations

We briefly switch focus from distinct counting to present a generalization to Boolean operations under randomized response that may be of theoretical interest. In distinct-count sketches, set unions correspond to bitwise-*or* operations, and the challenge posed by privacy is performing an equivalent operation over noisy bit vectors. PCSA, like other mergeable distinct counting methods, defines a homomorphism $\mathcal{S}$ from multisets to sketches. The commutative diagram below illustrates this mergeability property, since it does not matter which path one takes from D_1, D_2 to $v_1 \vee v_2$. Likewise, our merge operation g (Theorem 1.10) ensures that the privacy mechanism $\mathcal{M}$ makes the diagram commute. By preserving the structure of the union operation, inferences about the cardinality of the union can be made from merged, private sketches.



We generalize the randomized *or* ($\vee$) merge under $\mathcal{M}^{\text{sym}}$ to any logical operation, as summarized in Table 1.1. In particular, Corollary 1.12 shows a simple change in our target probabilities v^* (of Theorem 1.10) yields the appropriate randomized merge for *and* ($\wedge$), while Lemma 1.13 demonstrates a merge for *xor* ($\veebar$). In light of these results (and the original Theorem 1.10), for each choice of $\square \in \{\vee, \wedge, \veebar\}$, we have a merge operation $g_{\square,\varepsilon_1,\varepsilon_2}(x, y)$ and a privacy budget–combining function $\varepsilon^*_\square(\varepsilon_1, \varepsilon_2)$ from which the homomorphism emerges. In particular, define two semigroups A and B with operations $\cdot_A$ and $\cdot_B$:

$$
\begin{aligned}
A &= \{0,1\} \times \mathbb{R}_{\geq 0} \\
(x, \varepsilon_1) \cdot_A (y, \varepsilon_2) &= \left(x \,\square\, y,\ \varepsilon^*_\square(\varepsilon_1, \varepsilon_2)\right), \\
B &= \{\text{Bernoulli}(p) : p \in [0,1]\} \times \mathbb{R}_{\geq 0} \\
(X, \varepsilon_1) \cdot_B (Y, \varepsilon_2) &= \left(g_{\square,\varepsilon_1,\varepsilon_2}(X, Y),\ \varepsilon^*_\square(\varepsilon_1, \varepsilon_2)\right)
\end{aligned}
$$

Table 1.1. Boolean Operations on Bit Vectors under $\mathcal{M}^{\text{sym}}$

Op.	DP Op.	$\varepsilon^*(\varepsilon_1, \varepsilon_2)$
$\neg$	Lem. 1.14	—
$\vee$	Thm. 1.10	$-\log(e^{-\varepsilon_1} + e^{-\varepsilon_2} - e^{-(\varepsilon_1+\varepsilon_2)})$
$\wedge$	Cor. 1.12	$-\log(e^{-\varepsilon_1} + e^{-\varepsilon_2} - e^{-(\varepsilon_1+\varepsilon_2)})$
$\veebar$	Lem. 1.13	$\log\left(\frac{1+e^{\varepsilon_1+\varepsilon_2}}{e^{\varepsilon_1}+e^{\varepsilon_2}}\right)$

Then $\phi(x, \varepsilon) = (\mathcal{M}_\varepsilon^{\text{sym}}(x), \varepsilon)$ is a homomorphism from A to B, as

$$\begin{aligned}
\phi((x, \varepsilon_1) \cdot_A (y, \varepsilon_2)) &= \phi(x \,\square\, y,\ \varepsilon_\square^*(\varepsilon_1, \varepsilon_2)) \\
&= \left(\mathcal{M}^{\text{sym}}_{\varepsilon_\square^*(\varepsilon_1,\varepsilon_2)}(x \,\square\, y),\ \varepsilon_\square^*(\varepsilon_1, \varepsilon_2)\right) \\
&= \left(g_{\square,\varepsilon_1,\varepsilon_2}(\mathcal{M}_{\varepsilon_1}^{\text{sym}}(x), \mathcal{M}_{\varepsilon_2}^{\text{sym}}(y)),\ \varepsilon_\square^*(\varepsilon_1, \varepsilon_2)\right) \\
&= (\mathcal{M}_{\varepsilon_1}^{\text{sym}}(x), \varepsilon_1) \cdot_B (\mathcal{M}_{\varepsilon_2}^{\text{sym}}(x), \varepsilon_2) \\
&= \phi(x, \varepsilon_1) \cdot_B \phi(y, \varepsilon_2).
\end{aligned}$$

For brevity, we state the following two results—the *and* ($\wedge$) and *xor* ($\veebar$) merges under $\mathcal{M}^{\text{sym}}$—without proof, noting that the difficult work was performed earlier in Theorem 1.10. Complete proofs may be found in Appendix A.

Corollary 1.12. *Assume the setting of Theorem 1.10, but set* $v^* = (q^*, q^*, q^*, 1 - q^*)^T$. *Then* $g_{\varepsilon_1,\varepsilon_2}(\mathcal{M}_{\varepsilon_1}^{\text{sym}}(x), \mathcal{M}_{\varepsilon_2}^{\text{sym}}(y)) \overset{D}{=} \mathcal{M}_{\varepsilon^*}^{\text{sym}}(x \wedge y)$, *where* g *is taken bitwise and independently.*

Lemma 1.13. $\mathcal{M}_{\varepsilon_1}^{\text{sym}}(x) \veebar \mathcal{M}_{\varepsilon_2}^{\text{sym}}(y) \overset{D}{=} \mathcal{M}_{\varepsilon^*}^{\text{sym}}(x \veebar y)$ *for* $\varepsilon^* = \log(1 + e^{\varepsilon_1+\varepsilon_2}) - \log(e^{\varepsilon_1} + e^{\varepsilon_2})$.

As our final step in supporting general Boolean operations under $\mathcal{M}^{\text{sym}}$, we show that the unary operation *not* ($\neg$) commutes with $\mathcal{M}^{\text{sym}}$.

Lemma 1.14. *For any bit vector* x, *we have* $\neg(\mathcal{M}_{p,q}(x)) \overset{D}{=} \mathcal{M}_{p,q}(\neg x)$ *if and only if* $q = 1 - p$. *In particular,* $\neg(\mathcal{M}_\varepsilon^{\text{sym}}(x)) \overset{D}{=} \mathcal{M}_\varepsilon^{\text{sym}}(\neg x)$.

Proof. Note that

$$\neg(\mathcal{M}_{p,q}(x_i)) \overset{D}{=} \mathcal{M}_{1-p,1-p}(x_i) \overset{D}{=} \mathcal{M}_{1-q,1-p}(\neg x_i).$$

Consequently,

$$\neg(\mathcal{M}_{p,q}(x)) \overset{D}{=} \mathcal{M}_{p,q}(\neg x) \iff p = 1 - q \iff q = 1 - p.$$

□

In contrast with the results that leveraged randomized merging, we demonstrate that a deterministic merge for a given Boolean operation requires specific choices of randomized response mechanism, precluding the use of general Boolean operations under a single RR mechanism (Corollary 1.16). In particular, we demonstrate that a deterministic *and* ($\wedge$) merge requires a different privacy mechanism than the *or* ($\vee$) merge described in Theorem 1.3 (Corollary 1.15). Here again, we defer the proofs of these results to Appendix A, noting that the analysis is similar to that performed for Theorem 1.3.

Corollary 1.15. *Let $f_1 = \mathcal{F}_{p_1,q_1}, f_2 = \mathcal{F}_{p_2,q_2}, f_3 = \mathcal{F}_{p_3,q_3}$, and let $\bullet : \{0,1\}^2 \to \{0,1\}$ denote a deterministic and symmetric operation. The following conditions may only be satisfied simultaneously if $\circ = \veebar$ and $q_1 = q_2 = 1/2$:*

1. *f_1, f_2 are (respectively) ε_1-DP and ε_2-DP for $\varepsilon_1, \varepsilon_2 < \infty$.*
2. *$f_1(x) \bullet f_2(y) \overset{D}{=} f_3(x \wedge y)$.*
3. *$f_i(0) \overset{D}{\neq} f_i(1)$ for $i = 1, 2, 3$.*

From here, it follows that no privacy mechanism $\mathcal{M}_{p,q}$ can satisfy the conditions of Theorem 1.3 and Corollary 1.15 simultaneously.

Corollary 1.16. *Assume the setting of Theorem 1.3. Let $\bullet : \{0,1\}^2 \to \{0,1\}$ denote a deterministic and symmetric operation. It is impossible to satisfy conditions (1)–(3) of Theorem 1.3 in addition to the following:*

4. *$f_1(x) \bullet f_2(y) \overset{D}{=} f_3(x \wedge y)$.*

1.5 Cardinality Estimation

The Sketch-Flip-Merge summary developed so far satisfies privacy and mergeability requirements, but it remains to show how the SFM summary may be used to estimate cardinalities. We develop a composite likelihood–based estimator that is consistent and asymptotically optimal. We give an analytic estimator of the error that closely matches the true error in our experiments.

1.5.1 Composite-Likelihood Estimator

Likelihood-based approaches to cardinality estimation have been used in the non-private setting (Clifford and Cosma, 2012; Lang, 2017; Ertl, 2017; Ting, 2019), where they have demonstrated

greater accuracy than competing estimators for PCSA. While true maximum likelihood estimation of n given a sketch is computationally infeasible due to non-independence of bits in the sketch (Ting, 2019; Ertl, 2017), the marginal likelihood for any bit is easy to derive. Similar to Ting (2019), we derive a composite marginal likelihood estimator (Lindsay, 1988; Varin et al., 2011) for n.

Let C_{ij} denote the number of unique items $x \in D$ mapped to bucket i and value j in the sketch and ρ_{ij} be the probability an item is mapped to that location. Then, assuming the use of universal random hashes, the following generative process describes the SFM summary $T = \mathcal{M}_{p,q}(\mathcal{S}(D))$.

$$\begin{aligned}(C_{11}, \ldots, C_{BP}) &\sim \text{Multinomial}(n, \rho_{11}, \ldots, \rho_{BP}) \\ T_{ij} \mid C_{ij} &\sim \mathcal{F}_{p,q}(1(C_{ij} > 0))\end{aligned}$$

While the joint distribution of $\{T_{ij}\}_{i,j}$ involves a sum over integer partitions of n into at most BP parts and is intractable, the marginal distribution of a single bit T_{ij} is easy to compute. Note that cell C_{ij}'s probability of occupancy ρ_{ij} only depends on j, with $\rho_{ij} = 2^{-\min\{j,P-1\}}/B$. Then $\mathbf{P}(C_{ij} = 0 \mid n) = \gamma_j^n$, where $\gamma_j = 1 - \rho_{ij}$, and

$$T_{ij} \sim \text{Bernoulli}\left(p(1 - \gamma_j^n) + q\gamma_j^n\right).$$

The composite marginal log-likelihood (Lindsay, 1988; Varin et al., 2011) replaces the true log-likelihood by the surrogate $\ell_{p,q}(n; t)$ that sums over marginal log-probabilities,

$$\ell_{p,q}(n; t) = \sum_{ij}(1 - t_{ij}) \log\left(1 - p + (p - q)\gamma_j^n\right) + \sum_{ij} t_{ij} \log\left(p - (p - q)\gamma_j^n\right),$$

where $t = \mathcal{M}_{p,q}(\mathcal{S}(D))$ denotes a realized SFM summary. The corresponding composite maximum likelihood estimator is $\hat{n} = \max_n \ell_{p,q}(n; t)$ and can be optimized by Newton's method. The required first and second derivatives of $\ell_{p,q}$ are

$$\begin{aligned}\ell'_{p,q}(n; t) &= \sum_{ij}(1 - t_{ij}) \frac{(p-q)\gamma_j^n \log(\gamma_j)}{1-p+(p-q)\gamma_j^n} - \sum_{ij} t_{ij} \frac{(p-q)\gamma_j^n \log(\gamma_j)}{p-(p-q)\gamma_j^n}, \\ \ell''_{p,q}(n; t) &= \sum_{ij}(1 - t_{ij}) \frac{(1-p)(p-q)(\log \gamma_j)^2 \gamma_j^n}{(1-p+(p-q)\gamma_j^n)^2} - \sum_{ij} t_{ij} \frac{p(p-q)(\log \gamma_j)^2 \gamma_j^n}{(p-(p-q)\gamma_j^n)^2}.\end{aligned}$$

In the absence of privacy (i.e., $p = 1, q = 0$), $\ell_{p,q}(\,\cdot\,; t)$ is strictly concave. While this is not true in the private case, Theorem 1.17 states that the expectation of $\ell_{p,q}$ remains concave over the interval $(0, n + \Delta)$ for some $\Delta > 0$.

Theorem 1.17. *Let D be a multiset such that $|\texttt{set}(D)| = n$. Let $T = \mathcal{M}_{p,q}(\mathcal{S}(D))$. Let $f(\hat{n}) = \mathbb{E}[\ell_{p,q}(\hat{n}; T)]$, where the expectation is taken over the randomness of the hash functions h_1, h_2 and the privacy mechanism $\mathcal{M}_{p,q}$. Then $f(\hat{n})$ attains a global maximum at $\hat{n} = n$ and is concave on an interval containing $(0, n]$ in its interior.*

Proof. $f(\hat{n})$ and its second derivative are derived and analyzed in Appendix A. □

1.5.2 Theoretical Results

We choose to use composite marginal likelihood due to its attractive theoretical properties. In particular, the use of a true likelihood, even if incomplete, ensures that cardinality estimates are asymptotically consistent, and the Hessian of the composite likelihood provides an estimate of the variance (Ting, 2019). We further show the cardinality estimates are asymptotically optimal in the typical case when the cardinality is large relative to the sketch size.

Theorem 1.18. *Let S_n denote a PCSA summary of n distinct items with $B(n)$ buckets and $P = \infty$ levels. Let $\overline{S}_n$ denote a modified PCSA summary of* $\text{Poisson}(n)$ *distinct items, and $\tilde{S}_n$ denote one where the composite marginal likelihood is the true likelihood. If $B \log B = o(\sqrt{n})$, then there exists modified PCSA summaries $\tilde{S}_n, \overline{S}_n$ where*

$$\mathbf{P}(S_n = \tilde{S}_n = \overline{S}_n) \to 1 \quad \text{as } n \to \infty.$$

Corollary 1.19. *The composite likelihood estimator of the SFM sketch's cardinality is asymptotically efficient in the asymptotic regime in Theorem 1.18.*

We outline the proofs of Theorem 1.18 and Corollary 1.19 here and provide detailed proofs in Appendix A. The main difficulty is that entries in an SFM summary are dependent, since each item can only be allocated to one cell. By constructing a coupling between sketches $S_n, \overline{S}_n$ with dependent and independent entries, we show they are asymptotically equal. In these coupled processes, the bucket with maximum difference in item allocations can only differ by only a small amount, $O_p(\sqrt{n}/B \log B)$. By showing an item updates its bucket's sketch values with probability $O(1/v_i)$ (where v_i is the number of items in bucket i), we conclude the coupled sketches are, in fact, equal with probability going to 1 when the average bucket allocation $v_i \approx n/B$ grows fast enough to make the small differences in item allocation irrelevant. Since $\overline{S}_n, \tilde{S}_n$ have independent bits, we couple them via the inverse CDF method and directly bound the probability that they differ. Since the sketches are the same asymptotically, applying the exact same RR noise to them implies their privatized versions are the same, and any estimator on them has the same asymptotic sampling

distribution. Therefore, the asymptotically efficient MLE for the independent-entry sketch, i.e., the composite likelihood estimator, is also asymptotically efficient for the true SFM summary S_n.

Remark 1.20. *This result also proves the MLE derived under the approximation that each bin has* Poisson(n/B) *items is asymptotically efficient. This can be extended to HyperLogLog and other sketches to show pseudo-likelihood based estimators (Ertl, 2017; Ting, 2019) that have good empirical properties are, in fact, asymptotically optimal.*

1.5.3 Error Estimation

Like the Fisher information matrix for MLE's, the inverse Godambe (or sandwich) information provides a consistent estimate of the estimator's variance. The Godambe information is $G(n) = H(n)J^{-1}(n)H(n)$ where $-J(n)$ is the Hessian of the expected log composite likelihood at n and $H(n) = \mathrm{Var}(\ell'_{p,q}(n, T))$ is the variance of the composite score functions. In the non-private case, Ting (2019) demonstrated that composite marginal likelihood variance estimates for HyperLogLog based on Fisher information and Godambe information are nearly identical for large cardinalities and that the Fisher information overestimates the variance at small cardinalities due to the negative dependence of buckets. Figure 1.2 shows this overestimation is much less pronounced when independent randomized response noise is added. Thus, we use only the Hessian to define the *estimated standard error* as

$$\begin{aligned}\widehat{\mathrm{SE}}_{p,q}(B, P, n) &= \sqrt{1/\mathbb{E}[-\ell''_{p,q}(n; T)]} \\ &= \left[B(p-q)\sum_{j=1}^{P}(\log\gamma_j)^2\gamma_j^n\left(\frac{p}{p-(p-q)\gamma_j^n} - \frac{1-p}{1-p+(p-q)\gamma_j^n}\right)\right]^{-1/2}, \end{aligned} \tag{1.2}$$

where $T = \mathcal{M}_{p,q}(S)$ for a random PCSA sketch S of size $B\times P$ and cardinality n. Figure 1.4 in Section 1.6 demonstrates empirically that our error estimates $\widehat{\mathrm{SE}}_{p,q}(B, P, n)$ are a good approximation for the error.

1.6 Evaluation

We evaluate our methods on both real-world and synthetic datasets. We demonstrate empirically that the SFM summary is the first mergeable ε-DP distinct counting sketch with practical performance, since the errors for the Pagh and Stausholm (2021) sketch are impractically large. Among private sketches, our novel randomized-merge sketch construction dominates the deterministic-

merge sketches. Thus, our improvements on the construction, estimation, and privacy analysis yield practical gains. Moreover, our theoretical error closely approximates empirical error.

1.6.1 Experiment Setup

We consider four different private distinct counting sketches in our experiments. Among our methods, *SFM (sym)* pairs $\mathcal{M}_\varepsilon^{\text{sym}}$ with our randomized merge procedure, while *SFM (xor)* pairs $\mathcal{M}_\varepsilon^{\text{xor}}$ with the deterministic *xor* merge. Both SFM methods use the estimator of Section 1.5. We compare these methods against the sketch and estimator of Pagh and Stausholm (2021) implemented two ways: *PS (loose)* constructs sketches using $\mathcal{M}_\varepsilon^{\text{PS}} = \mathcal{M}_{p,q}$ with $p = 1/2, q = 1/(2+\varepsilon)$ as prescribed in Pagh and Stausholm (2021), while *PS (tight)* uses the tightened $\mathcal{M}_\varepsilon^{\text{xor}}$ (Definition 1.4). By Corollary 1.7, the sketch construction of *PS (tight)* at ε is equivalent to that of *PS (loose)* at $\varepsilon' = 2(e^\varepsilon - 1)$.

We also consider two non-private sketches as baseline comparisons in our final experiment, noting that we should not expect the accuracy of a private sketch to be as strong as a non-private one. In what follows, *FM85* denotes non-private PCSA using the estimator of Flajolet and Martin (1985), and *HLL* denotes HyperLogLog (Flajolet, Fusy, et al., 2007). We compare HLL sketches against PCSA sketches at equal bucket counts, noting that the HLL sketches are smaller per bucket than the corresponding PCSA sketches.

We measure estimation error primarily in the form of *relative root mean squared error (RRMSE)*, defined as

$$\text{RRMSE}(\hat{n}_1, \ldots, \hat{n}_m; n) = \frac{1}{n}\sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{n}_i - n)^2}.$$

We also measure the *relative efficiency* of two methods as the ratio of their mean squared error,

$$\text{RE}(\hat{n}^{(1)}, \hat{n}^{(2)}) = \frac{\text{MSE}(\hat{n}^{(2)})}{\text{MSE}(\hat{n}^{(1)})}.$$

If two sketches have unbiased estimators, a relative efficiency of r indicates that the less efficient sketch must asymptotically use r times more buckets to get the same accuracy. This is because the asymptotic MSE (variance) decreases proportionally to $1/B$ for these estimators.

The simulations use sketches with dimensions $B = 4096$, $P = 24$ by default, using the xxHash64 (Collet, 2022) hash function, averaged over $m = 1000$ trials.

Modification to Pagh and Stausholm (2021). In our experiments, the original estimator of Pagh and Stausholm (2021) frequently failed to produce an estimate. For a desired error tolerance β, the method computes an interval for each of the P levels of the sketch, then intersects them to

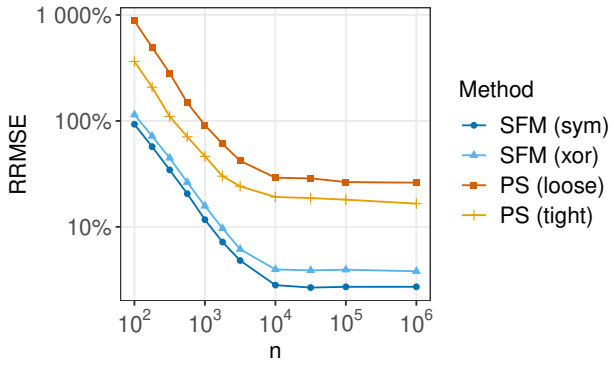


Figure 1.1. RRMSE vs. n at $\varepsilon = 1$ on log–log axes, compared across the four methods. RRMSE stabilizes for large n.

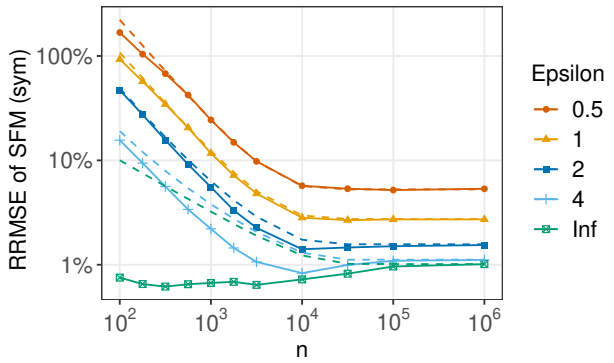


Figure 1.2. Dashed lines show estimated relative error, $\widehat{\mathrm{SE}}/n$, which is highly accurate for large n and small ε.

produce an estimate. However, this intersection was frequently empty for small β. To patch this, we perform a binary search for the smallest β resulting in a non-trivial intersection. We use the midpoint of this interval as our estimate of n.

Data Sources. Our experiments use both synthetic and real data. Synthetic data consist of random sets of integers with a fixed cardinality. Real data is taken from the BitcoinHeist paper (Akcora et al., 2020), which provides a database derived from a network of $N = 2{,}917{,}697$ Bitcoin transactions to $n = 2{,}631{,}095$ unique addresses. Note that cardinality estimation algorithms are insensitive to the value distribution of inputs since values are hashed as part of the processing.

1.6.2 Results

Figure 1.1 compares the accuracy of the four private methods on synthetic data as the cardinality n ranges from 10^2 to 10^6 given a fixed privacy budget of $\varepsilon = 1$. For large cardinalities, RRMSE tends

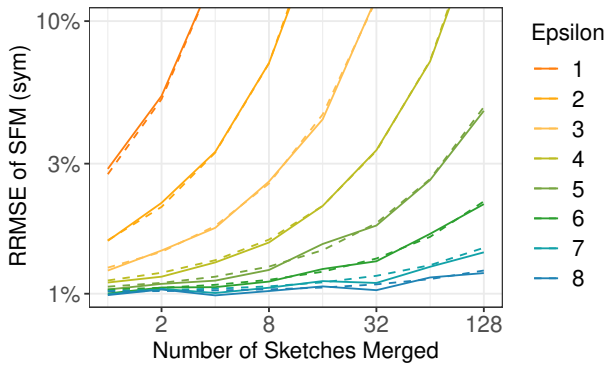


Figure 1.3. RRMSE at $n = 10^6$ after merging a given number of *SFM (sym)* summaries, each with a given privacy budget ε. Dashed lines show estimated relative error.

toward a fixed constant for each method. For small cardinalities, the relative error increases as the cardinality decreases, which is expected for differentially private methods. Figure 1.2 compares the accuracy for multiple values of ε but only for the best sketch, *SFM (sym)*. It also shows that the RRMSE stabilizes as $n \to \infty$ regardless of the choice of ε. In contrast to DP methods, the SFM summary with infinite privacy budget yields especially accurate estimates at small n. Figure 1.2 further shows that the estimated relative error $\widehat{\text{SE}}/n$ (Eq. 1.2) is an upper bound on the empirical error and yields a good estimate of RRMSE, especially for large cardinalities or small ε.

Figure 1.3 demonstrates the tradeoff between merging and privacy in SFM summaries at large cardinality ($n = 10^6$). Merge operations result in an accumulation of noise, requiring the use of larger privacy budgets to accommodate greater merge counts. The estimated relative error here is calculated according to Remark 1.11 and once again closely matches empirical error.

We also compare private methods against the real-world BitcoinHeist (Akcora et al., 2020) data over a variety of privacy budgets ε, ranging from 0.25 to 4. Here the true cardinality is $n =$ 2,631,095. The left panel of Figure 1.4 shows the relative efficiency of *SFM (sym)* as compared with the other private methods. *SFM (sym)* is uniformly more efficient than the PS estimators by at least an order of magnitude. Moreover, *SFM (sym)* outperforms *SFM (xor)* with relative efficiency tending toward 4 for larger ε, indicating that for larger privacy budgets, the randomized-merge *SFM (sym)* can achieve comparable accuracy to *SFM (xor)* in as little as one fourth the space. The estimation error from this experiment is depicted in absolute terms in the right panel, where we again see that the estimated relative error for SFM is a good approximation for RRMSE.

Finally, we compare SFM to popular non-private alternatives and show that error similarly decreases with the bucket count. Using synthetic data with cardinality $n = 10^6$, we construct sketches of varying bucket count B, using a privacy budget of $\varepsilon = 2$ for SFM. Figure 1.5 shows

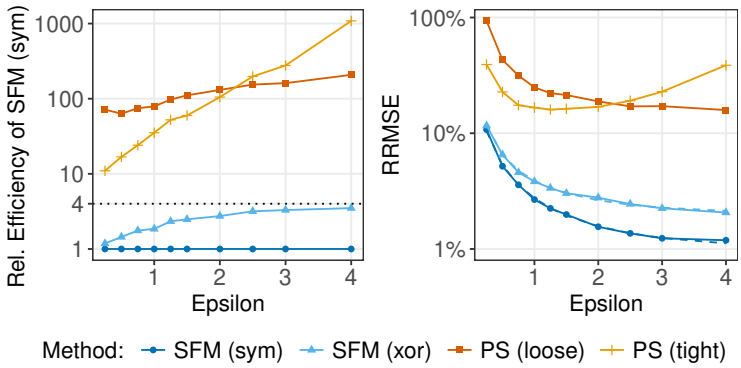


Figure 1.4. *(left)* Relative efficiency of *SFM (sym)* vs. other methods on BitcoinHeist data ($n \approx 2.6$ million). SFM makes large efficiency gains over PS. *SFM (sym)* tends toward 4x the efficiency of *SFM (xor)* for larger privacy budgets. *(right)* RRMSE vs. ε. For SFM, dashed lines show estimated relative error.

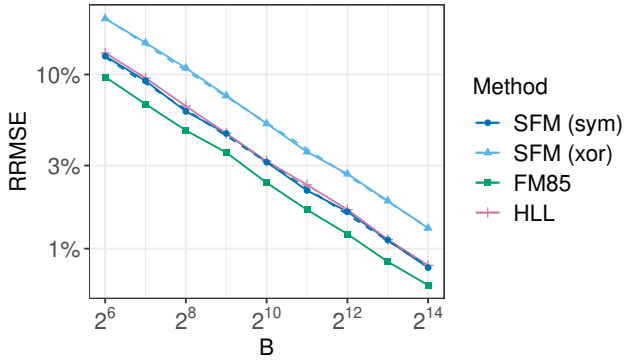


Figure 1.5. RRMSE at $n = 10^6$ vs. bucket count B for private sketches at $\varepsilon = 2$ vs. common non-private alternatives (on log–log axes). For SFM, dashed lines show estimated relative error. For all methods, RRMSE scales with $B^{-0.5}$.

RRMSE as a function of B for each method. Like the familiar non-private distinct counting sketches, our RRMSE decreases with $B^{-0.5}$. Thus, like non-private sketches, the RRMSE of our DP summaries can be easily characterized by a simple formula $c_\varepsilon/\sqrt{B}$ at large cardinalities, where c_ε is a constant specific to a method and privacy budget.

1.7 Discussion and Conclusion

The Sketch-Flip-Merge summaries demonstrate dramatic improvement over the current state-of-the-art mergeable and differentially private distinct-count sketches. This is achieved through novel merge algorithms (Theorem 1.10 and Section 1.4.3), asymptotically optimal estimation (Section 1.5), and an improved privacy analysis (Corollary 1.7).

An important limitation in mergeable private summaries is the inherent tension between privacy and mergeability. While both are attainable, repeated merging in the private setting degrades accuracy. This tradeoff, argued in the general distinct-count setting by Desfontaines et al. (2019), is explicitly quantified for SFM in Remark 1.11 and Figure 1.3.

Finally, we note the generality of some of our findings. In particular, our methods for aggregating noisy binary data provide fundamental machinery and a quantification of the noise-compounding effects of bitwise operations under randomized response that apply to a wide array of problems, particularly in the privacy-preserving space.

Chapter 2

Consistent Spectral Clustering under Edge Differential Privacy

Abstract. The stochastic block model (SBM) and degree-corrected block model (DCBM) are network models often selected as the fundamental setting in which to analyze the theoretical properties of community detection methods. We consider the problem of spectral clustering of SBM and DCBM networks under a local form of edge differential privacy. Using a randomized response privacy mechanism called the edge-flip mechanism, we develop theoretical guarantees for differentially private community detection, demonstrating conditions under which this strong privacy guarantee can be upheld while achieving spectral clustering convergence rates that match the known rates without privacy. We prove the strongest theoretical results are achievable for dense networks (those with node degree linear in the number of nodes), while weak consistency is achievable under mild sparsity (node degree greater than $\sqrt{n}$). We empirically demonstrate our results on a number of network examples.

Publication. This chapter and the accompanying Appendix B are adapted from a joint publication with Aleksandra Slavković and Xiaoyue Niu that appeared in the *Journal of Privacy and Confidentiality*.[1] As first author, I developed the main results, conducted the simulations, and led the drafting process.

Code. A Python implementation of our method, including simulation code and data, may be found at `https://github.com/jonhehir/private-spectral-clustering` (permanently archived in Hehir, 2022b).

[1] Jonathan Hehir, Aleksandra Slavkovic, and Xiaoyue Niu (Nov. 2022). "Consistent Spectral Clustering of Network Block Models under Local Differential Privacy". In: *Journal of Privacy and Confidentiality* 12.2. DOI: 10.29012/jpc.811.

2.1 Introduction

In the field of network data analysis, a common problem is community detection, the algorithmic discovery of clusters or communities of dense connection. There exist numerous parametric network models that exhibit community structure. Two fundamental models used in the analysis of community detection algorithms are the stochastic block model (SBM), first introduced in Holland et al. (1983), and the degree-corrected block model (DCBM) of Karrer and Newman (2011), a popular SBM extension that allows for networks with more realistic heterogeneity in node degree.

We consider the problem of estimating latent community structure in networks using SBM and DCBM via the method of spectral clustering. In the absence of privacy, this problem has gained popularity, as spectral clustering has been shown to be a computationally tractable method for group estimation with satisfying theoretical guarantees (McSherry, 2001; Ng et al., 2002; Rohe et al., 2011; T. Qin and Rohe, 2013; Lei and Rinaldo, 2015; Joseph and Yu, 2016; Binkiewicz et al., 2017; Abbe, 2018; Abbe, Fan, and K. Wang, 2022). In this setting, the clusters of a given network are informed by the relationships contained within that network. These relationships are represented by the edges of the network, which may reflect friendships, partnerships, collaborations, communications, financial transactions, similarities, or other interactions between some set of entities, say, people or businesses. In many cases, these relationships may constitute sensitive or confidential information, and so we may desire a clustering that reveals high-level structures in the network while preserving the privacy of low-level relationships.

To provide such a privacy guarantee, we turn to a form of local differential privacy (DP) for networks (see, e.g., Imola et al. (2021) for networks, and Duchi et al. (2013) for more general ideas on local DP). We assume that the nodes of the network are known but that their relationships are sensitive. We protect these relationships through a randomized-response mechanism applied to the edges of the network. Previous works, such as Mülle et al. (2015) and Karwa, Krivitsky, et al. (2017), have used similar techniques to obtain a synthetic network that satisfies ε-DP (of Dwork et al., 2006) with respect to the edges of the network. Recent works (Z. Qin et al., 2017; Imola et al., 2021) have emphasized that these synthetic networks can be constructed in a distributed fashion, requiring that no single party has knowledge of the full set of true network relationships. We construct our synthetic networks in such a manner, then apply a modified spectral clustering algorithm to obtain consistent estimates of group membership for SBM and DCBM networks.

The problem of performing DP community detection has previously been given empirical consideration (Mülle et al., 2015; Nguyen et al., 2016; Z. Qin et al., 2017), but to our knowledge,

our analysis of the cost of privacy in this setting is the first theoretical treatment of the subject. Our work generalizes the non-private results of Lei and Rinaldo (2015) to the DP setting, and we demonstrate certain conditions under which our DP estimator matches the known non-private convergence rates. We attribute these results to certain desirable properties of the edge-flip mechanism that we explore in some detail. While this method performs well for dense networks, the magnitude of error introduced by the local DP mechanism results in a slowing of the convergence rate for sparse networks and a requirement that node degree grows faster than $\sqrt{n}$. We derive these results in the context of more general finite-sample and asymptotic bounds on misclassification rates with privacy.

This chapter is structured as follows: We first introduce the network models and notation in Section 2.2. In Section 2.3, we review differential privacy and define the edge-flip mechanism, and we demonstrate that the edge-flip mechanism can be viewed as a mixture distribution (Lemma 2.7), leading to two corollaries: a closure property for SBM and a more general post-processing step that allows us to preserve the expected spectral embedddings of edge-flipped networks. In Section 2.4, we propose a modified spectral clustering algorithm based on this method—with concentration bounds for the estimation of spectral embeddings of SBM and DCBM networks proven in Lemma B.1. Our main theoretical results on finite-sample bounds on misclassification and asymptotic convergence rates for private spectral clustering are presented in Section 2.5. Proofs of these results are deferred to Appendix B. In Section 2.6, we evaluate the performance of private spectral clustering on both simulated and observed networks. Finally, we conclude with a discussion of limitations and open questions in Section 2.7.

2.2 Network Models and Notation

SBM and DCBM are models for binary, undirected networks without covariates. Edges in both networks occur as independent Bernoulli random variables. Each of the n nodes is assigned to one of k communities, or *blocks*, and these memberships are denoted in the parameter $\theta \in [k]^n$ (where $[k] = \{1, \ldots, k\}$). In SBM, the probability of a given edge occurring between nodes i and j depends only on the blocks to which i and j belong, with those block-to-block edge probabilities recorded in the symmetric matrix $B \in (0, 1]^{k \times k}$. In DCBM, edge probabilities further depend on a parameter $\psi \in (0, 1]^n$ that determines the relative expected node degree for each node. If we let Y

be the symmetric $n \times n$ adjacency matrix from a DCBM, the elements of Y are then distributed:

$$Y_{ij} \overset{\text{ind}}{\sim} \begin{cases} \text{Bernoulli}(\psi_i \psi_j B_{\theta_i \theta_j}), & i < j \\ 0, & i = j \\ Y_{ji}, & i > j \end{cases}$$

For such a network, we will write $Y \sim \text{DCBM}(\theta, \psi, B)$. The stochastic block model can be regarded as a special case of the DCBM, where $\psi = \mathbf{1}_n$, a vector of n ones, i.e.,

$$\text{SBM}(\theta, B) \overset{D}{=} \text{DCBM}(\theta, \mathbf{1}_n, B), \tag{2.1}$$

where $\overset{D}{=}$ indicates equality in joint distribution of network edges.

We let $C_j = \{i : \theta_i = j\}$ denote the set of nodes in the j-th community. We denote the size of the j-th community in an SBM or DCBM (i.e., the number of nodes in a block) as $n_j = |C_j|$, the smallest of which we denote $n_{\min} = \min_{j \in [k]} n_j$, the largest as $n_{\max} = \max_{j \in [k]} n_j$, and the second-largest as $n'_{\max}$. As in Lei and Rinaldo (2015), we denote the effective size of the j-th block in a DCBM as $\tilde{n}_j = \sum_{i \in C_j} \psi_i^2$ and the largest effective size of a block as $\tilde{n}_{\max} = \max_{j \in [k]} \tilde{n}_j$. As a measure of heterogeneity within the j-th block, we use $\nu_j = n_j^{-2} (\sum_{i \in C_j} \psi_i^{-2})(\sum_{i \in C_j} \psi_i^2)$.

We use λ_B to denote the smallest absolute nonzero eigenvalue of the matrix B. The largest entry in B is denoted $\max B = \max_{ij} B_{ij}$. We represent networks via adjacency matrices, e.g., $Y \in \{0,1\}^{n \times n}$. The i-th row of the matrix Y is denoted Y_{i*}. We use $\|\mathbf{x}\|_p$ to denote the ℓ_p norm of a vector $\mathbf{x}$, $\|A\|_F$ to denote the Frobenius norm of a matrix, and $\|A\|$ to denote the operator norm of the matrix A, i.e., $\|A\| = \sup_{\|\mathbf{x}\|_2 = 1} \|A\mathbf{x}\|_2$.

In asymptotic notation, we write $a_n = o(b_n)$ when $|a_n / b_n| \to 0$ as $n \to \infty$; $a_n = \omega(b_n)$ when $|a_n / b_n| \to \infty$ as $n \to \infty$; $a_n = O(b_n)$ when $|a_n / b_n| \leq C$ for some $C > 0$ and all n; $a_n = \Omega(b_n)$ when $|a_n / b_n| \geq C$ for some $C > 0$ and all n; and $a_n = \Theta(b_n)$ when $a_n = O(b_n)$ and $a_n = \Omega(b_n)$. Finally, we write $X_n = O_P(b_n)$ if for any $\alpha > 0$ there exists a constant M such that $P(|X_n / b_n| > M) < \alpha$ for all large n.

2.3 Privacy Mechanism as a Mixture Distribution

2.3.1 Defining Privacy

The framework of differential privacy offers more than just a formal privacy definition: it offers *many* privacy definitions that build on each other. To fully appreciate the privacy implications of

a given definition—and thereby to justify our specific choice of definition—we briefly review the core concepts of DP and discuss the nuanced distinctions between privacy definitions that could be applied to the problem at hand. Consider first the typical definition of DP:

Definition 2.1 (Differential Privacy (Dwork et al., 2006)). *Let $\varepsilon > 0, \delta \in [0, 1)$. Let $\mathcal{Y}, \mathcal{Z}$ be sets, and let $d : \mathcal{Y} \times \mathcal{Y} \to \mathbb{Z}_{\geq 0}$ be an integer-valued metric. Let $\mathcal{M} : \mathcal{Y} \to \mathcal{Z}$ be a randomized algorithm. We say $\mathcal{M}$ satisfies (ε, δ)–**differential privacy** with respect to d if for any $y, y' \in \mathcal{Y}$ satisfying $d(y, y') = 1$ and any $E \subseteq \mathcal{Z}$, we have:*

$$P(\mathcal{M}(y) \in E) \leq e^{\varepsilon} \cdot P(\mathcal{M}(y') \in E) + \delta. \tag{2.2}$$

*If $\delta = 0$, we say $\mathcal{M}$ satisfies ε–**differential privacy**.*

Such an algorithm $\mathcal{M}$ is called a *privacy mechanism*. The elements y and y' are often referred to as *databases* and thought of as a collection of records from some set, say, $\mathcal{Y} = \mathbb{R}^n$ or $\mathcal{Y} = \{0, 1\}^n$. The metric d is usually chosen to be the Hamming distance, such that (2.2) can be interpreted thus: changing any one record in an arbitrary database $y \in \mathcal{Y}$ does not significantly affect the distribution of $\mathcal{M}(y)$.[2] Consequently, one can gain virtually no information about a single record based on the private output $\mathcal{M}(y)$. The parameters ε and δ quantify the strength of the privacy guarantee, with smaller values conferring a stronger guarantee. The case when $\delta > 0$ is often referred to as *approximate differential privacy*, while the case where $\delta = 0$ is called *pure differential privacy*. The parameter ε is commonly called the *privacy-loss budget*.

In the network setting, there are two primary choices for the metric d that are frequently employed, and these correspond to the notions of edge DP and node DP (Kasiviswanathan et al., 2013; Z. Qin et al., 2017):

Definition 2.2 (Edge DP). *Let Y, Y' be two networks with n nodes. Then $d_{edge}(Y, Y')$ is given by the total number of edges that differ between Y and Y'. If $\mathcal{M}$ satisfies (ε, δ)–DP with respect to d_{edge}, then we say $\mathcal{M}$ satisfies (ε, δ)–**edge differential privacy**.*

Definition 2.3 (Node DP). *Let Y, Y' be two networks with n nodes. Then $d_{node}(Y, Y')$ is given by the minimum number of nodes in Y whose incident edges could be modified in order to obtain Y'. If $\mathcal{M}$ satisfies (ε, δ)–DP with respect to d_{node}, then we say $\mathcal{M}$ satisfies (ε, δ)–**node differential privacy**.*

Since $d_{\text{edge}}(Y, Y') \geq d_{\text{node}}(Y, Y')$, the definition of node DP is more strict than the definition of edge DP. Indeed, node DP implies edge DP, but the reverse is not true. In many cases, the

[2]Note that the databases y, y' are not presented as random quantities in this definition. One can think of $\mathcal{M}(y)$ as a distribution conditioned on the observed data, as in Wasserman and Zhou (2010).

strictness of node DP precludes meaningful analysis (Kasiviswanathan et al., 2013). This is the case, for example, in the problem we wish to solve. Suppose $\hat{\theta}(Y)$ is an ε–DP estimator of the group membership θ for the network Y with respect to a distance measure d. This implies that when we look at any given node's estimated label, $[\hat{\theta}(\cdot)]_i$, across two networks Y, Y' satisfying $d(Y, Y') = 1$, we must satisfy:

$$\frac{P([\hat{\theta}(Y)]_i = \ell)}{P([\hat{\theta}(Y')]_i = \ell)} \in [e^{-\varepsilon}, e^{\varepsilon}] \quad \forall \ell.$$

For edge DP ($d = d_{\text{edge}}$), this means that changing any single edge in the network cannot significantly affect the distribution of the estimated label for a given node. For node DP ($d = d_{\text{node}}$), however, we can take any given node and change any subset (or even all) of its incident edges without significantly affecting the distribution of its estimated label. Since our goal is to infer labels from precisely these edges, this notion of privacy is too strict for our purposes. For this reason, we will focus on edge-based definitions of DP, which aim to protect the privacy of individual relationships within the network.[3]

So far, the DP definitions we have considered all fall under the umbrella of *central DP*. In central DP, we assume that one party, often called the *trusted curator*, has complete knowledge of the true database y. This arrangement is not always desirable, which motivates the concept of *local DP* (Duchi et al., 2013). In local DP, records in the database are held by a number of distributed parties (e.g., users of a social network), and only these parties require true knowledge of their records. To facilitate some central analysis of the database, a randomized algorithm is independently applied to each record to produce a *local differentially private view* of that record. These local DP views may then be shared with a central processor for further analysis. Thus, in contrast with central DP, where a single mechanism is applied at the database level by a single database owner, local DP applies privacy mechanisms at the record level, eliminating the need for a trusted curator, while still allowing for centralized analysis of the data.

Applying local DP to the edge setting, we have further choices yet. Likely the most widely known definition of local DP in the edge setting is *edge local DP* of Z. Qin et al. (2017), but we will instead focus on *relationship DP* as defined in Imola et al. (2021). This definition is more tailored to the setting of undirected networks, where it provides a stronger privacy guarantee.[4] In relationship DP (as in edge local DP), each node reports a private view (or summary) of its

[3]Edge-based definitions of DP implicitly assume that the only sensitive information in the network is encoded in these relationships. For example, in a community detection setting, we assume that the identities of nodes are not sensitive but that their relationships are.

[4]More precisely, in an undirected network, ε–edge local DP implies 2ε–relationship DP (Imola et al., 2021).

neighbor list, the set of nodes with which it shares an edge. The formal definition is given below.

Definition 2.4 (Relationship DP (Imola et al., 2021))**.** *Let $\varepsilon > 0$, and let $\mathcal{M}_1, \ldots, \mathcal{M}_n$ be randomized algorithms with domain $\{0,1\}^n$. The algorithm $\mathcal{M}(Y) = (\mathcal{M}_1(Y_{1*}), \ldots, \mathcal{M}_n(Y_{n*}))$ is said to satisfy ε–**relationship DP** if for each $E \subseteq \text{Range}(\mathcal{M})$ and networks Y, Y' satisfying $d_{edge}(Y, Y') = 1$, we have:*

$$P(\mathcal{M}(Y) \in E) \leq e^{\varepsilon} \cdot P(\mathcal{M}(Y') \in E).$$

2.3.2 The Edge-Flip Mechanism

The specific privacy mechanism we employ is a simple randomized-response mechanism that produces an ε-relationship-DP synthetic copy $\mathcal{M}_\varepsilon(Y)$ of the original network Y by randomly flipping edges in Y. This can be performed locally by assigning each node to flip and self-report a subset of its edges. In this way, no single party ever needs full knowledge of the true adjacency matrix Y. Since differential privacy is closed under post-processing (Dwork et al., 2006), any analysis performed on the synthetic network $\mathcal{M}_\varepsilon(Y)$ preserves ε–relationship DP. Edge-flipping mechanisms have been utilized in several earlier papers and interpreted under various edge DP definitions, including (central) edge DP (Mülle et al., 2015; Karwa and Slavković, 2016; Karwa, Krivitsky, et al., 2017), edge local DP (Z. Qin et al., 2017), and relationship DP (Imola et al., 2021). We include here an explicit adaptation of the edge-flipping procedure from Imola et al. (2021), which we term the *symmetric edge-flip mechanism*.

Definition 2.5 (Symmetric Edge-Flip Mechanism)**.** *Let $\varepsilon > 0$. For $i \in [n]$, let $\mathcal{M}_i : \{0,1\}^n \to \{0,1\}^n$ such that:*

$$[\mathcal{M}_i(\mathbf{x})]_j \overset{ind}{=} \begin{cases} 0 & i \geq j \\ 1 - \mathbf{x}_j & i < j \quad w.p. \quad \frac{1}{1+e^{\varepsilon}} \\ \mathbf{x}_j & i < j \quad w.p. \quad \frac{e^{\varepsilon}}{1+e^{\varepsilon}} \end{cases},$$

and let

$$T(Y) = \begin{bmatrix} \mathcal{M}_1(Y_{1*}) \\ \vdots \\ \mathcal{M}_n(Y_{n*}) \end{bmatrix}.$$

*Then the $n \times n$ **symmetric edge-flip mechanism** is the mechanism $\mathcal{M}_\varepsilon(Y) = T(Y) + [T(Y)]^T$.*

Theorem 2.6. *The symmetric edge-flip mechanism $\mathcal{M}_\varepsilon$ satisfies ε–relationship DP.*

Proof. The proof of this follows from Theorem 3 of Imola et al. (2021) and the corresponding discussion. For completeness, we give a formal proof in Section B.5. □

The simplicity of the edge-flip mechanism affords it several key advantages: In practice, it is easy and flexible to implement. In theory, the distribution of the resulting synthetic network is transparent and tractable. In fact, in Lemma 2.7 we show that the network generated by the edge-flip mechanism is a mixture of the original non-private network with an Erdős–Rényi network. This in turn leads to closure and statistical inference properties that are important and useful, both theoretically and practically. These properties are demonstrated in the corollaries that follow.

Lemma 2.7. *Let $Y \in \{0,1\}^{n\times n}$ be a random, undirected, binary network. Let $Z \sim G(n, \frac{1}{2})$ be an Erdős–Rényi random graph with n nodes and edge probability $\frac{1}{2}$, and let $U_{ij} \overset{ind}{\sim} \text{Bernoulli}\left(\frac{2}{e^\varepsilon+1}\right)$ for $1 \le i < j \le n$. Define $\mathcal{M}'(Y)$ to be the symmetric mixture network such that for $i < j$:*

$$\mathcal{M}'(Y)_{ij} \mid Y, Z, U = (Z_{ij})^{U_{ij}} (Y_{ij})^{(1-U_{ij})}.$$

Then $\mathcal{M}'(Y)$ is equal in distribution to $\mathcal{M}_\varepsilon(Y)$.

Proof. It is sufficient to show that the conditional distributions of $\mathcal{M}'(Y) \mid Y$ and $\mathcal{M}_\varepsilon(Y) \mid Y$ are equivalent. Conditioned on Y, the entries of $\mathcal{M}'(Y)$ and $\mathcal{M}_\varepsilon(Y)$ are independent, binary random variables, and:

$$\begin{aligned}
P(\mathcal{M}'(Y)_{ij} = 1 \mid Y) &= E[\mathcal{M}'(Y)_{ij} \mid Y] \\
&= E[\ E[\mathcal{M}'(Y)_{ij} \mid Y, Z, U] \mid Y] \\
&= E\left[U_{ij}Z_{ij} + (1-U_{ij})Y_{ij} \,\middle|\, Y\right] \\
&= \frac{1}{e^\varepsilon+1} + \frac{e^\varepsilon-1}{e^\varepsilon+1} Y_{ij} \\
&= \frac{1}{e^\varepsilon+1}(1-Y_{ij}) + \frac{e^\varepsilon}{e^\varepsilon+1} Y_{ij} \\
&= E[\mathcal{M}_\varepsilon(Y)_{ij} \mid Y] \\
&= P(\mathcal{M}_\varepsilon(Y)_{ij} = 1 \mid Y)
\end{aligned}$$

□

Lemma 2.7 provides useful intuition about the edge-flip mechanism. As $\varepsilon \to \infty$, our synthetic network $\mathcal{M}_\varepsilon(Y)$ approaches the true network Y, and as $\varepsilon \to 0$, we approach an Erdős–Rényi network. For block models, this is a welcome property, as it means the community structure of the network is exactly preserved. For the SBM in particular, we have closure of the model family under $\mathcal{M}_\varepsilon$. The following corollary follows from a simple extension of the proof of Lemma 2.7.

Corollary 2.8. *If $Y \sim \text{SBM}(\theta, B)$, then $\mathcal{M}_\varepsilon(Y) \sim \text{SBM}(\theta, \tau_\varepsilon(B))$, where:*

$$\tau_\varepsilon(B) = \frac{1}{e^\varepsilon+1}\mathbf{1}_k\mathbf{1}_k^T + \frac{e^\varepsilon-1}{e^\varepsilon+1}B. \tag{2.3}$$

Closure of a particular family of models under edge flipping can be used to obtain theoretical guarantees for differentially private procedures via direct application of non-private theory. SBM is not alone in holding this convenient property, with some extensions, such as the mixed-membership SBM of Airoldi et al. (2008), sharing a similar closure property. Unfortunately, not all SBM extensions afford such a closure property. In particular, DCBM is not closed under edge-flipping. For this reason, we will explore a more generally applicable framework here, relying on a weaker property of the edge-flip mechanism that holds for any random binary network: via a small "downshift" transformation to an edge-flipped network, we can recover a network whose expectation matches that of the original network, up to a scaling factor.

Corollary 2.9. *Let $Y \in \{0,1\}^{n\times n}$ be a random, undirected, binary network, and let $(d_\varepsilon \circ \mathcal{M}_\varepsilon)(Y) = \mathcal{M}_\varepsilon(Y) - (e^\varepsilon+1)^{-1}(\mathbf{1}_n\mathbf{1}_n^T - I_n)$ be the downshifted edge-flipped network. Then:*

$$E(d_\varepsilon \circ \mathcal{M}_\varepsilon)(Y) = \frac{e^\varepsilon - 1}{e^\varepsilon + 1} EY.$$

Proof. This follows from a slight rewriting of the derivation in the proof of Lemma 2.7, namely:

$$\begin{aligned} E[\mathcal{M}_\varepsilon(Y)] &= E[\ E[\mathcal{M}_\varepsilon(Y) \mid Y]\ ] \\ &= \frac{1}{e^\varepsilon+1}(\mathbf{1}_n\mathbf{1}_n^T - I_n) + \frac{e^\varepsilon-1}{e^\varepsilon+1}EY. \end{aligned}$$

□

Lemma 2.9 has important implications: for any random, undirected, binary network Y, the matrices $E[Y]$ and $E[(d_\varepsilon \circ \mathcal{M}_\varepsilon)(Y)]$ share the same eigenvectors. This fact leads to a more general post-processing step under DP that allows us to preserve the expected spectral embeddings of edge-flipped networks.

2.4 Modified Clustering Methods with Concentration Bounds

Our goal is to use spectral clustering to estimate the unobserved block membership parameter θ while satisfying ε-relationship DP. A variety of related clustering algorithms exist under the umbrella term of *spectral clustering* (Ng et al., 2002; Von Luxburg, 2007; Rohe et al., 2011; Lei and Rinaldo, 2015; Binkiewicz et al., 2017). In our setting, we focus on the method of adjacency spectral clustering with k known, following the framework of Lei and Rinaldo (2015). In adjacency spectral clustering, we place the leading k eigenvectors (by absolute value of corresponding eigenvalues) of the observed adjacency matrix Y into an $n \times k$ matrix $\hat{U}$. We consider each row of $\hat{U}$ to be the

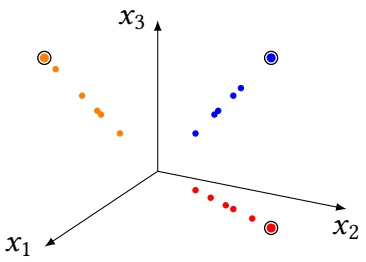


Figure 2.1. Hypothetical expected embeddings U for DCBM in $\mathbb{R}^3$, where points represent rows of U and colors represent blocks. (Circled points represent structure of expected SBM embeddings.)

spectral embedding of the respective node in the network, and we perform a clustering process over these embeddings: for SBM, we perform simple k-means, and for DCBM, we normalize the embeddings to have unit norm before performing k-medians clustering. The resulting estimated cluster memberships serve as our estimator, $\hat{\theta}$.

This approach is theoretically justified by demonstrating that the observed embeddings $\hat{U}$ concentrate around a set of "expected" embeddings U that satisfy appropriate geometric properties, as shown in Figure 2.1. In particular, let $Y \sim \text{DCBM}(\theta, \psi, B)$, let P be the $n \times n$ matrix with entries $P_{ij} = \psi_i \psi_j B_{\theta_i \theta_j}$, and let U be the $n \times k$ matrix that holds the leading k eigenvectors of P. (Note that $E[Y] = P$ except on the diagonal.) In the case of SBM, it can be shown that the rows of U correspond to k distinct points, and two nodes belong to the same block if and only if their expected embeddings are equal. For general DCBM, the rows of U fall along k rays emanating from the origin, with each ray corresponding to a unique block (Jin, 2015; Lei and Rinaldo, 2015).

Unfortunately, these geometric properties are not, in general, preserved by the edge-flipping routine: the embeddings of $\mathcal{M}_\varepsilon(Y)$ do not have the same expected geometric properties as the embeddings of Y. However, our Corollary 2.9 suggests a modified approach: Since $E[(d_\varepsilon \circ \mathcal{M}_\varepsilon)(Y)]$ and $E[Y]$ share the same eigenvectors, our intuition suggests that the embeddings of $(d_\varepsilon \circ \mathcal{M}_\varepsilon)(Y)$ and Y will concentrate in the same locations. Indeed, we prove such concentration bounds in Lemma B.1. This is also the key fact that powers the main results of Section 2.5. To perform spectral clustering on the edge-flipped network, then, we need only to modify the algorithms to use the downshifted adjacency matrix $(d_\varepsilon \circ \mathcal{M}_\varepsilon)(Y)$ in place of $\mathcal{M}_\varepsilon(Y)$. Our proposed modified algorithms for SBM and DCBM are given in Algorithms 1 and 2, respectively. The k-means and k-medians problems, as well as their approximations, are defined in Section B.1 of Appendix B.

To evaluate the performance of the clustering algorithms, we assume knowledge of ground truth group labels θ and measure two forms of misclassification. The first is an overall measure of misclassification that captures the proportion of misidentified nodes, up to a permutation of labelings. Let $S_{[k]}$ denote the set of all permutations $\sigma : [k] \to [k]$, and let $\mathbb{I}(\cdot)$ denote an indicator

Algorithm 1 Edge-Flipped Spectral Clustering (k-means)

Input: edge-flipped adjacency matrix $A \in \{0,1\}^{n \times n}$, number of blocks k, approximation error γ, privacy budget ε

Output: private estimate of block membership $\hat{\theta} \in [k]^n$

Let $A_\downarrow = \begin{cases} A & \varepsilon = \infty \\ A - (e^\varepsilon + 1)^{-1}(\mathbf{1}_n \mathbf{1}_n^T - I_n) & \varepsilon < \infty \end{cases}$.

Let $\hat{u}_1, \ldots, \hat{u}_k \in \mathbb{R}^n$ be the k leading eigenvectors (by absolute value) of $A_\downarrow$.

Let $\hat{\theta}$ be a $(1+\gamma)$-approximate solution to the k-means problem over the rows of $\hat{U}_\downarrow = [\hat{u}_1 \ \ldots \ \hat{u}_k] \in \mathbb{R}^{n \times k}$.

return $\hat{\theta}$

Algorithm 2 Edge-Flipped Spectral Clustering (Normalized k-medians)

Input: edge-flipped adjacency matrix $A \in \{0,1\}^{n \times n}$, number of blocks k, approximation error γ, privacy budget ε

Output: private estimate of block membership $\hat{\theta} \in [k]^n$

Let $A_\downarrow = \begin{cases} A & \varepsilon = \infty \\ A - (e^\varepsilon + 1)^{-1}(\mathbf{1}_n \mathbf{1}_n^T - I_n) & \varepsilon < \infty \end{cases}$.

Let $\hat{u}_1, \ldots, \hat{u}_k \in \mathbb{R}^n$ be the k leading eigenvectors (by absolute value) of $A_\downarrow$.

Let $\hat{U}_\downarrow = [\hat{u}_1 \ \ldots \ \hat{u}_k] \in \mathbb{R}^{n \times k}$.

 # construct row-normalized matrix $\hat{U}'_\downarrow$ over non-zero rows

Let $I_+ = \{i \in [n] : \|(\hat{U}_\downarrow)_{i*}\|_2 > 0\}$, $f : [|I_+|] \to [n]$ s.t. $f(i) = (I_+)_i$

for $i = 1, \ldots, |I_+|$ **do**

  Let $(\hat{U}'_\downarrow)_{i*} = (\hat{U}_\downarrow)_{f(i)*} / \|(\hat{U}_\downarrow)_{f(i)*}\|_2$

end for

Let $\hat{\theta}'$ be a $(1+\gamma)$-approximate solution to the k-medians problem over the rows of $\hat{U}'_\downarrow$.

Let $\hat{\theta}_i = \begin{cases} 1 & i \notin I_+ \\ \hat{\theta}'_{f^{-1}(i)} & i \in I_+ \end{cases}, \quad i \in [n]$

return $\hat{\theta}$

function. Then the *overall misclassification* for the estimated groups $\hat{\theta}$ is given by:

$$L(\theta, \hat{\theta}) = \min_{\sigma \in S_{[k]}} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(\sigma(\hat{\theta}_i) \neq \theta_i)). \tag{2.4}$$

As in Lei and Rinaldo (2015), we also consider the *worst-case misclassification* within a single

community in order to account for trivialities[5]. This error is given by:

$$\tilde{L}(\theta, \hat{\theta}) = \max_{j \in [k]} \min_{\sigma \in S_{[k]}} \frac{1}{|C_j|} \sum_{i \in C_j} \mathbb{I}(\sigma(\hat{\theta}_i) \neq j). \tag{2.5}$$

2.5 Misclassification Bounds for Private Clustering

We generalize the results of Lei and Rinaldo (2015) developed for the non-private network setting to derive finite-sample misclassification bounds for spectral clustering of SBM and DCBM under local differential privacy using the modified spectral clustering algorithms proposed in Section 2.4 (Algorithms 1 and 2). We argue in Section 2.4 and prove in Section B that the spectral embeddings of Y and $(d_\varepsilon \circ \mathcal{M}_\varepsilon)(Y)$ will concentrate in the same locations. In Lemma B.1, we also demonstrated that the concentration weakens with a stronger privacy guarantee (i.e., smaller ε). In this section, we use these concentration results to derive misclassification bounds for private spectral clustering of SBM and DCBM; all proofs are provided in Section B. We begin with the results for SBM.

2.5.1 SBM

Theorem 2.10. *Let $Y \sim \mathrm{SBM}(\theta, B)$ with B full rank, $\max B \geq \log n/n$, and minimum absolute eigenvalue $\lambda_B > 0$. Let $\varepsilon \in (0, \infty]$, and let $\hat{\theta}_\varepsilon$ be the result of Algorithm 1 on $\mathcal{M}_\varepsilon(Y)$ (where $\mathcal{M}_\infty(Y) = Y$), using an approximation error of γ. Let:*

$$g_\varepsilon(B) = \begin{cases} \frac{e^\varepsilon+1}{e^\varepsilon-1}\left(\max B + \frac{1}{e^\varepsilon-1}\right), & \varepsilon < \infty \\ \max B, & \varepsilon = \infty \end{cases} \tag{2.6}$$

There exists a universal constant c_1 such that if:

$$\frac{(2+\gamma)kn}{n_{min}^2 \lambda_B^2} g_\varepsilon(B) < c_1^{-1}, \tag{2.7}$$

[5]Consider an asymptotic regime with $k = 2$ where one community is of constant size N_1 and the other is of size $n - N_1$. In this setting, a trivial estimator such as $\hat{\theta} = \mathbf{1}_n$ can achieve consistency in the sense that $L(\theta, \hat{\theta}) \to 0$.

then with probability at least $1 - n^{-1}$*:*

$$\tilde{L}(\theta, \hat{\theta}_\varepsilon) \leq c_1 \frac{(2+\gamma)kn}{n_{min}^2 \lambda_B^2} g_\varepsilon(B)$$
$$\textit{and} \quad L(\theta, \hat{\theta}_\varepsilon) \leq c_1 \frac{(2+\gamma)kn'_{max}}{n_{min}^2 \lambda_B^2} g_\varepsilon(B).$$

Taking $\varepsilon = \infty$ (i.e., the non-private setting) in Theorem 2.10 recovers a result that is exactly equivalent to the results for SBM in Lei and Rinaldo (2015). Consequently, the cost of privacy on misclassification is captured by $g_\varepsilon(B)$. Noting that $g_\varepsilon(B)$ is a decreasing function in ε and that $\lim_{\varepsilon \to \infty} g_\varepsilon(B) = g_\infty(B)$, Theorem 2.10 provides a smooth interpolation between the known finite-sample misclassification bounds without privacy and the corresponding bounds under local DP.

We can use the finite-sample bounds from Theorem 2.10 to construct particular asymptotic regimes under which $\hat{\theta}_\varepsilon$ is consistent and derive convergence rates. For a given sequence of SBM networks, it suffices to show that:

$$\frac{kn}{n_{\min}^2} g_\varepsilon(B) = o(1).$$

If this holds, condition (2.7) is met for large n, and both measures of misclassification, $\tilde{L}(\theta, \hat{\theta}_\varepsilon)$ and $L(\theta, \hat{\theta}_\varepsilon)$, tend to zero.

Since $g_\varepsilon(B)$ explains the difference in classification performance between private and non-private clustering, a comparison of convergence rates can be boiled down to the differences in asymptotic behavior of $g_\varepsilon(B)$. We note that if $\varepsilon = \infty$, then $g_\varepsilon(B) = \max B$. This will serve as our baseline. On the other hand, if $\varepsilon < \infty$, then $g_\varepsilon(B) = O(\max B + \zeta_\varepsilon^{-1} + \zeta_\varepsilon^{-2})$, where $\zeta_\varepsilon = e^\varepsilon - 1$, an increasing function of ε satisfying $\zeta_0 = 0$. (See Fact B.2.) Thus we have:

$$\tilde{L}(\theta, \hat{\theta}_\varepsilon) = \begin{cases} O_P\left(\frac{kn}{n_{\min}^2 \lambda_B^2} \max B\right) & \varepsilon = \infty \\ O_P\left(\frac{kn}{n_{\min}^2 \lambda_B^2} (\max B + \zeta_\varepsilon^{-1} + \zeta_\varepsilon^{-2})\right) & \varepsilon < \infty \end{cases}$$
$$\text{and} \quad L(\theta, \hat{\theta}_\varepsilon) = \begin{cases} O_P\left(\frac{kn'_{\max}}{n_{\min}^2 \lambda_B^2} \max B\right) & \varepsilon = \infty \\ O_P\left(\frac{kn'_{\max}}{n_{\min}^2 \lambda_B^2} (\max B + \zeta_\varepsilon^{-1} + \zeta_\varepsilon^{-2})\right) & \varepsilon < \infty \end{cases}.$$

From this, it is clear that if ε and $\max B$ are bounded away from zero, we can obtain the same convergence rate under local DP as in the absence of privacy. For example, for a fixed privacy budget and a sequence of dense SBMs (i.e., one for which the edge probabilities do not tend to zero), we can effectively employ local DP with no cost to the convergence rate (up to a constant).

For a sparse SBM (one in which $\max B \to 0$) with fixed privacy budget, however, this does not hold, as $g_\varepsilon(B) = \Theta(1)$ for $\varepsilon < \infty$, but $g_\infty(B) = \max B = o(1)$. Thus the convergence rate for sparse SBM slows by a factor of $(\max B)^{-1}$.

This slowing of convergence for sparse SBM limits the extent of sparsity that can be handled under this local DP mechanism. Consider a regime in which k is fixed, communities grow proportionally ($n_{\min} = \Theta(n)$), and B changes with n only via some scaling parameter $\alpha_n \to 0$, i.e., $B = \alpha_n B_0$ for some fixed matrix B_0. In this case, we have k constant, $\max B = \Theta(\alpha_n)$, $\lambda_B = \Theta(\alpha_n)$, and so the worst-case misclassification is:

$$\tilde{L}(\theta, \hat{\theta}_\varepsilon) = \begin{cases} O_P\left(\frac{1}{n\alpha_n}\right) & \varepsilon = \infty \\ O_P\left(\frac{\alpha_n + \zeta_\varepsilon^{-1} + \zeta_\varepsilon^{-2}}{n\alpha_n^2}\right) & \varepsilon < \infty \end{cases}. \tag{2.8}$$

Theorem 2.10 allows us to choose α_n as small as $\log n/n$, so we have consistency for non-private clustering down to $\log n/n$. For private clustering with a fixed privacy budget, however, we need $\alpha_n = \omega(n^{-1/2})$. The only way to achieve greater sparsity is to allow the privacy budget to grow arbitrarily large. For example, choosing $\varepsilon = \log(1 + \alpha_n^{-1})$, we have $\zeta_\varepsilon^{-1} = \alpha_n$, and thus we can recover the same convergence rate and accommodate the same level of sparsity as without privacy—but at the cost of allowing unbounded privacy loss for large n.

2.5.2 DCBM

Generalizing the theoretical results for SBM to DCBM requires a bit of care. As described in Section 2.3, in the more general setting of DCBM, the expected embeddings for a given block fall along distinct rays emanating from the origin—in contrast with distinct points in the case of SBM. It is for this reason that Algorithm 2 is used for DCBM. The key theorethical result from this more general treatment is given below.

Theorem 2.11. *Let $Y \sim \mathrm{DCBM}(\theta, \psi, B)$ with $\max_{i \in C_j} \psi_i = 1$ for $j \in [k]$, B full rank, $\max B \geq \log n/n$, and minimum absolute eigenvalue $\lambda_B > 0$. Let $\varepsilon \in (0, \infty]$, and let $\hat{\theta}_\varepsilon$ be the result of Algorithm 2 on $\mathcal{M}_\varepsilon(Y)$ (where $\mathcal{M}_\infty(Y) = Y$), using an approximation error of γ. Let $g_\varepsilon(B)$ as in Theorem 2.10. There exists a universal constant c_2 such that if:*

$$\frac{(2.5 + \gamma)\sqrt{kn\, g_\varepsilon(B)}}{\tilde{n}_{min}\lambda_B} < c_2^{-1} \frac{n_{min}}{\sqrt{\sum_{j=1}^k n_j^2 \nu_j}}, \tag{2.9}$$

then with probability at least $1 - n^{-1}$*:*

$$L(\theta, \hat{\theta}_\varepsilon) \leq c_2 \frac{(2.5 + \gamma)}{\tilde{n}_{min} \lambda_B} \sqrt{\frac{k}{n} \left(\sum_{j=1}^{k} n_j^2 \nu_j \right) g_\varepsilon(B)}.$$

As was the case with SBM, the results of Theorem 2.11 for $\varepsilon = \infty$ (non-private setting) match the original results of Lei and Rinaldo (2015). The cost of privacy is once again determined by the function $g_\varepsilon(B)$. In this case, however, the bound changes with the square root of $g_\varepsilon(B)$.

Comparing the results of Theorems 2.10 and 2.11, the more general result involves additional parameters, of course, but also generally provides a weaker result. Ignoring constants, condition (2.9) is more stringent than (2.7), as $\tilde{n}_{\min} \leq n_{\min}$, and $n_{\min} / \sqrt{\sum_{j=1}^{k} n_j^2 \nu_j} \leq k^{-1/2}$. The resulting upper bound on $L(\theta, \hat{\theta}_\varepsilon)$ also results in weaker convergence rates. In keeping with Lei and Rinaldo (2015), Theorem 2.11 offers no bound on $\tilde{L}(\theta, \hat{\theta}_\varepsilon)$, as its proof bounds only the total number of misclassified nodes in the network. A trivial bound for $\tilde{L}$ can be obtained by observing that:

$$\tilde{L}(\theta, \hat{\theta}_\varepsilon) \leq (n / n_{\min}) L(\theta, \hat{\theta}_\varepsilon). \tag{2.10}$$

To simplify matters, we illustrate an asymptotic regime in which communities grow proportionally ($n_j = \Theta(n/k)$ for all $j \in [k]$) and assume that there exists a global lower bound $0 < a \leq \psi_i$ for $i \in [n]$. Under these conditions, we can state a simple asymptotic result.

Lemma 2.12. *Suppose* Y *satisfies the conditions of Theorem 2.11, and suppose further that* $0 < a \leq \psi_i \leq 1$ *for all* $i \in [n]$ *and* $n_j = \Theta(n/k)$ *for all* $j \in [k]$*. Then:*

$$\frac{k^2 \sqrt{g_\varepsilon(B)}}{a^3 \lambda_B \sqrt{n}} = o(1) \implies L(\theta, \hat{\theta}_\varepsilon) = O_P \left(\frac{k \sqrt{g_\varepsilon(B)}}{a^3 \lambda_B \sqrt{n}} \right).$$

To compare this to the results seen in Eq. (2.8), consider again the case when $B = \alpha_n B_0$ for some fixed matrix B_0 and sequence $\alpha_n \to 0$. Then since k is fixed, $n / n_{\min} = \Theta(1)$, and so combining Lemma 2.12 with Eq. (2.10) yields a convergence rate of:

$$\tilde{L}(\theta, \hat{\theta}_\varepsilon) = \begin{cases} O_P \left(\frac{1}{a^3 \sqrt{n \alpha_n}} \right) & \varepsilon = \infty \\ O_P \left(\frac{\sqrt{\alpha_n + \zeta_\varepsilon^{-1} + \zeta_\varepsilon^{-2}}}{a^3 \alpha_n \sqrt{n}} \right) & \varepsilon < \infty \end{cases}.$$

For SBM, where $a = 1$—or more generally for any regime in which a is bounded away from zero—this bound is precisely the square root of the bound obtained in Eq. (2.8). Once again,

we see that the convergence rate for sparse networks slows under private methods, and while some sparsity is attainable with privacy through this method, to accommodate the same level of sparsity with privacy as without, we would again need to allow the privacy-loss budget ε to grow arbitrarily large.

2.6 Empirical Evaluations

2.6.1 Simulation Studies

The theoretical results above suggest that the edge-flip privacy mechanism can be used to achieve convergence for spectral clustering of SBM and DCBM networks that is similar in rate for dense networks, while slower for sparse networks. Here we evaluate these results empirically through simulation. To facilitate this analysis, we consider two special cases of the SBM and DCBM:

Definition 2.13 (Symmetric SBM)**.** *The **symmetric SBM** is an SBM network consisting of n nodes, k equal-sized blocks, and $B = pI_k + r\mathbf{1}_k\mathbf{1}_k^T$. For such a network, we write $Y \sim \text{SSBM}(n, k, p, r)$.*

Definition 2.14 (Symmetric DCBM)**.** *The **symmetric DCBM** is a DCBM network consisting of n nodes, k equal-sized blocks, $B = pI_k + r\mathbf{1}_k\mathbf{1}_k^T$, and $\psi \in [a, 1]^n$ taking value 1 for the first node of a given block and values randomly drawn from $\text{Uniform}(a, 1)$ elsewhere. For such a network, we write $Y \sim \text{SDCBM}(n, k, p, r, a)$.*

Starting with SBM, we simulate two regimes following the symmetric SBM. In the first setting, we use a dense symmetric SBM. We consider 16 values of n ranging from $n = 30$ to $n = 12000$ and $\varepsilon \in \{0.5, 0.75, 1, 1.5, 2, 3, 4, \infty\}$, where $\varepsilon = \infty$ represents the original network (i.e., no privacy). For each n, ε pair, we draw 100 networks $Y^{(i)} \sim \text{SSBM}(n, k = 3, p = 0.2, r = 0.05)$. We then apply Algorithm 1 to $\mathcal{M}_\varepsilon(Y^{(i)})$, and report the mean overall misclassification rate $L(\theta^{(i)}, \hat{\theta}^{(i)})$ of Eq. (2.4). The results of this experiment are plotted on log–log axes in the left panel of Figure 2.2. Each curve, corresponding to a different value of ε, appears to be approximately linear and match the rest in slope, suggesting that they have approximately equal polynomial rates of convergence. This is consistent with the results of Section 2.5.

For the second setting, we use a sparse symmetric SBM. Here we consider 13 values of n ranging from $n = 10$ to $n = 12800$ and the same eight values of ε as in the first setting. For each n, ε pair, we draw 100 networks $Y^{(i)} \sim \text{SSBM}(n, k = 2, p = 1.5n^{-.3}, r = .15n^{-.3})$, then apply Algorithm 1 to $\mathcal{M}_\varepsilon(Y^{(i)})$, as in the first setting. The results are plotted on log–log axes in the right panel of Figure 2.2. Here, the curves corresponding to each value of ε are no longer parallel: for smaller values of ε, the convergence rate slows.

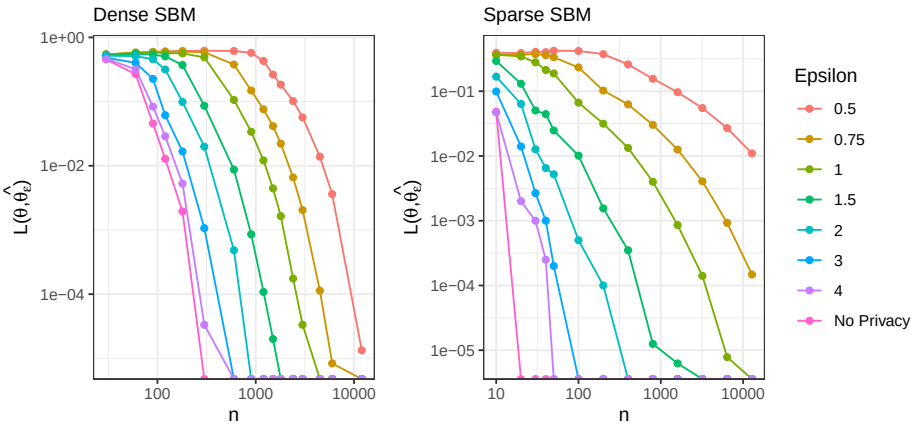


Figure 2.2. Proportion of misclassified nodes in simulated SBM networks for various values of ε, n. Left (dense): $Y \sim \text{SSBM}(n, k = 3, p = 0.2, r = 0.05)$, Right (sparse): $Y \sim \text{SSBM}(n, k = 2, p = 1.5n^{-.3}, r = .15n^{-.3})$

For DCBM, we simulate two additional regimes, one dense and one sparse, in a manner similar to the SBM simulations. In the dense setting, we use a dense symmetric DCBM with the same values of n, ε as used in the dense SBM simulations. For each n, ε pair, we draw 100 networks $Y^{(i)} \sim \text{SDCBM}(n, k = 3, p = 0.4, r = 0.05, a = 0.3)$. In the sparse setting, we use the same ε values as in the other settings and 12 values of n ranging from $n = 20$ to $n = 12800$. Then we draw 100 networks $Y^{(i)} \sim \text{SDCBM}(n, k = 2, p = 2n^{-.25}, r = 0.1n^{-.25}, a = 0.3)$. In both DCBM settings, we report the mean overall misclassification rate after applying Algorithm 2. The results of these simulations are plotted on log–log axes in Figure 2.3. In contrast with the SBM simulations, the DCBM simulations generally exhibit slower convergence. Similar to the SBM simulations, we see a clear slowing of convergence rate for smaller values of ε in the sparse setting.

While the results of all these simulations are consistent with the theory of Section 2.5, the precise nature in which convergence rate slows with ε in a sparse regime is not fully captured by the theoretical results. Indeed, for larger values of ε, the convergence properties are similar to the non-private setting. In general, based on these simulations, it seems the convergence bounds given here and in Lei and Rinaldo (2015) are not tight, in the sense that the observed convergence rates appear to beat the theoretical guarantees, sometimes by considerable order.

2.6.2 Performance on Observed Networks

To assess the practicality of these methods, we applied the edge-flip mechanism and Algorithm 2 to several real-world datasets, then compared the performance of spectral clustering on the private networks over various privacy budgets. We show the trade-offs of privacy loss and accuracy.

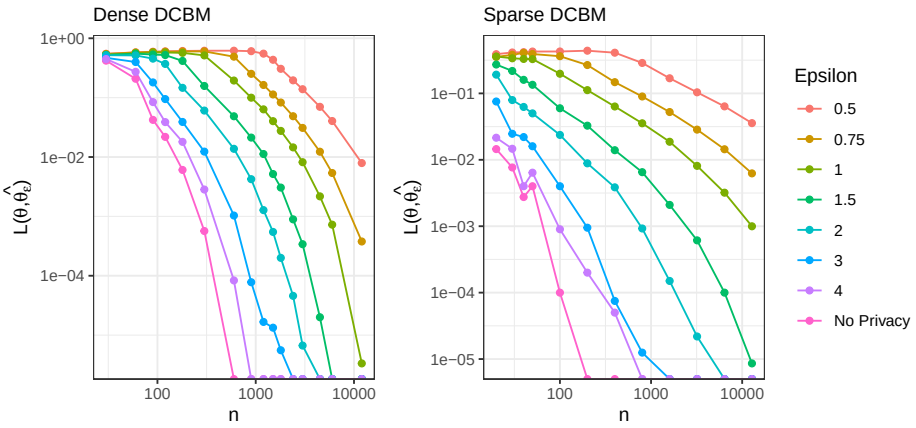


Figure 2.3. Proportion of misclassified nodes in simulated DCBM networks for various values of ε, n. Left (dense): $Y \sim \text{SDCBM}(n, k = 3, p = 0.4, r = 0.05, a = 0.3)$, Right (sparse): $Y \sim \text{SDCBM}(n, k = 2, p = 2n^{-.25}, r = 0.1n^{-.25}, a = 0.3)$

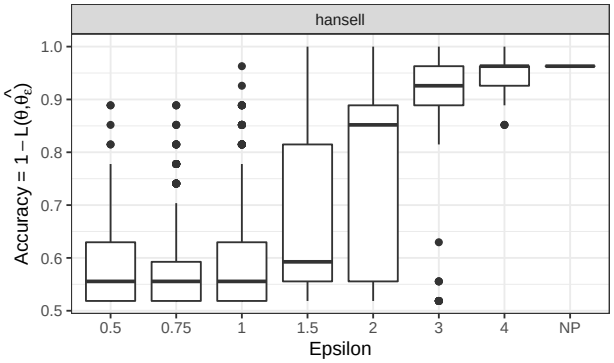


Figure 2.4. Trade-off of privacy-loss and accuracy of private spectral clustering on a small dataset, Hansell's friendship data (Hansell, 1984), with varying privacy-loss ε, and "NP" denoting non-private data.

The first dataset we considered is a small, classical network used in the SBM literature: Hansell's friendship data (Hansell, 1984), which was used in Y. J. Wang and Wong (1987) and Snijders and Nowicki (1997). This is a directed network of $n = 27$ (13 male, 14 female) students, and the presence of an edge (i, j) indicates that student i considers student j to be a friend. We symmetrized the network by setting $Y_{ij} = \max\{Y_{ij}, Y_{ji}\}$, and we used the students' sexes as ground-truth labels.

For this small network, we considered $\varepsilon \in \{0.5, 0.75, 1, 1.5, 2, 3, 4, \infty\}$, and for each value of ε, we applied the privacy mechanism and spectral clustering algorithm 500 times. Box plots showing the overall classification accuracy, $1 - L(\theta, \hat{\theta})$ from Eq. 2.4, for each value of ε are given in Figure 2.4. Accuracy remains high for values of $\varepsilon > 2$; for such a small network, this is about as good as we can expect.

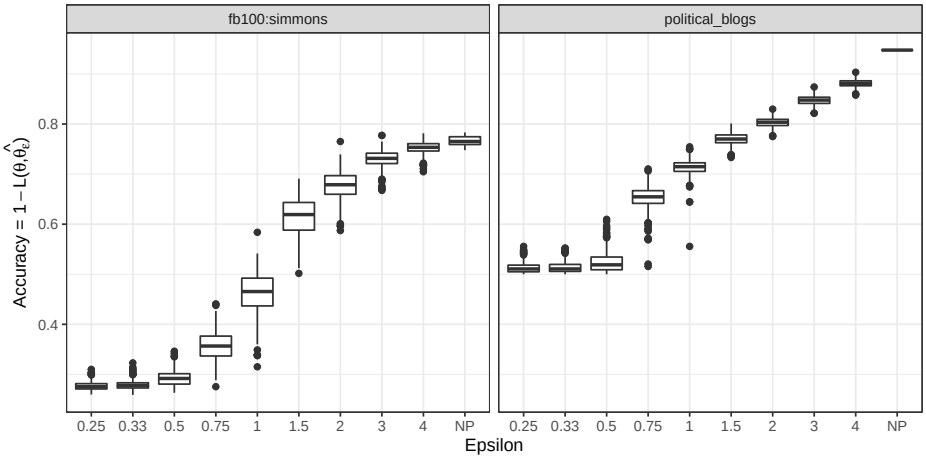


Figure 2.5. Trade-off of privacy-loss and accuracy of private spectral clustering on well-known DCBM datasets, with varying privacy-loss ε, and "NP" denoting non-private data. Left: Facebook friendships of Simmons College students (Traud et al., 2012), Right: political blogs network (Adamic and Glance, 2005)

Next, we considered two datasets from the DCBM literature. The first network consists of $n = 1137$ Facebook users from Simmons College (Traud et al., 2012; Y. Chen et al., 2018), and edges represent friendships. Each student belongs to one of $k = 4$ class years (2006 to 2009). These class years are used as ground-truth group memberships, which are inferred through their friendships. The second dataset is the well-known network of political blogs from Adamic and Glance (2005), which has been widely used in the DCBM literature (Karrer and Newman, 2011; Jin, 2015; Y. Chen et al., 2018; Abbe, 2018). This network consists of $n = 1222$ blogs on U.S. politics, which have been categorized as either left-leaning or right-leaning ($k = 2$ groups). An edge in this network represents the existence of a hyperlink between the two blogs, and these hyperlinks are used to infer the left- or right-leaning label for each node. The performance of the private spectral clustering methods on these networks is shown in Figure 2.5. In both of these datasets, we see a steady decline in performance as ε decreases, without a clear inflection point.

The theoretical and simulated results from earlier suggest that these methods are best paired with suitably large or dense networks. A relevant collection of networks can be found, for example, in Andris et al. (2015), where networks of Democrat and Republican members of the U.S. House of Representatives are constructed based on voting similarity. Based on their methods and data from Lewis et al. (2021), we reconstructed these networks for a number of sessions of the U.S. House of Representatives and U.S. Senate (Hehir, 2022a).[6]

[6]Although these networks are produced from public information, networks involving public figures like congress-people illustrate a case where edge privacy is particularly relevant, since the set of nodes is public knowledge, but their interactions or relations may be sensitive.

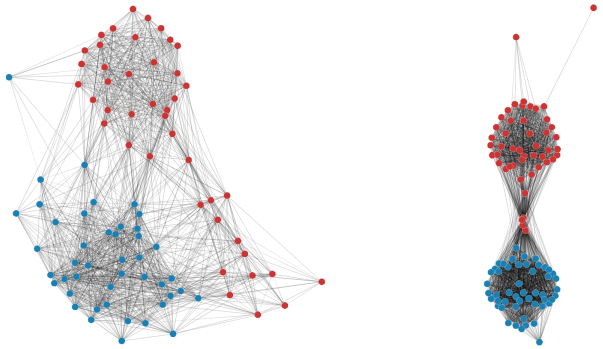


Figure 2.6. Visualization of congressional voting networks for 70th U.S. Senate (left) and 110th U.S. Senate (right). Democrats are depicted as blue nodes and Republican as red.

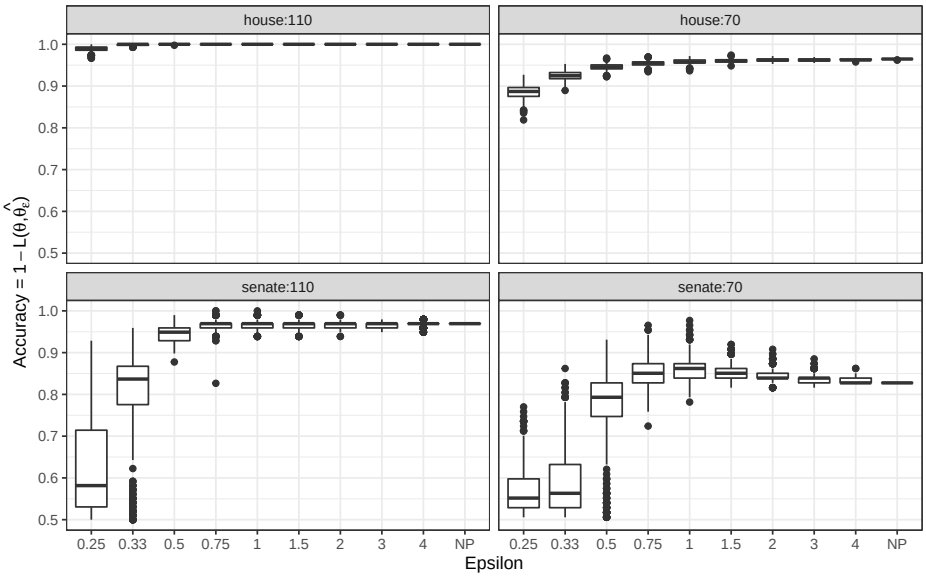


Figure 2.7. Trade-off of privacy-loss and accuracy of private spectral clustering on U.S. Congressional voting networks. Clockwise from top left: 110th House of Representatives, 70th House of Representatives, 70th Senate, 110th Senate

Changes over time in the patterns of voting among congresspeople offer us a range of networks that span nearly complete separation of the parties (particularly in recent years) to networks that exhibit greater levels of connectivity across parties. Figure 2.6 depicts two such networks, with the 70th U.S. Senate on the left and the 110th U.S. Senate on the right. In the 70th Senate, 13% of cross-party pairs are connected, while 76% and 59% of Democrat and Republican pairs are connected (respectively). In the 110th Senate, 9% of cross-party pairs are connected, while 98% and 88% of Democrat and Republican pairs are connected (respectively).

For the 70th Senate ($n = 87$), 110th Senate ($n = 98$), 70th House ($n = 425$), and 110th

House ($n = 423$), we applied the private spectral clustering methods 50 times for each value of $\varepsilon \in \{0.25, 0.33, 0.5, 0.75, 1, 1.5, 2, 3, 4, \infty\}$. The results are depicted in Figure 2.7. The high density and relatively low variability of degree in these networks enables impressive clustering performance: for the House networks, clustering performance is solid for all ε considered; in the smaller Senate networks, we see a drop in performance for values of $\varepsilon < 0.5$.

2.7 Discussion

Just as SBM and DCBM are fundamental network models featuring community structure, the edge-flip mechanism is a fundamental network privacy mechanism providing local differential privacy on networks at the edge level. A great advantage of the edge-flip mechanism is that it produces a synthetic network with clear distributional properties. In particular, the explicit mixture distribution of $\mathcal{M}_\varepsilon(Y)$ described in Lemma 2.7—in contrast with the unclear distributional properties of some alternative mechanisms—greatly facilitates the process of accounting for privacy in various statistical procedures. Moreover, by nature of producing a synthetic network, any number of statistical procedures can be applied to the single output $\mathcal{M}_\varepsilon(Y)$ with a fixed privacy budget ε.

Our work demonstrates how to use the clear distribution of $\mathcal{M}_\varepsilon(Y)$ to extend the theory of consistent community detection to the local DP setting. This is, to our knowledge, the first attempt to address this important problem. Algorithms 1 and 2 represent modest extensions to the algorithms originally proposed in Lei and Rinaldo (2015), requiring only the addition of a small "downshift" transformation to unlock a powerful privacy guarantee with a clear accounting of the cost to clustering performance. In fact, Lemma 2.8 suggests that for SBM networks in particular, this downshift transformation is optional, as $\mathcal{M}_\varepsilon(Y)$ is itself an SBM whose community structure is identical to Y.

Focusing on the specific case of SBM, we may ask whether it is preferable to perform clustering for SBM on $(d_\varepsilon \circ \mathcal{M}_\varepsilon)(Y)$, as prescribed by Algorithm 1, or to proceed with ordinary spectral clustering on $\mathcal{M}_\varepsilon(Y)$ per the closure property of Lemma 2.8. By combining the closure property with the $\varepsilon = \infty$ case of Theorem 2.10, one can obtain performance bounds in terms of the parameter $\tau_\varepsilon(B)$. In some cases, these bounds may be superior to the bounds resulting from Theorem 2.10 alone.[7] Nonetheless, we have opted to present the results for the downshifted procedure for the sake of generality. This allows us to keep the SBM misclassification bounds in terms of λ_B instead of $\lambda_{\tau_\varepsilon(B)}$, where a general relationship between these two quantities is elusive.

[7]For example, if B is positive-definite, then one can show that $g_\infty(\tau_\varepsilon(B))/\lambda^2_{\tau_\varepsilon(B)} < g_\varepsilon(B)/\lambda_B^2$ via Weyl's inequality.

The benefits of the edge-flip mechanism's tractable distribution are, of course, not limited to the network models considered in this chapter, nor to the field of network clustering. For example, Karwa, Krivitsky, et al. (2017) have previously shown how to adjust inferential procedures for exponential random graph models using the edge-flip mechanism. These examples are likely only scratching the surface of the theory that can be extended to accommodate privacy under the edge-flip mechanism.

The primary drawback to the edge-flip mechanism is its relationship to sparse networks. We posit that this limitation is inherent to working with a local-DP synthetic network, in which the magnitude of noise required for privacy begins to dwarf the signal present in a sparse network. Looking at the edge-flip mechanism in particular, for a sparse network Y, the mixture $\mathcal{M}_\varepsilon(Y)$ will be considerably more dense, and a disproportionate number of its edges will result from the privacy mechanism, not the original network of interest. Although these changes preserve key properties of SBM and DCBM networks, reductions in sparsity dilute the signal-to-noise ratio considerably. While the theory supports the claim that these methods are consistent for SBM and DCBM networks exhibiting mild sparsity, empirical performance for sparse networks indicates a considerable utility loss when a very strong privacy guarantee is applied to a sparse network. Other privacy mechanisms preserving the sparsity of a network, especially non-local mechanisms, could be considered for greater performance in these cases.

The trade-off of privacy-loss and accuracy that we observe for label recovery on dense networks such as the Congressional voting networks, where considerable accuracy is maintained at $\varepsilon = 0.5$, suggests that the methods employed here are of more than just theoretical interest. By comparison, in their work with exponential random graph models and the edge-flip mechanism, Karwa, Krivitsky, et al. (2017) use values of ε ranging from 3 to 6 on a network of similar size. Nonetheless, it would be useful to develop theoretical results for alternative privacy mechanisms that can accommodate greater sparsity.

Lastly, while the results given here focus on spectral clustering of SBM and DCBM, some of the insights from these techniques are suggestive of further theoretical developments in related fields. For example, concentration bounds for the spectral embeddings of the downshifted private network could likely be extended in the more general family of random dot product graphs (Athreya, Fishkind, et al., 2017). This opens a wide range of avenues for future work.

Chapter 3

Perfect Spectral Clustering with Discrete Covariates

Abstract. Among community detection methods, spectral clustering enjoys two desirable properties: computational efficiency and theoretical guarantees of consistency. Most studies of spectral clustering consider only the edges of a network as input to the algorithm. Here we consider the problem of performing community detection in the presence of discrete node covariates, where network structure is determined by a combination of a latent block model structure and homophily on the observed covariates. We propose a spectral algorithm that we prove achieves perfect clustering with high probability on a class of large, sparse networks with discrete covariates, effectively separating latent network structure from homophily on observed covariates. To our knowledge, our method is the first to offer a guarantee of consistent latent structure recovery using spectral clustering in the setting where edge formation is dependent on both latent and observed factors.

Publication. This chapter and the accompanying Appendix C are adapted from a joint publication with Xiaoyue Niu and Aleksandra Slavković, under review at the time of this dissertation.[1] As first author, I developed the main results, software, and simulations, and I led the drafting process.

Code. A Python implementation of our proposed method, including simulation code and additional examples, is available at `https://github.com/jonhehir/acsbm` (permanently archived in Hehir, 2023).

3.1 Introduction

A structural pattern commonly observed in social networks is *homophily*, the tendency for two nodes sharing a certain trait to be more (or sometimes less) likely to form a connection (McPherson et al., 2001). Homophily may occur on any number of traits, observed or latent, and is known to confound problems of causal inference in the social sciences (K. P. Smith and Christakis, 2008; Shalizi and Thomas, 2011; Goldsmith-Pinkham and Imbens, 2013; Lee and Ogburn, 2021).

[1] Jonathan Hehir, Xiaoyue Niu, and Aleksandra Slavkovic (2022). "Perfect Spectral Clustering with Discrete Covariates". In: *arXiv preprint arXiv:2205.08047*.

Homophily, meanwhile, lies at the heart of such issues as segregation (Shrum et al., 1988; Henry et al., 2011), job access (Ibarra, 1992), and political partisanship (Huber and Malhotra, 2017), where homophily on observed traits may be the subject of estimation in its own right. In order to fully understand the effects of network patterns like observed homophily, we first need to separate them from further latent network structure.

In the literature on community detection, latent structure is frequently recovered through a clustering process involving only the network edges, reserving node covariates to validate the clustering results in an approach that conflates latent structure with observed structure (Peel et al., 2017). What we wish to do instead is to separate the latent from the observed structural patterns. To this end, we consider an extension of the stochastic block model (SBM) (Holland et al., 1983) that incorporates homophily on observed, discrete node covariates into a generalized linear model (GLM). We define this model, which we call the *additive-covariate SBM (ACSBM)*, in Section 3.2. The model was previously studied by Mele et al. (2022) and allows for flexible modeling choices in which latent communities take a block model structure, covariates may or may not depend on community membership, and the effects of homophily may be modeled through a range of link functions. We give an explicit representation of this model as an SBM (Proposition 3.3), which motivates the use of spectral clustering to estimate the latent structure.

In the context of SBMs, spectral clustering is known as a fast method that achieves consistency in community detection down to established recovery thresholds (McSherry, 2001; Von Luxburg, 2007; Rohe et al., 2011; Lei and Rinaldo, 2015; Su et al., 2019; Abbe, Fan, K. Wang, and Zhong, 2020). In Section 3.3 of this work, we propose a computationally efficient spectral algorithm for recovering the latent structure of the ACSBM. Building on techniques from the field of random dot product graphs (Young and Scheinerman, 2007; Rubin-Delanchy et al., 2022), we develop new algebraic tools to synthesize latent structure over an ACSBM network partitioned by its covariate data. We are able to prove that our method recovers the latent communities of the ACSBM perfectly for sufficiently large networks with node degree at least polylogarithmic in n. Our theoretical analysis is outlined in Section 3.4, with proofs and derivations deferred to Appendix C. We provide simulation-based evidence in Section 3.5 and apply our method to network of Facebook friendships among Harvard students in Section 3.6. In the Harvard example, we see both strong and subtle homophily over observed covariates (class year and gender, respectively), and we uncover additional latent structure not explained by these covariates using our method. We conclude with a discussion of the results, their implications, and future generalizations in Section 3.7.

Related Work. Community detection with covariates is a very active area of research, with a

wide variety of methods for modeling community structure, estimating effects of covariates in edge formation, and recovering community memberships. Studies that demonstrate consistency in community recovery assume a generating process with ground-truth communities. Quite commonly, these generating processes feature conditional independence between covariates and edges, given community memberships (e.g., Binkiewicz et al., 2017; Deshpande et al., 2018; Yang et al., 2013; Tallberg, 2004; Newman and Clauset, 2016; Weng and Feng, 2021). In these models, any two nodes belonging to the same latent community have the same connectivity patterns, regardless of their observed covariates.

Explicit separation of latent from observed effects in edge formation is possible in models lacking this conditional independence structure. Such models include (e.g., Hoff, 2007; Handcock et al., 2007; D. S. Choi et al., 2012; Vu et al., 2013; Sweet, 2015; Huang and Feng, 2018; Mele et al., 2022; Zhang et al., 2019; Roy et al., 2019; Ma et al., 2020), many of which could be considered broader cases of the model we consider. For example, (Hoff, 2007; Handcock et al., 2007; Ma et al., 2020) model latent network structure via more general latent position models, which include SBM as a special case. The remainder focus more explicitly on extending SBM but usually allow greater flexibility in the role of covariates, up to and including allowing arbitrary edge covariates. Since working with SBM likelihood is computationally expensive (Snijders and Nowicki, 1997), many of these studies rely on approximate methods; only a small handful offer methods that scale to large networks and carry a theoretical guarantee of consistent classification. In particular, Huang and Feng (2018) provides a consistency guarantee for spectral clustering only when covariates are independent of community membership, and Ma et al. (2020) provides guarantees only under the assumption of a positive semi-definite latent structure. Our results do not require these assumptions.

By far the most similar paper to ours is Mele et al. (2022), which considers the same model, ACSBM, but under a different spectral estimation method. The main results concern estimation of covariate effects, while we focus on consistency of latent community recovery. Moreover, the results of Mele et al. (2022) implicitly rely on strong assumptions about the community structure that we wish to avoid (see Section 3.3) and require node degrees of larger order than $\sqrt{n}$. A follow-up paper (Mu et al., 2022) proposes a modification to the algorithm to improve robustness, but results are limited to the specific case of a single covariate under the identity link, with linear node degree.

Contribution. We propose a novel spectral algorithm that is computationally efficient and yields perfect clustering for sufficiently large ACSBM networks with high probability. We prove this result for networks with node degree at least polylogarithmic in n in which homophily effects

are multiplicative on the probabilty of edge formation; empirical results suggest greater generality. To our knowledge, our method is the first to offer a guarantee of consistent latent structure recovery using spectral clustering in the important setting where edge formation is dependent on both latent and observed factors.

Notation. Let $[n] = \{1, \ldots, n\}$, with $S_{[n]}$ denoting the set of all permutations $[n] \to [n]$. The function $\mathbb{I}(\cdot)$ is the indicator function. We represent networks as adjacency matrices, e.g., $Y \in \{0,1\}^{n \times n}$. The i-th row of the matrix Y is denoted Y_{i*}, and the i-th column Y_{*i}. $\mathbf{1}_n$ denotes a column vector of n ones. We use $\|x\|_2$ to denote the ℓ_2 norm of a vector x, $\|A\|_F$ to denote the Frobenius norm of a matrix, and $\|A\|_2$ to denote the spectral norm of the matrix A, i.e., $\|A\|_2 = \sup_{\|x\|_2=1} \|Ax\|_2$. All functions of matrices are taken element-wise, with the exception of the matrix absolute value, $|A| = \sqrt{A^T A}$. When $n \to \infty$, we write $a_n = o(b_n)$ if $|a_n/b_n| \to 0$; $a_n = \omega(b_n)$ if $|a_n/b_n| \to \infty$; $a_n = O(b_n)$ if $|a_n/b_n| \leq C$ for some $C > 0$ and all n; and $a_n = \Theta(b_n)$ if $|a_n/b_n| \in (C_1, C_2)$ for some $C_2 > C_1 > 0$ and all n. Finally, we write $X_n = O_P(b_n)$ if for any $\alpha > 0$ there exists a constant C such that $\mathbf{P}(|X_n/b_n| > C) < \alpha$ for all large n; and $X_n = o_P(a_n)$ if $\mathbf{P}(|X_n/a_n| > \varepsilon) \to 0$ for all $\varepsilon > 0$. Further notation is defined in text as needed.

3.2 Network Model and Representation

The network model we consider is an extension of the popular stochastic block model (SBM) (Holland et al., 1983), which we recall in Definition 3.1.

Definition 3.1. *Conditioned on community membership $\theta \in [K]^n$, the undirected network $Y \sim \mathrm{SBM}(\theta, B)$ is an SBM with edge probabilities $B \in [0,1]^{K \times K}$ if:*

$$Y_{ij} \overset{ind}{\sim} \mathrm{Bernoulli}(B_{\theta_i \theta_j}), \quad i < j.$$

The extension we study is what we call the *additive-covariate stochastic block model* (ACSBM), which is also the model studied in Mele et al. (2022). In this setting, we observe a network with n nodes and K communities, along with a set of M discrete covariates. Links are formed independently, depending on community assignments, as in SBM, as well as on covariate similarity, allowing for explicit modeling of homophily based on the observed covariates. Homophily is therefore modeled in a manner similar to exponential random graph models (Goodreau et al., 2009), with latent structure modeled like SBM. The specific nature of the covariate influence is captured by a known link function g. We state a formal definition of this model in Definition 3.2.

Definition 3.2. *For nodes $i \in [n]$, let $\theta_i \in [K]$ denote latent community membership, and let $Z_i \in [L_1] \times \cdots \times [L_M]$ be a vector of M discrete, observed covariates. Let $Z = [Z_1 \mid \cdots \mid Z_n]^T$. Conditioned on θ and Z, the undirected network $Y \sim \text{ACSBM}(\theta, Z, B, \beta, g)$ is an additive-covariate SBM with covariate effects $\beta \in \mathbb{R}^M$ and known link function g if:*

$$Y_{ij} \overset{ind}{\sim} \text{Bernoulli}\left(g^{-1}\left(B_{\theta_i\theta_j} + \sum_{m=1}^{M} \beta_m \mathbb{I}(Z_{im} = Z_{jm})\right)\right), \quad i < j.$$

While the link function g could in principle be any strictly increasing function whose range includes $[0, 1]$, typical choices inspired by similar models include the logit link (e.g., Handcock et al., 2007; D. S. Choi et al., 2012; Vu et al., 2013; Roy et al., 2019; Ma et al., 2020), log link (e.g., Huang and Feng, 2018), probit link (e.g., Hoff, 2007), or identity link (Mu et al., 2022). Choice of link function should be informed by the nature in which covariates are believed to affect edge formation. Our theoretical analysis in Section 3.4 employs the log link, in which the effects of observed homophily are multiplicative on the probability of edge formation. Such effects are particularly reasonable to assume in sparse networks, easily interpreted (if estimated), and mimic the form of other popular models like the degree-corrected block model (Karrer and Newman, 2011).

The ACSBM's combination of independent edges and discrete attributes leads to an important representation result: the ACSBM, viewed one way as an extension of the SBM, may also be represented by a special case of the SBM. Specifically, Proposition 3.3 represents the ACSBM as an SBM by subdividing each latent community in ACSBM by the observed covariates, yielding an SBM over the resulting set of "subcommunities." This generalizes a similar result stated by Mele et al. (2022).

Proposition 3.3. *If $Y \sim \text{ACSBM}(\theta, Z, B, \beta, g)$, then Y is equal in distribution to a $(K\tilde{L})$-block SBM, namely $Y \overset{D}{=} \text{SBM}(\tilde{\theta}, \tilde{B})$ for:*

$$\begin{aligned}
\tilde{L} &= \prod_{m=1}^{M} L_m \\
\tilde{\theta} &= \tilde{L}(\theta - \mathbf{1}_n) + \sum_{m=1}^{M-1} \left[\prod_{m'=m+1}^{M} L_{m'}\right](Z_{*m} - \mathbf{1}_n) + Z_{*M}, \\
\tilde{B} &= g^{-1}(B \boxplus \beta_1 I_{L_1} \boxplus \cdots \boxplus \beta_P I_{L_M}),
\end{aligned}$$

where g^{-1} is taken element-wise, and $A_1 \boxplus A_2 = (A_1 \otimes \mathbf{1}_{d_2}\mathbf{1}_{d_2}^T) + (\mathbf{1}_{d_1}\mathbf{1}_{d_1}^T \otimes A_2)$ for matrices $A_1 \in \mathbb{R}^{d_1 \times d_1}, A_2 \in \mathbb{R}^{d_2 \times d_2}$.

Remark 3.4. *$\tilde{\theta}$ is formed from a bijection from $[K]\times[L_1]\times\cdots\times[L_M]$ to $[K\tilde{L}]$. In an abuse of notation, we will refer to this mapping later in the paper as $\tilde{\theta}(\cdot,\cdot)$ where for $k \in [K], z \in [L_1]\times\cdots\times[L_M]$, $\tilde{\theta}(k,z) = \tilde{L}(k-1) + \sum_{m=1}^{M-1}\left[\prod_{m'=m+1}^{M} L_{m'}\right](z_m - 1) + z_M$.*

The proof of Proposition 3.3 is constructive and is given in Section C.2 of Appendix C. This representation result leads to a natural idea: since any ACSBM network is equivalently represented as an SBM, perhaps familiar SBM-fitting methods can be adapted to fit the ACSBM.

3.2.1 Random Dot Product Graphs

Spectral clustering of SBMs has been studied extensively in the context of (generalized) random dot product graphs (RDPGs) (Athreya, Fishkind, et al., 2017; Rubin-Delanchy et al., 2022). The class of (g)RDPGs lends itself well to spectral estimation methods, and any binary, undirected, independent-edge network can be formulated as a generalized random dot product graph. In particular, it is well established that SBMs may be represented as gRDPGs (Rubin-Delanchy et al., 2022). Below we state the definition of a gRDPG and follow it with a representation result for ACSBM analogous to Proposition 3.3.

Definition 3.5. *The matrix $I_{pq} = \mathrm{diag}(I_p, -I_q)$ is the diagonal matrix whose first p diagonal entries are equal to $+1$ and whose remaining q diagonal entries are equal to -1. For $x, y \in \mathbb{R}^d$ and some nonnegative integers $p+q=d$, the indefinite inner product of x and y with signature (p,q) is given by $\langle x, y\rangle_{pq} = \langle x, I_{pq}y\rangle = x^T I_{pq} y$. The indefinite orthogonal group with signature (p,q) is given by the set of matrices $\mathbb{O}(p,q) = \{Q \in \mathbb{R}^{d\times d} : Q^T I_{pq} Q = I_{pq}\}$.*

Definition 3.6. *Let F_X be a distribution on $\mathbb{R}^d$. We say the undirected network $Y \sim \mathrm{gRDPG}(n, F_X)$ is a generalized random dot product graph with signature (p,q) if $X_1,\ldots,X_n \overset{iid}{\sim} F_X$, and $Y_{ij} \mid X_1,\ldots,X_n \overset{ind}{\sim} \mathrm{Bernoulli}(\langle X_i, X_j\rangle_{pq})$ for $i<j$. The variable X_i is referred to as the latent position of the i-th node.*

Remark 3.7. *When $q=0$, we say Y is a random dot product graph (without the "generalized" qualification) (Young and Scheinerman, 2007). In this case, $I_{pq} = I$, the indefinite inner product coincides with the usual dot product (i.e., $\langle x, y\rangle_{pq} = \langle x, y\rangle$), and $\mathbb{O}(p,q)$ coincides with the familiar group of $p\times p$ orthogonal matrices.*

Both RDPGs and gRDPGs suffer from inherent identifiability issues. In the case of RDPGs, for example, if any set of latent positions is altered by a common orthogonal transformation, the resulting RDPG has the same distribution, since $\langle x, y\rangle = \langle Qx, Qy\rangle$ for any orthogonal Q. In gRDPGs,

latent positions can only be identified up a common indefinite orthogonal transformation (Rubin-Delanchy et al., 2022). (For a comprehensive approach to the non-identifiability of gRDPGs, see Agterberg et al. (2020).) Unlike orthogonal transformations, indefinite orthogonal transformations do not preserve distances or angles, rendering them more burdensome to work with. In the following proposition, we choose our canonical latent positions based on a spectral decomposition, but we clarify that this choice of latent positions is not unique. The proof of Proposition 3.8, given in Section C.2 of Appendix C, follows as a corollary to Proposition 3.3, based on well known results in the gRDPG literature (e.g., Rubin-Delanchy et al., 2022, Section 2.1).

Proposition 3.8. *If $(\theta_i, Z_i) \in [K] \times [L_1] \times \cdots \times [L_M]$ are drawn i.i.d. from a distribution with p.m.f. $\mathbf{P}_{\theta,Z}$, and $Y \mid \theta, Z \sim \text{ACSBM}(\theta, Z, B, \beta, g)$ for $Z = [Z_1 \mid \cdots \mid Z_n]^T$ and some $\beta \in \mathbb{R}^M$, then Y is equal in distribution to a gRDPG, Y_{grdpg}, with latent positions sampled i.i.d. from a mixture of point masses. A canonical distribution for these latent positions is as follows. Let $\tilde{B}$ as in Proposition 3.3, and let $U_{\tilde{B}}\Lambda_{\tilde{B}}U_{\tilde{B}}^T$ be an eigendecomposition of $\tilde{B}$. Let $X_{\tilde{B}} = U_{\tilde{B}}|\Lambda_{\tilde{B}}|^{1/2}$, and let $X_{\tilde{B}}(k,z)$ denote the $\tilde{\theta}(k,z)$-th row of $X_{\tilde{B}}$. Let $F_{X_{\tilde{B}}}$ as follows:*

$$F_{X_{\tilde{B}}} = \sum_{\substack{k \in [K], \\ z \in [L_1] \times \cdots \times [L_M]}} \mathbf{P}_{\theta,Z}(\theta = k, Z = z)\delta_{X_{\tilde{B}}(k,z)}.$$

Letting q denote the number of negative entries in $\Lambda_{\tilde{B}}$, we have $Y_{grdpg} \sim \text{gRDPG}(n, F_{X_{\tilde{B}}})$ with signature $(p, q) = (K\tilde{L} - q, q)$.

3.3 Proposed Spectral Clustering Procedure

We propose a three-part algorithm (Algorithm 3) to estimate the latent community membership θ for an ACSBM network. Since an ACSBM with K latent communities is equivalently a $(K\tilde{L})$-block SBM per Proposition 3.3, we begin by trying to find the $K\tilde{L}$ "subcommunities" (i.e., $\tilde{\theta}$) of the SBM representation. Assuming we can recover the $K\tilde{L}$ subcommunities suitably, the primary remaining challenge is to merge these subcommunities into the original K desired communities (i.e., θ).

This fundamental idea is similar to that underlying Mele et al. (2022) and Mu et al. (2022), but we propose a new method for delineating the subcommunities and matching each subcommunity back to its original latent community, allowing for provably consistent results under mild assumptions. In both Mele et al. (2022) and Mu et al. (2022), the process of finding the $K\tilde{L}$ subcommunities relies only on the expected separation of their spectral embeddings in Euclidean space—a condition not met if any β_m is sufficiently small (or zero). Moreover, subsequent estimation of β in Mele et al.

(2022) and Mu et al. (2022) relies implicitly on an assumption that the diagonal entries in B are unique, so that an estimate of $\mathrm{diag}(\tilde{B})$ can be clustered into K sets of similar values corresponding to the K latent communities. In contrast, our method is robust to non-significant homophily effects and allows for any choice of B that satisfies a full-rank assumption.

Algorithm 3 Spectral Clustering of ACSBM

Input: adjacency matrix $Y \in \{0,1\}^{n \times n}$, discrete covariates $Z = [z_1 \mid \cdots \mid z_n]^T$, number of latent communities K, embedding dimension d

Output: estimated block membership $\hat{\theta} \in [K]^n$

 # Part 1: Recover the subcommunities $\tilde{\theta}$

Let $\hat{X}_Y := U|\Lambda|^{1/2}$, where $U\Lambda U^T$ is the truncated eigendecomposition of Y with dimension d

Let $L_1, \ldots, L_M := \max(Z_{*1}), \ldots, \max(Z_{*M})$

for z in $[L_1] \times \cdots \times [L_M]$ **do**

  Let $\mathcal{I}_z := \{i : z_i = z\}$

  Let $\hat{\theta}_z : \mathcal{I}_z \to [K]$ be a function returning cluster assignments over the rows of $\hat{X}_Y$ corresponding to the indices $\mathcal{I}_z$

end for

 # Part 2: Estimate $\tilde{B}$

for $1 \le k_1 \le k_2 \le K\tilde{L}$ **do**

  Let $D_{k_1,k_2} := \{(i,j) \in [n] \times [n] : i \neq j, \tilde{\theta}(\hat{\theta}_{z_i}(i), z_i) = k_1, \tilde{\theta}(\hat{\theta}_{z_j}(j), z_j) = k_2\}$

  Set $\hat{\tilde{B}}_{k_1,k_2} = \hat{\tilde{B}}_{k_2,k_1} := \sum_{(i,j) \in D_{k_1,k_2}} A_{ij} / \max\{1, |D_{k_1,k_2}|\}$

end for

 # Part 3: Reconcile θ using $z = \mathbf{1}_M$ as reference level

Let $\hat{X}_{\tilde{B}}(k,z)$ be the $\tilde{\theta}(k,z)$-th row of $V|\Psi|^{1/2}$, where $V\Psi V^T$ is an eigendecomposition of $\hat{\tilde{B}}$

for z in $[L_1] \times \cdots \times [L_M]$ **do**

  Let $\hat{\sigma}_z := \arg\min_{\sigma \in S_{[K]}} \sum_{k=1}^{K} \|\hat{X}_{\tilde{B}}(\sigma(k), z) - \hat{X}_{\tilde{B}}(k, \mathbf{1}_M)\|_2^2$

end for

return $\hat{\theta} = [\hat{\sigma}_{z_i}(\hat{\theta}_{z_i}(i))]_{i=1}^n$

Remark 3.9. *Algorithm 3 takes as input an embedding dimension d. This corresponds to the dimension of the latent positions in Proposition 3.8, which cannot exceed $K\tilde{L}$. In the absence of oracle*

knowledge, this maximum value appears to be a suitable choice for d.

Part 1 of the algorithm essentially seeks to recover $\tilde{\theta}$ of Proposition 3.3. To do so, we first find adjacency spectral embeddings for the full network. Then we consider each possible covariate configuration $z \in [L_1] \times \cdots \times [L_M]$ (of which there are $\tilde{L}$ total), and cluster the embeddings corresponding to nodes bearing this covariate configuration into K clusters. This yields a set of subcommunities that are each pure in their covariate distribution, since we know that $Z_i \neq Z_j \implies \tilde{\theta}_i \neq \tilde{\theta}_j$. A range of clustering methods (e.g., K-means) may be used here; existing theory suggests Gaussian mixture models may provide the best finite-sample performance (Athreya, Priebe, et al., 2016; Rubin-Delanchy et al., 2022). The computational complexity of Part 1 will depend on the specific clustering method employed.

Part 2 of the algorithm estimates $\tilde{B}$ so that we may estimate a latent position for each subcommunity. While the embeddings of Part 1 also serve as estimates of latent positions, these estimates are only consistent up to an indefinite orthogonal transformation, which would pose problems for the geometry of Part 3. In practical implementations, Part 2 can be performed in linear time, relative to the number of edges in the network.

Successful clustering in Part 1 of the algorithm implies that we are able to recover θ up to a separate permutation for any set of nodes with the same covariates. Part 3 of the algorithm seeks a common permutation for all nodes by attempting to reconcile each covariate configuration with a given reference level (canonically $z = \mathbf{1}_M$). This is achieved by finding the matching that minimizes the sum of squared distances between estimates of latent positions for each cluster. This optimization is a case of the assignment problem, which can be completed efficiently using the Hungarian algorithm (Edmonds and Karp, 1972). The computational complexity of Part 3 depends only on K and $\tilde{L}$. The analysis in Section 3.4 assumes these quantities are constant in n. If allowed to grow, however, we would only expect consistency of subcommunity recovery (i.e., Part 1) if $K\tilde{L}$ grew slower than $\sqrt{n}$, based on existing results in SBM recovery (e.g., Lei and Rinaldo, 2015). Under this assumption, the overall complexity of Part 3 of the algorithm is $o(n^{1.5})$ in time and $o(n)$ in space.

3.4 Consistency Results

Breaking Algorithm 3 into its three main parts, we first show that Part 1 consistently recovers $\tilde{\theta}$ from Proposition 3.3. Next, Part 2 yields a consistent estimate of $\tilde{B}$, given $\tilde{\theta}$ from Part 1. Finally, Part 3 yields a consistent estimate of θ, given $\tilde{\theta}$ from Part 1 and a suitable approximation of $\tilde{B}$ from Part 2. While detailed proofs of these results are left to Appendix C, we state the major theorems

and give an outline of the proof ideas here. To make things concrete, we consider the following setting.

Setting. Let M be a positive integer, and let $K, L_1, \ldots, L_M$ be integers greater than 1. Let $\mathbf{P}_{\theta Z}$ be a probability mass function on $[K] \times [L_1] \times \cdots \times [L_M]$. Let $\beta \in \mathbb{R}^M$ be a vector of covariate coefficients and $B_0 \in \mathbb{R}^{K \times K}$ be a symmetric matrix of latent block coefficients. To allow for sparsity, let $\alpha_n \in (0, 1]$ be a sequence controlling the expected degree of our networks. For each $n \geq 1$, we draw $\{(\theta_i, Z_i)\}_{i=1}^n \in ([K] \times [L_1] \times \cdots \times [L_M])^n$ from $(\mathbf{P}_{\theta Z})^n$. Letting $B = B_0 + \log(\alpha_n)\mathbf{1}_K\mathbf{1}_K^T$, we then draw $Y \mid \theta, Z \sim \text{ACSBM}(\theta, Z, B, \beta, \log)$.

As discussed in Section 3.2, under the log link, the effects of observed homophily are multiplicative on the probability of edge formation. When $\alpha_n \to 0$, this is essentially equivalent to the canonical logit link in the limit, since $\lim_{n\to\infty} \log^{-1}(b + \log(\alpha_n))/\text{logit}^{-1}(b + \log(\alpha_n)) = 1$ for any constant b. We note that in this setting, all edge probabilities scale by α_n, so the expected degree of each node is $\Theta(n\alpha_n)$. Although we drop the subscripts, the quantities $\tilde{B}$ and $X_{\tilde{B}}$ depend on n. When we desire constant quantities, we will use $\alpha_n^{-1}\tilde{B}$ and $\alpha_n^{-1/2}X_{\tilde{B}}$.

Assumptions. Our full set of results will require the following assumptions. Assumption ***(A1)*** is a relatively standard sparsity constraint in the SBM recovery literature. Assumption ***(A2)*** is equivalent to saying the latent SBM structure is full-rank, which is also common. Assumption ***(A3)*** requires that each latent community contains a node of each type with nonzero probability.

(A1) $\alpha_n = \omega(\log^{4c} n/n)$ for the universal constant c in Lemma 3.10.

(A2) $\exp(B_0)$ is full-rank.

(A3) $\mathbf{P}_{\theta Z}(\theta = k, Z = z) > 0$ for all $(k, z) \in [K] \times [L_1] \times \cdots \times [L_M]$.

We begin by recasting the ACSBM as a gRDPG with signature (p, q), as prescribed by Proposition 3.8. Let $\hat{X}_Y = U|\Lambda|^{1/2}$ (where $Y \approx U\Lambda U^T$) as in Algorithm 3, and let $\hat{X}_i$ denote the i-th row of $\hat{X}_Y$ (i.e., the spectral embedding for node i). Results from the gRDPG literature tell us that these spectral embeddings will be consistent estimates of the latent positions of the gRDPG, up to an unknown transformation from the indefinite orthogonal group $\mathbb{O}(p, q)$. This is stated in Lemma 3.10, which follows from Rubin-Delanchy et al. (2022, Theorem 3).

Lemma 3.10 (Rubin-Delanchy et al. (2022))**.** *Under assumptions **(A1)** and **(A3)**, there exists a universal constant $c > 1$ and a sequence of matrices $Q \in \mathbb{O}(p, q)$ such that:*

$$\max_{i \in [n]} \|Q\hat{X}_i - X_{\tilde{B}}(\theta_i, Z_i)\|_2 = O_P\left(\frac{\log^c n}{\sqrt{n}}\right).$$

The uniform consistency of Lemma 3.10 is the key to Part 1 of the algorithm. In particular, when we look at the spectral embeddings for nodes of a given covariate configuration $z \in [L_1] \times \cdots \times [L_M]$, this result yields perfect separation of the embeddings with high probability (Theorem 3.11).

Theorem 3.11. *Fix $z \in [L_1] \times \cdots \times [L_M]$. Let $\mathcal{I}_z = \{i : Z_i = z\}$. Assuming **(A1)** and **(A3)**, there exist K sequences of balls $\mathcal{B}_{1,z}, \ldots, \mathcal{B}_{K,z}$ such that $\hat{X}_i \in \mathcal{B}_{\theta_i,z}$ for all $i \in \mathcal{I}_z$ and $\mathcal{B}_{1,z}, \ldots, \mathcal{B}_{K,z}$ are disjoint with probability approaching 1.*

Theorem 3.11 is proven in Section C.2 of Appendix C and is sufficient to support exact recovery of $\tilde{\theta}$ with high probability under a variety of clustering algorithms, such as K-means (Lyzinski et al., 2014). However, while Lemma 3.10 states spherical concentration bounds, the clusters of embeddings generally are not spherical but are asymptotically normal, per the discussion in Rubin-Delanchy et al. (2022). For this reason, Gaussian mixture modeling is often preferred over K-means for finite-sample performance (Athreya, Priebe, et al., 2016; Rubin-Delanchy et al., 2022).

In view of Theorem 3.11, from here we assume knowledge of $\tilde{\theta}$ in order to demonstrate consistency in Parts 2 and 3 of the algorithm. Recall that Part 2 of the algorithm estimates $\tilde{B}$ from Proposition 3.3. While this estimate is not our end goal, we will use this reconstruction of $\tilde{B}$ to estimate the canonical latent positions $X_{\tilde{B}}$ from Proposition 3.8.

Theorem 3.12. *Let $\hat{\theta}_z : \mathcal{I}_z \to [K]$. Suppose for each $z \in [L_1] \times \cdots \times [L_M]$, there exists $\tau_z \in S_{[K]}$ such that $\hat{\theta}_z(i) = \tau_z(\theta_i)$ for all $i \in \mathcal{I}_z$. Assuming **(A1)**–**(A3)**, if $\hat{\tilde{B}}$ is constructed as in Algorithm 3, then there exists a sequence of $K\tilde{L} \times K\tilde{L}$ permutation matrices T such that:*

$$\alpha_n^{-1} \|\hat{\tilde{B}} - T\tilde{B}T^{-1}\|_F = o_P\left(\frac{1}{\sqrt{n \log^c n}}\right).$$

Theorem 3.12 follows from the fact that, conditioned on $\tilde{\theta}$, $\hat{\tilde{B}}$ is the maximum likelihood estimate for a matrix of SBM probabilities corresponding to the subcommunities of $\tilde{\theta}$ (up to relabeling). The bounds thus follow from a bit of algebraic manipulation of well-known results (Bickel et al., 2013; Tang et al., 2022), as outlined in Section C.2 of Appendix C. Finally, we move on to the main act: reconciling the $\tilde{L}$ per-covariate clusterings into a single clustering for all nodes.

Theorem 3.13. *Let $\hat{\theta}_z : \mathcal{I}_z \to [K]$ and $\hat{X}_{\tilde{B}}(k, z)$ as in Algorithm 3. Suppose for each $z \in [L_1] \times \cdots \times [L_M]$, there exists $\tau_z \in S_{[K]}$ such that $\hat{\theta}_z(i) = \tau_z(\theta_i)$ for all $i \in \mathcal{I}_z$. Let:*

$$\hat{\sigma}_z = \arg\min_{\sigma \in S_{[K]}} \sum_{k=1}^{K} \|\hat{X}_{\tilde{B}}(\sigma(k), z) - \hat{X}_{\tilde{B}}(k, \mathbf{1}_M)\|_2^2. \tag{3.1}$$

*Then, assuming **(A1)–(A3)**, $\hat{\sigma}_z(\hat{\theta}_z(i)) = \tau_{\mathbf{1}_M}(\theta_i)$ for all $i \in [n]$ with probability approaching 1.*

Theorem 3.13 involves an abundance of permutations. We assume that for each covariate configuration z, we have a function $\hat{\theta}_z(\cdot)$ that recovers the values of θ_i up to a permutation τ_z. We can find such functions with high probability from Part 1 of our algorithm. Then, for each z, we estimate a permutation $\hat{\sigma}_z$ in an attempt to "reverse" these permutations. Since the true permutations τ_z are unknowable, we cannot hope to invert τ_z exactly. Instead, we seek a permutation that satisfies $\hat{\sigma}_z \circ \tau_z = \tau_0$ for some common unidentifiable permutation $\tau_0 \in S_{[K]}$. By using $z = \mathbf{1}_M$ as our reference level, we end up recovering $\tau_0 = \tau_{\mathbf{1}_M}$.

The proof of Theorem 3.13 is broken into a number of intermediate results in the supplementary materials, of which we give an overview here. We first consider the task of solving an analog to the matching problem (3.1) using the true latent positions $X_{\tilde{B}}$ (Section C.2, Theorem C.15). A handful of linear algebra reduces this task to an optimization problem over a submatrix of $|\tilde{B}| = \sqrt{\tilde{B}\tilde{B}}$. Analysis of the entries of $|\tilde{B}|$ is tractable under the log link, as $\tilde{B}$ decomposes into a chain of Kronecker products (Section C.1, Facts C.10, C.12). Under assumption ***(A2)***, we find that the desired permutation is the unique optimum for the matching problem.

Having shown that the matching problem yields the desired result in the absence of estimation error, it remains to show that the estimation error vanishes asymptotically (Section C.2, Lemma C.16). The estimation error is bounded by a multiple of $\| \, |\hat{\tilde{B}}| - |T\tilde{B}T^{-1}| \, \|_F$, a bound for which follows from Theorem 3.12. This, indeed, shrinks to zero faster than the gap between the optimal and second-best matching. A formal proof of Theorem 3.13 tying these results together is given in Section C.2 of Appendix C.

In sum, Theorems 3.11–3.13 demonstrate that Algorithm 3 perfectly recovers ACSBM's latent community assignment variable, θ, in the limit. From this, asymptotically unbiased estimation of the remaining ACSBM parameters—including the marginal homophily effects, β—follows in a straightforward manner, using standard GLM-fitting approaches with $\hat{\theta}$ as a plug-in estimator for θ.

3.5 Simulations

We evaluate the empirical performance of our method on a variety of sequences of ACSBM networks. First, we consider two sequences of sparse networks ($\alpha_n = n^{-0.8}$) with $K = 2$ latent communities and $M = 2$ covariates drawn i.i.d. as Bernoulli(0.5). The link function is chosen to be $g = \log$. In the first setting, we use a "regular" structure for the latent SBM, $B_0 = 1.5\,\mathbf{1}_2\mathbf{1}_2^T - I_2$. In the second, we consider something more "irregular," with $B_0 = \mathbf{1}_2\mathbf{1}_2^T + \mathrm{diag}(1, -0.2)$. In both

cases, covariate effects are $\beta_1 = 1, \beta_2 = -0.5$. For each of ten values of n ranging from $n = 125$ to $n = 128000$, we generate 100 networks, then apply Algorithm 3, using Gaussian mixture modeling as our clustering method for Part 1. We calculate a misclassification rate (up to relabeling) as $\min_{\sigma \in S_{[K]}} n^{-1} \sum_{i=1}^{n} \mathbb{I}(\sigma(\hat{\theta}_i) \neq \theta_i)$. The median misclassication rate is plotted in the left panel of Figure 3.1, with error bands denoting the interquartile range (IQR). The dashed line represents the worst possible misclassification rate of one half. As we might hope, as n increases, misclassification falls toward zero.

The second set of simulations evaluates the performance of the algorithm on dense networks ($\alpha_n = 1$), with four settings corresponding to different choices of link function: identity, log, logit, and probit. In each case, we model the underlying latent structure as an SBM with $K = 3$ communities and model $M = 2$ binary covariates, drawn i.i.d. as Bernoulli(0.5). For the identity link, we choose $B = 0.2\,\mathbf{1}_3\mathbf{1}_3^T - 0.1 I_3, \beta_1 = 0.05, \beta_2 = -0.05$. For the remaining links, we use $B = -\mathbf{1}_3\mathbf{1}_3^T - 0.5 I_3, \beta_1 = -0.7, \beta_2 = 0.1$. For seven values of n ranging from $n = 125$ to $n = 8000$, we simulate 100 networks and apply the same clustering methodology as in the previous set of simulations. The results are plotted in the right panel of Figure 3.1. Here we see consistency for a greater variety of link functions than was proven in Section 3.4, suggesting even greater generality for our proposed method. In our dense simulations, we achieve perfect clustering in the overwhelming majority of cases when $n \geq 2000$.

We caution against direct comparisons of the simulation settings presented here. For example, in the dense network simulations, one may notice that convergence appears fastest for the log link and slowest for the logit link, but each setting is different in ways that complicate comparisons. While these two settings share the same parameters, the difference in link function subtly affects the relations between entries in $\tilde{B}$ and leads to a network of lower density for the logit link, since $\text{logit}^{-1}(x) < \log^{-1}(x)$ for any $x \in \mathbb{R}$.

These simulations were conducted on a high performance cluster, but each individual network was simulated and fit using a single CPU core (2.2 GHz Intel Xeon). The most demanding simulation setting was the sparse, regular setting at $n = 128000$ nodes, where each network had about 6.2 million edges on average. The average running time for this setting using our Python-based algorithm was 4.35 minutes per network, of which 4.25 minutes were spent in Part 1 of Algorithm 3.

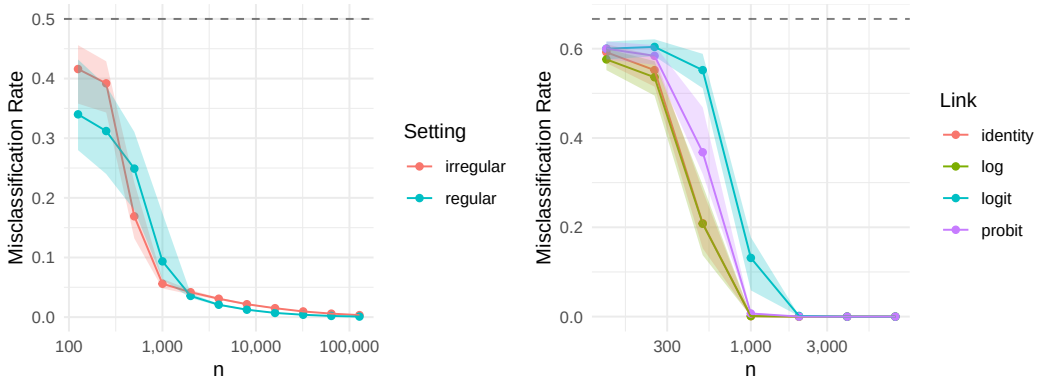


Figure 3.1. Median proportion (and IQR) of misclassified nodes on repeated simulations of ACSBM models. Left: Sparse settings with $K = 2, M = 2, g = \log, \alpha_n = n^{-0.8}$. Right: Dense settings with $K = 3, M = 2$, various g, $\alpha_n = 1$. Dashed line represents worst possible misclassification $(1 - 1/K)$. Specific parameters given in text.

3.6 Application to Harvard Facebook Data

We illustrate the real-world value of our method on a network of Facebook friendships between Harvard University students. The network we consider is a subgraph of a network originally published in Traud et al. (2012), consisting of Facebook friendships between 15,126 individuals from Harvard. Of the 5,970 profiles known to belong to students that declare their gender and a class year between 2006 and 2009, we restrict our attention to the largest connected component of $n = 5{,}917$ nodes and 629,864 edges. Letting $Z_{*1} \in [2]^n$ denote the genders of the students and $Z_{*2} \in [4]^n$ denote the four class years (suitably recoded), we consider a model with $K = 2$ latent communities.

We apply Algorithm 3, using as our clustering method a version of K-means that operates over the rows of $\hat{X}_Y$ normalized to have unit length. This is inspired by a popular method employed in fitting degree-corrected block models (Karrer and Newman, 2011; Lei and Rinaldo, 2015) and allows for varying node degrees within subcommunities. We denote the resulting latent communities $\hat{\theta}_{\text{covariate}}$. The first of these groups contains 3,772 nodes, and the second contains 2,145. The estimated $\tilde{B}$ matrix is depicted in Figure 3.2, where the latent structure is represented by the four main quadrants of the matrix, and the homophily patterns between gender and class year are captured by the repetitive structures within each of these quadrants.

We estimate the coefficients of the ACSBM model by fitting a logistic regression model using $\hat{\theta}_{\text{covariate}}$ as a plug-in estimator for θ and the empirical edge counts from $\hat{\tilde{B}}$. The resulting estimated

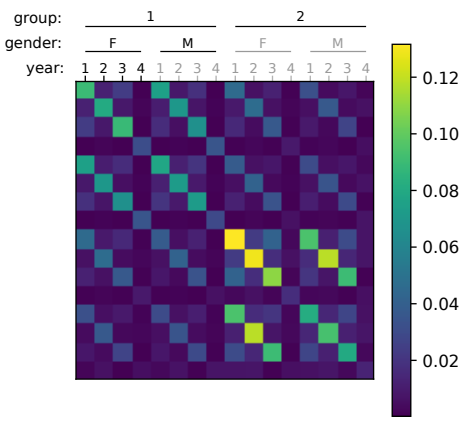


Figure 3.2. Estimated $\tilde{B}$ matrix applying Algorithm 3 to a network of Harvard students, accounting for observed covariates of class year and gender

model is:

$$\log\left(\frac{\hat{\pi}_{ij}}{1-\hat{\pi}_{ij}}\right) = \begin{bmatrix} -5.077 & -5.520 \\ -5.520 & -4.290 \end{bmatrix}_{\theta_i\theta_j} + 0.102\,\mathbb{I}(\text{gender}_i = \text{gender}_j) + 2.113\,\mathbb{I}(\text{year}_i = \text{year}_j),$$

where π_{ij} denotes the probability of an edge occurring between nodes i and j. We interpret this as follows: after accounting for the latent structure we have discovered, sharing the same gender (or class year) is associated with odds of forming a friendship that are $e^{0.102} \approx 1.107$ times (or $e^{2.113} \approx 8.273$ times, respectively) higher, when holding the other variables constant for the pair. The remaining coefficients in the 2×2 matrix represent intercept terms that vary depending on the clusters to which a pair of nodes belongs. We note that the overall differences between these coefficients lie somewhere between the effect size of gender and that of class year. All of this matches our intuition from Figure 3.2, where differences between same-sex and opposite-sex friendship patterns are slight, differences by class year are stark, and the latent structure contributes modestly.

The question of what the latent structure represents has no easy answer, since it corresponds to an unobserved feature of the network. We note, however, a curious correlation between this latent feature and class year. For students with class years of 2006, 2007, 2008, and 2009, the proportions assigned to latent group 1 are, respectively, 53.5%, 50.5%, 60.8%, and 90.1%. Noting that incoming freshmen at Harvard are traditionally assigned to designated housing and offered unique social programs, we conjecture that the latent feature captured by our method corresponds

to residential and social patterns that correlate with the freshman experience, while going beyond what is captured by class year alone.

Comparison with vanilla spectral clustering. Comparing $\hat{\theta}_{\text{covariate}}$ to the communities discovered through a spectral clustering process that ignores the covariates highlights the value of our proposed method. Specifically, we compare our method against that described in Lei and Rinaldo (2015), but substituting the more commonly used K-means in place of K-medians. The resulting communities we denote $\hat{\theta}_{\text{vanilla}}$. As expected from the results of previous studies (e.g., Y. Chen et al., 2018), the "latent" communities discovered in $\hat{\theta}_{\text{vanilla}}$ largely recover information already contained in our observed covariates: with $K = 2$, $\hat{\theta}_{\text{vanilla}}$ almost perfectly predicts whether the student is a freshman ($\text{year}_i = 4$), with agreement between the latent community and freshman status on 98.5% of nodes. Expanding the search to $K = 4$ recovers clusters that agree with class year on 95.3% of nodes (up to a permutation of labels). At $K = 8$, we begin to discover a latent structure: the recovered clusters $\hat{\theta}_{\text{vanilla}} \in [8]^n$ agree with a suitably recoded combination of class year and our $\hat{\theta}_{\text{covariate}}$ on 71.8% of nodes. However, recovering a hierarchical structure from these labels remains an elusive task. This is precisely the challenge addressed by our algorithm, which first identifies a flattened set of "subcommunities" in Algorithm 3, Part 1, before working backward to recover the full hierarchical structure in Part 3.

3.7 Discussion

The task of separating latent from observed structure in networks is critical to a variety of network inference tasks. The method we have proposed is, to our knowledge, the first to offer a rigorous guarantee of consistency of latent structure recovery using spectral clustering in the setting where edge formation is dependent on both observed and latent factors. Our proposed method is computationally efficient and theoretically appealing, using distance in latent space as a means of reconnecting a network partitioned by observed covariates.

We would like to note the limitations of our current work and highlight opportunities for future research. First and foremost, the combinatorial nature of the algorithm restricts its use to discrete covariates. Moreover, since Part 3 of the algorithm estimates permutations over network partitions, any error in permutation selection is likely to introduce considerable error in the final clustering of nodes. A post-processing step akin to spectral clustering with adjustment (SCWA) of Huang and Feng (2018) may be useful to avoid finite-sample permutation errors but has yet to be explored. Finally, while we consider only a fixed number of latent communities and covariates, it would be useful to extend our analysis to the case where these quantities grow. Based on

existing results for SBM recovery (e.g., Lei and Rinaldo, 2015), we anticipate the total number of subcommunities of Proposition 3.3 is limited to $K\tilde{L} = o(\sqrt{n})$. It would be interesting, but well outside the scope of this paper, to extend these ideas to a continuous setting, which may alleviate these limitations.

We believe that our proposed method offers promise beyond what has been proven so far. As an example, the simulations of Section 3.5 suggest consistency for a wide range of link functions that remains to be rigorously proven. Additionally, the analysis of Harvard students' Facebook friendships in Section 3.6 employed a degree-corrected post-processing step for the spectral embeddings, in a manner commonly employed with the degree-corrected block model (DCBM) of Karrer and Newman (2011). While the consistency of our method has not been rigorously proven in conjunction with degree correction, the empirical results look promising, as the method successfully identified a latent structure distinct from the covariates considered. Moreover, there is theoretical intuition behind this method. The matching problem of Algorithm 3, Part 3 may be recast as an optimization over the *angles* between subcommunities in latent space, while the latent positions in a degree-corrected analog of the ACSBM would be expected to fall along distinct rays corresponding to subcommunities. Such a theoretical extension for degree correction would greatly expand the practicality of the model we consider, allowing for nodes to exhibit greater variation in node degree, as commonly seen in observed networks, while retaining the simplicity and flexibility of the underlying latent block model structure.

Concluding Remarks

In the search for solutions to three distinct and specific problems, this dissertation uncovers tools to solve additional problems and raises further questions and areas for exploration.

Generalizations and Extensions

Chapter 1 relies on randomized response to transform sensitive binary data sketches to a private and mergeable equivalent, with a specific focus on extending the classical sketch of Flajolet and Martin (1985). However, the proposed methods of constructing and merging sketches are based on mechanisms designed at a very primitive level: bitwise operations on binary data. As a result, the techniques are not limited to this sketch—they apply similarly to other classical data structures such as Bloom filters (Broder and Mitzenmacher, 2004) or the more recently developed Liquid Legions sketch (Kreuter et al., 2020). Moreover, these techniques extend to questions not directly addressed in the chapter, such as how to merge two private data sketches of different dimension. There is reason to believe these fundamental techniques could be applied to a wealth of questions beyond the count-distinct problem as well.

Randomized response is the privacy mechanism of choice again in Chapter 2. Although the task at hand is specific—community detection on a certain class of models—the techniques employed are once again broad. The main results are based on general concentration results developed for the spectral embeddings of private networks, which may potentially be used for other forms of analysis. For example, these techniques have since been employed to analyze the topological structure of networks, which may be seen as a sort of generalization of the community detection problem (Vishwanath and Hehir, 2022).

The final chapter (Chapter 3) ends with a discussion of the limitations of the existing theory, noting that in practice, the proposed method works well in broader settings than proven. This conjectured generalization, a gap between the theoretical and the practical, is one of several areas left as future work.

Future Work and Limitations

In Chapter 1, where we performed bitwise operations under randomized response, we see the surprising result that the method resulting in the smallest estimation error is the one that relies on *randomized* operations. It remains to rigorously show what portion of the final estimation error is due to this randomization or to detail the settings in which this error could be meaningfully improved. For example, it's possible to obtain the expectation of the composite likelihood function (with respect to the randomness of the merge) by recording the expectation of a merge operation instead of re-randomizing the bits. However, the theoretical properties and practical tradeoffs (e.g., in storage space) of this approach have not been settled.

In the community detection problem of Chapter 2, we perform randomized response over the edges of a network to produce a private synthetic network. For a fixed privacy budget, this turns sparse networks into dense networks, with the majority of edges resulting from noise rather than the original network relationships. The generality of this privacy mechanism offers flexibility in implementation and allows for a variety of analyses to be run over the same synthetic network, but this comes at the price of limited support for sparse networks. After the initial publication of our results, Mohamed et al. (2022) published results on the recoverability limits of stochastic block models (SBMs)—a more specific subset of the models we consider—under edge differential privacy. Of the methods they study, the randomized response approach is the only approach that runs in polynomial time, but it suffers from the same sparsity limitations that we observed. On the other hand, their results identify alternative mechanisms capable of sparse network recovery at the cost of higher computational complexity. Even more recently, H. Chen et al. (2023) published a polynomial-time algorithm for sparse SBM recovery with a slightly weaker (ε, δ)–edge differential privacy guarantee. The existence of an efficient approach to sparse community detection under pure differential privacy remains an open question.

Finally, the application of spectral clustering to the setting of Chapter 3 falls into a particularly nascent area of the literature, with the earliest work applying spectral methods to this sort of problem appearing only very recently (Mele et al., 2022). Beyond the open questions posed in this chapter—generalization of the theoretical results to a wider range of models—it remains to be seen just how much potential these methods have to solve related statistical inference problems and how they will stand against their competition in practice.

Appendix A

Supporting Proofs for Chapter 1

In the proofs to follow, we use the following notation. $\mathbf{1}_k$ is the column vector of k ones, and I_k is the $k \times k$ identity matrix (with subscripts omitted when dimensions are clear). We denote by e_i the i-th elementary basis vector, i.e., the vector whose entries are all 0, except at position i, where the entry is 1. We use $\langle \cdot, \cdot \rangle$ to denote the inner product between two vectors.

A.1 Preliminaries

We state below two facts that will be useful in proving the main results.

Fact A.1. *Let $X \sim \text{Bernoulli}(p), Y \sim \text{Bernoulli}(q)$ be independent. Then:*

1. $X \wedge Y \sim \text{Bernoulli}(pq)$.
2. $X \vee Y \sim \text{Bernoulli}(p(1-q) + q(1-p) + pq)$.
3. $X \veebar Y \sim \text{Bernoulli}(p(1-q)+q(1-p))$. *Moreover, if $p = \frac{1}{2}$ or $q = \frac{1}{2}$, then $X \veebar Y \sim \text{Bernoulli}(\frac{1}{2})$.*

Proof.

$$\begin{aligned}
\mathbf{P}(X \wedge Y = 1) &= \mathbf{P}(X = 1, Y = 1) \\
&= pq, \\
\mathbf{P}(X \veebar Y = 1) &= \mathbf{P}(X = 1, Y = 0) + \mathbf{P}(X = 0, Y = 1) \\
&= p(1-q) + q(1-p), \\
\mathbf{P}(X \vee Y = 1) &= \mathbf{P}(X \wedge Y = 1) + \mathbf{P}(X \veebar Y = 1).
\end{aligned}$$

□

Fact A.2. *Let $A_{m_A \times n_A}, B_{m_B \times n_B}$ be matrices satisfying $A\mathbf{1} = \mathbf{1}$ and $B\mathbf{1} = \mathbf{1}$. Then $(AB)\mathbf{1} = \mathbf{1}$ and $(A \otimes B)\mathbf{1} = \mathbf{1}$. Additionally, if A^{-1} exists, then $A^{-1}\mathbf{1} = \mathbf{1}$.*

Proof. Since $A\mathbf{1} = \mathbf{1}$ and $B\mathbf{1} = \mathbf{1}$, we must have $(AB)\mathbf{1} = A(B\mathbf{1}) = A\mathbf{1} = \mathbf{1}$.

Next, write $\mathbf{1}_{m_A m_B} = \mathbf{1}_{m_A} \otimes \mathbf{1}_{m_B}$. Then

$$\begin{aligned}(A \otimes B)\mathbf{1}_{m_A m_B} &= (A \otimes B)(\mathbf{1}_{m_A} \otimes \mathbf{1}_{m_B}) \\ &= (A\mathbf{1}_{m_A}) \otimes (B\mathbf{1}_{m_B}) \\ &= \mathbf{1}_{m_A} \otimes \mathbf{1}_{m_B} \\ &= \mathbf{1}_{m_A m_B}.\end{aligned}$$

Finally, in the case when A is invertible, $A^{-1}A = I$, and so we must have

$$A^{-1}\mathbf{1} = A^{-1}A\mathbf{1} = I\mathbf{1} = \mathbf{1}.$$

□

A.2 Proofs of Results in Section 1.3

Proof of Theorem 1.2. The proof follows along the same lines as Erlingsson et al. (2014, Theorem 2). We first prove that $\mathcal{F}_{p,q}$ is ε-DP if and only if $p \in (0, 1)$ and $\max\{p/q, (1-q)/(1-p)\} \leq e^{\varepsilon}$. For this to hold, we must satisfy

$$\mathbf{P}(\mathcal{F}_{p,q}(x) = y) \leq e^{\varepsilon}\, \mathbf{P}(\mathcal{F}_{p,q}(1-x) = y)$$

for all $x, y \in \{0, 1\}$. If $p = q$, $\mathcal{F}_{p,p}$ is a mechanism that ignores its inputs and outputs a randomly chosen bit value and so this holds trivially. Assuming $p \neq q$, this holds if

$$\mathbf{P}(\mathcal{F}_{p,q}(1-x) = y) > 0 \quad \text{and} \quad \frac{\mathbf{P}(\mathcal{F}_{p,q}(x) = y)}{\mathbf{P}(\mathcal{F}_{p,q}(1-x) = y)} \leq e^{\varepsilon}.$$

The first condition is equivalent to the requirement that $p, q \in (0, 1)$, and the second condition is equivalent to $\max\{p/q, (1-q)/(1-p)\} \leq e^{\varepsilon}$.

For $\mathcal{M}_{p,q}$, note that the entries of $\mathcal{M}_{p,q}$ are independent by construction, so for $x, x' \in \{0, 1\}^d$ differing only on one bit $x_j = 1 - x'_j$ and some $y \in \{0, 1\}^d$, we have:

$$\frac{\mathbf{P}(\mathcal{M}_{p,q}(x) = y)}{\mathbf{P}(\mathcal{M}_{p,q}(x') = y)} = \frac{\prod_i \mathbf{P}(\mathcal{F}_{p,q}(x_i) = y_i)}{\prod_i \mathbf{P}(\mathcal{F}_{p,q}(x'_i) = y_i)} = \frac{\mathbf{P}(\mathcal{F}_{p,q}(x_j) = y_j)}{\mathbf{P}(\mathcal{F}_{p,q}(1-x_j) = y_j)}.$$

As shown above, this quantity is bounded by e^{ε} under the stated conditions on p and q. □

Theorem 1.2 is used below to demonstrate that both $\mathcal{M}_\varepsilon^{\text{xor}}$ and $\mathcal{M}_\varepsilon^{\text{sym}}$ satisfy ε-DP via Corollaries 1.5 and 1.9.

A.3 Proofs of Results in Section 1.4.1

Proof of Theorem 1.3. We begin with some simple necessary conditions for (1)–(3) to hold. From Theorem 1.2, we know that p_1, p_2, q_1, q_2 must lie in $(0, 1)$. Another necessary condition is that $p_1 \neq q_1$ and $p_2 \neq q_2$, as otherwise (3) is violated. For this reason, we can assume $p_1 \neq q_1$ and $p_2 \neq q_2$.

Modulo negation, there exist four symmetric operations $\{0,1\}^2 \to \{0,1\}$: and ($\wedge$), or ($\vee$), xor ($\veebar$), and the trivial operator that maps all inputs to 0. We will now rule out the operators other than $\veebar$.

(Trivial Operator) Let $\circ$ denote the operator $x \circ y = 0$. Then $f_1(x) \circ f_2(y) = 0$ for any x, y. If (2) holds, then $f_3(0) = f_3(1) = 0$, violating (3).

(And) Let $\circ = \wedge$, and assume (2) holds. Then we must have:

$$f_1(0) \wedge f_2(1) \overset{D}{=} f_3(1) \overset{D}{=} f_1(1) \wedge f_2(1).$$

By Fact A.1, this implies that:

$$q_1 p_2 = p_1 p_2.$$

Since we assumed $p_2 \neq 0$, this implies that $p_1 = q_1$, in contradiction to our assumption.

(Or) Let $\circ = \vee$, and assume (2) holds. Then we must have:

$$f_1(0) \vee f_2(1) \overset{D}{=} f_3(1) \overset{D}{=} f_1(1) \vee f_2(1).$$

By Fact A.1, this implies that:

$$p_1(1 - q_2) + q_2(1 - p_1) + p_1 q_2 = p_1(1 - p_2) + p_2(1 - p_1) + p_1 p_2.$$

After rearranging terms, we obtain:

$$0 = (1 - p_1)(p_2 - q_2).$$

Since we assumed $p_1 \neq 1$, this implies that $p_2 = q_2$, in contradiction to our assumption.

(Xor) Now we will show that when $\circ = \veebar$, we must have $p_1 = p_2 = \frac{1}{2}$. Assuming condition (2)

holds, we have:

$$f_1(0) \veebar f_2(1) \overset{D}{=} f_3(1) \overset{D}{=} f_1(1) \veebar f_2(1),$$

which implies (by Fact A.1):

$$q_1(1-p_2) + p_2(1-q_1) = p_1(1-p_2) + p_2(1-p_1).$$

Rearranging terms yields:

$$p_2(p_1 - q_1) = (1-p_2)(p_1 - q_1).$$

Since we assumed $p_1 \neq q_1$, we must have $p_2 = \frac{1}{2}$. A similar argument shows $p_1 = \frac{1}{2}$. □

Proof of Lemma 1.5. We need to show that $p/q \leq e^\varepsilon$ and $(1-q)/(1-p) \leq e^\varepsilon$ for $p = 1/2$ and $q = 1/(2e^\varepsilon)$. Clearly, $p/q = e^\varepsilon$, satisfying the first component. Next, consider the expression

$$\begin{aligned} e^\varepsilon - \frac{1-q}{1-p} &= e^\varepsilon - \frac{1 - \frac{1}{2}e^{-\varepsilon}}{\frac{1}{2}} \\ &= e^\varepsilon - (2 - e^{-\varepsilon}) \\ &= e^\varepsilon + e^{-\varepsilon} - 2 \\ &= (e^{\varepsilon/2} - e^{-\varepsilon/2})^2 \\ &\geq 0. \end{aligned}$$

Therefore, $(1-q)/(1-p) \leq e^\varepsilon$ as required. □

Proof of Theorem 1.6. Since the entries of $\mathcal{M}^{\text{xor}}(\cdot)$ are independent, it suffices to show this holds for $\mathcal{M}^{\text{xor}}$ applied to arbitrary $x_i, y_i \in \{0,1\}$. Observe that if $x_i = 1$ or $y_i = 1$, then $x_i \vee y_i = 1$, and so $\mathcal{M}^{\text{xor}}_{\varepsilon^*}(x_i \vee y_i) \sim \text{Bernoulli}(\frac{1}{2})$. On the other hand, since we know $x_i = 1$ or $y_i = 1$, then $\mathcal{M}^{\text{xor}}_{\varepsilon_1}(x_i) \sim \text{Bernoulli}(\frac{1}{2})$ or $\mathcal{M}^{\text{xor}}_{\varepsilon_2}(y_i) \sim \text{Bernoulli}(\frac{1}{2})$, and so by Fact A.1, $\mathcal{M}^{\text{xor}}_{\varepsilon_1}(x_i) \veebar \mathcal{M}^{\text{xor}}_{\varepsilon_2}(y_i) \sim \text{Bernoulli}(\frac{1}{2})$.

Thus all that remains to show is that $\mathcal{M}^{\text{xor}}_{\varepsilon^*}(x \vee y) \overset{D}{=} \mathcal{M}^{\text{xor}}_{\varepsilon_1}(x_i) \veebar \mathcal{M}^{\text{xor}}_{\varepsilon_2}(y_i)$ when $x_i = y_i = 0$. In this case, $\mathcal{M}^{\text{xor}}_{\varepsilon_1}(x_i) \sim \text{Bernoulli}(\frac{1}{2}e^{-\varepsilon_1})$, and $\mathcal{M}^{\text{xor}}_{\varepsilon_2}(y_i) \sim \text{Bernoulli}(\frac{1}{2}e^{-\varepsilon_2})$. By Fact A.1, we have that:

$$\mathcal{M}^{\text{xor}}_{\varepsilon_1}(x_i) \veebar \mathcal{M}^{\text{xor}}_{\varepsilon_2}(y_i) \sim \text{Bernoulli}(q^*),$$

where

$$\begin{aligned} q^* &= \frac{1}{2}e^{-\varepsilon_1}\left(1-\frac{1}{2}e^{-\varepsilon_2}\right)+\frac{1}{2}e^{-\varepsilon_2}\left(1-\frac{1}{2}e^{-\varepsilon_1}\right) \\ &= \frac{1}{2}\left(e^{-\varepsilon_1}+e^{-\varepsilon_2}-e^{-(\varepsilon_1+\varepsilon_2)}\right) \\ &= \frac{1}{2}\exp\left(-\left[\underbrace{-\log\left(e^{-\varepsilon_1}+e^{-\varepsilon_2}-e^{-(\varepsilon_1+\varepsilon_2)}\right)}_{\varepsilon^*}\right]\right) \\ &= \frac{1}{2}e^{-\varepsilon^*}. \end{aligned}$$

Finally, since $x_i \vee y_i = 0$, we have that $\mathcal{M}^{\text{xor}}_{\varepsilon^*}(x_i \vee y_i) \sim \text{Bernoulli}(\frac{1}{2}e^{-\varepsilon^*})$. □

A.4 Proofs of Results in Section 1.4.2

Proof of Lemma 1.9. Since $p = e^\varepsilon/(e^\varepsilon+1)$ and $q = 1-p = 1/(e^\varepsilon+1)$, we have $p/q = (1-q)/(1-p) = e^\varepsilon$, and so the result follows immediately from Theorem 1.2. □

We now state a more general form of Theorem 1.10 for proof. Where Theorem 1.10 gives a randomized merge for 2 bits, which may be invoked repeatedly to merge $k > 2$ bits, Theorem A.3 considers a simultaneous merge of $k \geq 2$ bits. Beyond simply serving to prove the original pairwise theorem, this generalization shows that nothing is gained in ε^* (i.e., the noise level of the final sketch) by simultaneously merging k bits vs. performing repeated pairwise merges.

Theorem A.3. *Fix an integer $k \geq 2$. For $i \in [k]$, assume $\varepsilon_i > 0$. Let*

$$q(\varepsilon) = \frac{1}{e^\varepsilon+1},\ K_\varepsilon = \begin{bmatrix} 1-q(\varepsilon) & q(\varepsilon) \\ q(\varepsilon) & 1-q(\varepsilon) \end{bmatrix},$$

$$\varepsilon^* = -\log\left(1-\prod_{i=1}^{k}(1-e^{-\varepsilon_i})\right),\ q^* = q(\varepsilon^*),$$

and let $v^* \in \mathbb{R}^{2^k}$ be the vector whose first entry is q^* with all other entries $1-q^*$. Let

$$t = (t_{\ldots 01}, t_{\ldots 10}, t_{\ldots 11}, \ldots)^T = (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})v^*,$$

$$g(x_1, \ldots, x_k) \sim \text{Bernoulli}(t_{x_1 \ldots x_k}).$$

Then $g(\mathcal{M}^{\text{sym}}_{\varepsilon_1}(x_1), \ldots, \mathcal{M}^{\text{sym}}_{\varepsilon_k}(x_k)) \overset{D}{=} \mathcal{M}^{\text{sym}}_{\varepsilon^*}(x_1 \vee \cdots \vee x_k)$.

Proof of Theorem A.3. Since all operations are performed bitwise and independently, assume without loss of generality that the bit vectors x_i are scalar, i.e., $x_1, \ldots, x_k \in \{0, 1\}$.

The fundamental idea is to model the chain of operations performed on the bits x_i as a Markov chain. This involves three different types of transition probability matrices. The matrices K_ε defined in the theorem statement map the state space of a single bit in $\{0, 1\}$ to another bit in $\{0, 1\}$ via the application of $\mathcal{M}_\varepsilon^{\text{sym}}$. Next, we define K^{or} to be the $2^k \times 2$ matrix mapping k bits in $\{0, 1\}^k$ to a single bit in $\{0, 1\}$ via an *or* operation. Finally, we define K^{merge} to be the $2^k \times 2$ matrix corresponding to our desired merge operation, which maps $\{0, 1\}^k$ to $\{0, 1\}$.

Since the bit-flipping operations of K_ε are performed independently, the matrix $K_{\varepsilon_1} \otimes \cdots \otimes K_{\varepsilon_k}$ represents the $2^k \times 2^k$ matrix jointly mapping the state space of the original bits $\{x_i\}_{i=1}^k$ to $\{\mathcal{M}_{\varepsilon_i}^{\text{sym}}(x_i)\}_{i=1}^k$. Thus we wish to solve:

$$(K_{\varepsilon_1} \otimes \cdots \otimes K_{\varepsilon_k}) K_{\varepsilon'}^{\text{merge}} = K^{\text{or}} K_{\varepsilon'}, \tag{A.1}$$

where ε' is a free parameter that we will fix, and $K_{\varepsilon'}^{\text{merge}}$ is the unknown quantity. We proceed by solving the matrix equation above, finding the maximum ε' for which $K_{\varepsilon'}^{\text{merge}}$ represents a valid transition probability matrix.

Let $q_i = q(\varepsilon_i), q' = q(\varepsilon')$. We note that K_{ε_i} is invertible and write $K_{\varepsilon_i}^{-1}$ as follows:

$$K_{\varepsilon_i}^{-1} = \frac{1}{1 - 2q_i} \begin{bmatrix} 1 - q_i & -q_i \\ -q_i & 1 - q_i \end{bmatrix}.$$

So we may solve our matrix equation (A.1) by left-multiplication of $(K_{\varepsilon_1} \otimes \cdots \otimes K_{\varepsilon_k})^{-1}$:

$$K_{\varepsilon'}^{\text{merge}} = (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1}) K^{\text{or}} K_{\varepsilon'}.$$

The first column of $K^{\text{or}} K_{\varepsilon'}$ is equal to

$$w' = (1 - q', q', \ldots, q')^T.$$

It follows from Fact A.2 that $K_{\varepsilon'}^{\text{merge}} \mathbf{1} = \mathbf{1}$ and therefore $K_{\varepsilon'}^{\text{merge}}$ is stochastic if and only if:

$$u' = (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})\, w' \in [0, 1]^{2^k}.$$

We may write u_i', the i-th entry of u', as the inner product of w' with the i-th row of $(K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})$.

We denote by r_i this row vector. Writing $w' = q'\mathbf{1} + (1-2q')e_1$, u'_i may be written:

$$\begin{aligned} u'_i &= \langle r_i,\ w' \rangle \\ &= \langle r_i,\ q'\mathbf{1} + (1-2q')e_1 \rangle \\ &= q'\langle r_i,\ \mathbf{1} \rangle + (1-2q')\langle r_i, e_1 \rangle \\ &= q' + (1-2q')(K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})_{i1}, \end{aligned}$$

where the final equality comes from the fact that $\langle r_i, \mathbf{1} \rangle = 1$ (Fact A.2) and that $\langle r_i, e_1 \rangle$ is equal to the $(i,1)$-th entry of $K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1}$.

Since each u'_i entry must be in $[0,1]$, each entry defines a constraint on q'. In particular, since these values are affine functions of q', each constraint corresponds to an interval. We can see that $q' = \frac{1}{2}$ is valid for each of these constraints, as $u'_i = \frac{1}{2}$ when $q' = \frac{1}{2}$ for all i. Thus it suffices to find a lower bound for q' using these constraints.

We ask next which values of u'_i are most extreme. For any fixed choice of $q' \in [0, \frac{1}{2}]$, we obtain the largest entry u'_i where $(K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})_{i1}$ is maximized and the smallest where that same value is minimized. In particular:

$$\max_i (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})_{i1} = \left(\prod_i (1-2q_i)^{-1} \right) \prod_i (1-q_i),$$

since $1 - q_i > q_i > 0$ for all i. (The constant $\prod_i (1-2q_i)^{-1}$ appears in all entries.) Similarly, we can see that:

$$\min_i (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1})_{i1} = \left(\prod_i (1-2q_i)^{-1} \right) (-q_j) \prod_{i \neq j} (1-q_i),$$

where $j = \arg\max_j q_j$, since this yields the most extreme negative term.

It suffices to constrain the two most extreme entries of u' to $[0,1]$. The constraints defined by these two entries are:

$$q' + (1-2q') \underbrace{\frac{\prod_i (1-q_i)}{\prod_i (1-2q_i)}}_{=:c_1} \leq 1 \iff q' \overset{(a)}{\geq} \frac{c_1 - 1}{2c_1 - 1},$$

and

$$q' + (1-2q') \underbrace{(-q_j) \frac{\prod_{i \neq j} (1-q_i)}{\prod_i (1-2q_i)}}_{=:-c_2} \geq 0 \iff q' \overset{(b)}{\geq} \frac{c_2}{1 + 2c_2}.$$

We note that $c_1 > 1$ and $c_2 > 0$. Moreover, comparing c_1 and c_2, we find that $c_1 > c_2 + 1$, as:

$$\begin{aligned}
c_1 - (c_2 + 1) &= \frac{\prod_i (1 - q_i)}{\prod_i (1 - 2q_i)} - \frac{q_j \prod_{i \neq j} (1 - q_i) + \prod_i (1 - 2q_i)}{\prod_i (1 - 2q_i)} \\
&= \frac{(1 - q_j) \prod_{i \neq j} (1 - q_i)}{\prod_i (1 - 2q_i)} - \frac{q_j \prod_{i \neq j} (1 - q_i) + \prod_i (1 - 2q_i)}{\prod_i (1 - 2q_i)} \\
&= \frac{(1 - 2q_j) \prod_{i \neq j} (1 - q_i) - \prod_i (1 - 2q_i)}{\prod_i (1 - 2q_i)} \\
&> 0.
\end{aligned}$$

So it follows that:

$$\begin{aligned}
\frac{c_1 - 1}{2c_1 - 1} - \frac{c_2}{1 + 2c_2} &= \frac{(c_1 - 1)(1 + 2c_2) - c_2(2c_1 - 1)}{(2c_1 - 1)(1 + 2c_2)} \\
&= \frac{c_2 - (c_1 + 1)}{(2c_1 - 1)(1 + 2c_2)} \\
&> 0,
\end{aligned}$$

which indicates that constraint (a) implies constraint (b). We denote by q^* the minimal q' allowed under constraint (a):

$$q^* = \frac{c_1 - 1}{2c_1 - 1} = \frac{\prod_i \frac{1 - q_i}{1 - 2q_i} - 1}{2 \prod_i \frac{1 - q_i}{1 - 2q_i} - 1}.$$

Putting this in terms of ε, note that $1 - q_i = \frac{e^{\varepsilon_i}}{e^{\varepsilon_i} + 1}$, $1 - 2q_i = \frac{e^{\varepsilon_i} - 1}{e^{\varepsilon_i} + 1}$, so:

$$\frac{1 - q_i}{1 - 2q_1} = \frac{e^{\varepsilon_i}}{e^{\varepsilon_i} - 1} = (1 - e^{-\varepsilon_i})^{-1},$$

and hence:

$$q^* = \frac{\prod_i (1 - e^{-\varepsilon_i})^{-1} - 1}{2 \prod_i (1 - e^{-\varepsilon_i})^{-1} - 1} = \frac{1 - \prod_i (1 - e^{-\varepsilon_i})}{2 - \prod_i (1 - e^{-\varepsilon_i})},$$

which gives a final ε^* of:

$$\begin{aligned}
\varepsilon^* &= q^{-1}(q^*) \\
&= \log \left((q^*)^{-1} - 1 \right) \\
&= \log \left(\frac{2 - \prod_i (1 - e^{-\varepsilon_i})}{1 - \prod_i (1 - e^{-\varepsilon_i})} - 1 \right) \\
&= \log \left(\frac{1}{1 - \prod_i (1 - e^{-\varepsilon_i})} \right) \\
&= -\log \left(1 - \prod_{i=1}^{k} (1 - e^{-\varepsilon_i}) \right).
\end{aligned}$$

Finally, to translate from transition probability matrices back to the theorem statement, note that $K_{\varepsilon^*}^{\text{merge}} = (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1}) K^{\text{or}} K_{\varepsilon^*}$ maps the $\{0,1\}^k$ state space to $\{0,1\}$—i.e., the 2^k possible inputs map to Bernoulli random variables with probabilities taken from the second column of $K_{\varepsilon^*}^{\text{merge}}$. It follows from the preceding discussion that this vector is precisely $(K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1}) v^*$. □

A.5 Proofs of Results for Section 1.4.3

Proof of Corollary 1.12. We prove this in the more general setting of merging k bits, as in Theorem A.3. Indeed, proving this is essentially equivalent to Theorem A.3, except that we must replace K^{or} with a transition matrix K^{and} that maps $(x_1, \ldots, x_k)$ to 1 only when $x_1 = \cdots = x_k = 1$. Then for fixed ε' we have a potential solution:

$$K_{\varepsilon'}^{\text{merge}} = (K_{\varepsilon_1}^{-1} \otimes \cdots \otimes K_{\varepsilon_k}^{-1}) K^{\text{and}} K_{\varepsilon'}.$$

Once again, we seek the largest ε' for which $K_{\varepsilon'}^{\text{merge}}$ is a valid transition probability matrix. This time, we will use the second column of $K^{\text{and}} K_{\varepsilon'}$ to determine constraints on ε'. (This is allowable since $K^{\text{and}} K_{\varepsilon'} \mathbf{1} = \mathbf{1}$.) We write the second column as:

$$w' = (q', \ldots, q', 1 - q')^T = q' \mathbf{1} + (1 - 2q') e_{2k}.$$

Using w' as in the proof of Theorem A.3, we obtain the same constraints on ε'. Applying the remainder of the proof of Theorem A.3 yields the final result. □

Proof of Lemma 1.13. Let $p_1 = e^{\varepsilon_1}/(e^{\varepsilon_1} + 1), p_2 = e^{\varepsilon_2}/(e^{\varepsilon_2} + 1)$. Using Fact A.1, we have that:

$$\mathcal{M}_{\varepsilon_1}^{\text{sym}}(x_i) \veebar \mathcal{M}_{\varepsilon_2}^{\text{sym}}(y_i) \sim \text{Bernoulli}(\theta_{x_i, y_i}),$$

where

$$\theta_{0,0} = \theta_{1,1} = p_1(1 - p_2) + p_2(1 - p_1)$$
$$\theta_{0,1} = \theta_{1,0} = p_1 p_2 + (1 - p_1)(1 - p_2).$$

Through a bit of algebra, we obtain

$$\begin{aligned}\theta_{0,0} = \theta_{1,1} &= \frac{e^{\varepsilon_1}}{(e^{\varepsilon_1}+1)(e^{\varepsilon_2}+1)} + \frac{e^{\varepsilon_2}}{(e^{\varepsilon_1}+1)(e^{\varepsilon_2}+1)} \\ &= \frac{e^{\varepsilon_1}+e^{\varepsilon_2}}{e^{\varepsilon_1+\varepsilon_2}+e^{\varepsilon_1}+e^{\varepsilon_2}+1} \\ &= \frac{1}{\frac{1+e^{\varepsilon_1+\varepsilon_2}}{e^{\varepsilon_1}+e^{\varepsilon_2}}+1},\end{aligned}$$

while

$$\theta_{1,0} = \theta_{0,1} = 1 - \theta_{0,0}.$$

So we have

$$\mathcal{M}_{\varepsilon_1}^{\text{sym}}(x_i) \veebar \mathcal{M}_{\varepsilon_2}^{\text{sym}}(y_i) \overset{D}{=} \mathcal{M}_{\theta_{1,0},1-\theta_{1,0}}(x_i \veebar y_i).$$

Finally, to obtain ε^*, note that $\mathcal{M}_{\theta_{1,0},1-\theta_{1,0}} = \mathcal{M}_{\varepsilon^*}^{\text{sym}}$ for

$$e^{\varepsilon^*} = \frac{1+e^{\varepsilon_1+\varepsilon_2}}{e^{\varepsilon_1}+e^{\varepsilon_2}}.$$

□

Proof of Corollary 1.15. Note that this theorem statement is identical to that of Theorem 1.3 except that condition (2) has changed from using the *or* operation $\vee$ to the *and* operation $\wedge$ and that the necessary condition is no longer on p_1, p_2 but instead q_1, q_2.

Consider the original condition (2) of Theorem 1.3. Since this must hold for all $x, y \in \{0,1\}$, we may alternatively write this condition in terms of $\neg x$ and $\neg y$ instead:

$$\begin{aligned}\mathcal{F}_{q_1,p_1}(x) \circ \mathcal{F}_{q_2,p_2}(y) &\overset{D}{=} \mathcal{F}_{p_1,q_1}(\neg x) \circ \mathcal{F}_{p_2,q_2}(\neg y) \\ &\overset{D}{=} f_1(\neg x) \circ f_2(\neg y) \\ &\overset{D}{=} f_3((\neg x) \vee (\neg y)) \\ &\overset{D}{=} f_3(\neg(x \wedge y)) \\ &\overset{D}{=} \mathcal{F}_{q_3,p_3}(x \wedge y).\end{aligned}$$

From Theorem 1.3, we know that we can only satisfy this condition simultaneously with (1) and (3) if $p_1 = p_2 = 1/2$. Recognizing that this statement is equivalent to condition (2) of the corollary but with the roles of p_i and q_i swapped, it is apparent that to satisfy (1)–(3) of our corollary, we must have $\bullet = \veebar$ and $q_1 = q_2 = 1/2$. □

Proof of Corollary 1.16. Assume conditions (1)–(4) are satisfied. By Theorem 1.3, we must have $p_1 = p_2 = 1/2$, while Corollary 1.15 states that $q_1 = q_2 = 1/2$. This results in a contradiction: $f_1(0) \overset{D}{=} \text{Bernoulli}(1/2) \overset{D}{=} f_1(1)$, violating (3). □

A.6 Proofs of Results for Section 1.5

Proof of Theorem 1.17. Let $f(\hat{n}) = \mathbb{E}[\ell(\hat{n}; T)]$. We will use the notation $\mathbb{E}_n[\cdot]$ to denote expectation under a cardinality of n, while abusing notation with $\mathbb{E}_{\hat{n}}[\cdot]$ to denote the equivalent quantity with $\hat{n}$ replacing n, as below:

$$\begin{aligned}
\mathbb{E}_n[T_{ij}] &= \mathbf{P}(T_{ij} = 1) \\
&= p(1 - \gamma_j^n) + q\gamma_j^n \\
&= p - (p - q)\gamma_j^n \\
\mathbb{E}_{\hat{n}}[T_{ij}] &= p - (p - q)\gamma_j^{\hat{n}}.
\end{aligned}$$

(Note that $\mathbb{E}_{\hat{n}}[\cdot]$ is not truly an expectation, since when $\hat{n}$ is non-integer, the distribution of T is not defined.) Observe that:

$$\begin{aligned}
\mathbb{E}_n[f'(\hat{n})] = B \sum_{j=1}^{P} &(1 - \mathbb{E}_n[T_{ij}])(p - q)\gamma_j^{\hat{n}} \log(\gamma_j)(1 - \mathbb{E}_{\hat{n}}[T_{ij}])^{-1} \\
&- B \sum_{j=1}^{P} \mathbb{E}_n[T_{ij}](p - q)\gamma_j^{\hat{n}} \log(\gamma_j)(\mathbb{E}_{\hat{n}}[T_{ij}])^{-1} \\
= B(p - q) &\sum_{j=1}^{P} \phi_j'(\hat{n}),
\end{aligned}$$

where

$$\phi_j'(\hat{n}) = \gamma_j^{\hat{n}} \log(\gamma_j) \left(\frac{1 - \mathbb{E}_n[T_{ij}]}{1 - \mathbb{E}_{\hat{n}}[T_{ij}]} - \frac{\mathbb{E}_n[T_{ij}]}{\mathbb{E}_{\hat{n}}[T_{ij}]} \right).$$

As expected, this equals zero when $\hat{n} = n$. Moreover, it is strictly positive for $\hat{n} < n$ and strictly negative for $\hat{n} > n$. Thus the same properties hold for $f'(\hat{n})$, and so n is the global maximizer of f.

By similar logic, we may write

$$\mathbb{E}_n[f''(\hat{n})] = B(p - q) \sum_{j=1}^{P} \phi_j''(\hat{n}),$$

where

$$\phi_j''(\hat{n}) = (\log \gamma_j)^2 \gamma_j^{\hat{n}} \left((1-p) \frac{1 - \mathbb{E}_n[T_{ij}]}{(1 - \mathbb{E}_{\hat{n}}[T_{ij}])^2} - p \frac{\mathbb{E}_n[T_{ij}]}{(\mathbb{E}_{\hat{n}}[T_{ij}])^2} \right).$$

Although $\phi_j''(\hat{n}) > 0$ for sufficiently large $\hat{n}$, note that the parenthetical quantity is monotonically increasing in $\hat{n}$. Moreover, we know $\phi_j''(n) < 0$ since $\phi_j'(n)$ corresponds to a maximum. Thus the parenthetical (and indeed all of $\phi_j''(\hat{n})$) must be negative for $\hat{n} \leq n$. Since this is true for all j, it follows that $f''(\hat{n}) < 0$ for $\hat{n} \leq n$. □

Lemma A.4. *Consider a bucket in a PCSA summary with v items where the bucket has $P = \infty$ bits. The probability that a new item allocated to the bucket modifies the bucket is bounded by c/v for all $v > v_0$ for some constants c, v_0.*

Proof. The probability a bucket containing v items is modified by a new item allocated to the bucket is $\sum_{\ell=1}^{\infty} 2^{-\ell} \left(1 - 2^{-\ell}\right)^v$. Split this sum into the ranges $\ell \in \mathcal{I}_1 := (0, \log_2 v - \log_2 \log_2 v)$, $\ell \in \mathcal{I}_2 := [\log_2 v - \log_2 \log_2 v, \log_2 v)$, $\ell \in \mathcal{I}_3 := [\log_2 v, \infty)$. Since $2^{-\ell} \leq 1$ and $(1 - 2^{-\ell})^v < \exp(-2^{-\ell} v) \leq 1$,

$$\begin{aligned}
\sum_{\ell \in \mathcal{I}_1} 2^{-\ell} \left(1 - 2^{-\ell}\right)^v &\leq \log_2 v \exp(-\log_2 v) = o(1), \\
\sum_{\ell \in \mathcal{I}_2} 2^{-\ell} \left(1 - 2^{-\ell}\right)^v &\leq \int_{\log_2 v - \log_2 \log_2 v}^{\log_2 v} 2^{-x} \exp(-2^{-x} v) dx \\
&= \frac{1}{v \log 2} \exp(-v 2^{-x}) \Big|_{\log_2 v - \log_2 \log_2 v}^{\log_2 v} \\
&= \frac{e^{-1}}{v \log(2)} - O(\exp(-\log_2 v)/v), \\
\sum_{\ell \in \mathcal{I}_3} 2^{-\ell} \left(1 - 2^{-\ell}\right)^v &\leq \sum_{\ell \in \mathcal{I}_3} 2^{-\ell} \leq \frac{1}{v}.
\end{aligned}$$

Summing these components gives the desired result. □

Proof of Theorem 1.18. The modified PCSA summary $\overline{S}_n$ can be generated in the following way. Draw a new cardinality $\overline{N} \sim \text{Poisson}(n)$. The first $\min\{n, \overline{N}\}$ items are shared for the regular SFM summary S_n and modified summary $\overline{S}_n$. Each of these items are allocated to the same bucket for both summaries. Denote the remaining items by $R = |\overline{N} - n|$. The variance of a Poisson(n) gives $R = O_p(\sqrt{n})$.

Using Theorem 3 in Kolchin et al. (1978) for the asymptotic distribution of the maximum value in a multinomial vector, the bucket allocated the largest number of remaining items has $O_p(R/B \log B) = O_p(\sqrt{n}/B \log B)$ items. Likewise the bucket with the minimum number of items has $n/B + O_p(\sqrt{n}/B \log B)$ items. By Lemma A.4, the probability a new item in a bucket will

update the bucket's value is $O(1/V_i)$ given V_i, the number of items already in the bucket. Thus, the probability that no bucket is updated by one of the remaining items is

$$\left(1 - \frac{O(\sqrt{n}/B \log B)}{n/B + O(\sqrt{n}/B \log B)}\right)^B + o(1) = \left(1 - \frac{O(\log B)}{\sqrt{n}}\right)^B + o(1)$$
$$= (1 - o(1/B))^B + o(1) \to 1$$

as $n \to \infty$. This gives that $\mathbf{P}(\overline{S}_n = S_n) \to 1$ as $n \to \infty$ and the true PCSA sketch and the Poissonized one are asymptotically equal.

We can also relate the Poissonized PCSA sketch to the one whose true likelihood is the composite marginal likelihood. By Poisson splitting, the entries of $\overline{S}_n$ are independent. Thus, we can couple the entries of $\overline{S}_n$ with those of $\tilde{S}_n$ via the inverse CDF method by using the same underlying Uniform$(0, 1)$ random variables. The probability that an entry in level j is different across the coupled sketches is

$$\begin{aligned}
\mathbf{P}(\tilde{S}_n(i,j) \neq \overline{S}_n(i,j)) &= \exp\left(-\frac{n}{B\,2^j}\right) - \left(1 - \frac{1}{B\,2^j}\right)^n \\
&= \exp\left(-\frac{n}{B\,2^j}\right)\left(1 - \exp\left[n \log\left(1 - \frac{1}{B2^j}\right) + \frac{n}{B2^j}\right]\right) \\
&< \exp\left(-\frac{n}{B\,2^j}\right)\left(1 - \exp\left[-\frac{n}{B^2 2^{2j+1}}\right]\right) \\
&< \exp\left(-\frac{n}{B\,2^j}\right)\frac{n}{B^2 2^{2j+1}}.
\end{aligned}$$

Applying a union bound gives and splitting the sum at some positive integer k gives

$$\begin{aligned}
\mathbf{P}(\tilde{S}_n \neq \overline{S}_n) &\leq B \sum_{j=1}^{\infty} \exp\left(-\frac{n}{B\,2^j}\right)\frac{n}{B^2 2^{2j+1}} \\
&\leq \sum_{j=1}^{k} \exp\left(-\frac{n}{B\,2^j}\right)\frac{n}{B} + \sum_{j=k+1}^{\infty} \frac{n}{B2^{2j+1}} \\
&\leq \exp\left(-\frac{n}{B\,2^k}\right)\frac{nk}{B} + \frac{n}{B2^{2k}}.
\end{aligned}$$

Take $k = \log_2\left(\frac{n/B}{2\log(n/B)}\right) + \delta$ for some $\delta \in [-1/2, 1/2)$. Then the first part

$$\exp\left(-\frac{n}{B\,2^k}\right)\frac{nk}{B} = \frac{nk}{B}\exp\left(-2^{1-\delta}\log(n/B)\right)$$

$$\leq \frac{nk}{B}(n/B)^{-3/2} \to 0.$$

as $n/B \to \infty$. Likewise, the second part

$$\frac{n}{B2^{2k}} = \frac{n}{B}\left(\frac{n/B}{2\log(n/B)}\right)^{-2} 2^{-2\delta} = \frac{4\log^2(n/B)}{n/B}2^{-2\delta} \to 0$$

as $n/B \to \infty$. Thus $\mathbf{P}(\tilde{S}_n = \overline{S}_n) \to 1$ as $n \to \infty$ as well.

□

Proof of Corollary 1.19. Since both a PCSA sketch S_n and modified sketch with independent bins $\tilde{S}_n$ are equal with probability going to 1 as $n \to \infty$, the private SFM sketches T_n, $\tilde{T}_n$ obtained by applying the same randomized response noise to them are also equal with probability going to 1. Let $\hat{\theta}(T_n)$ be some cardinality estimator and $V(\hat{\theta}(T_n))$ denote its asymptotic variance. Then $\min_{\hat{\theta}\in\Theta} V(\hat{\theta}(T_n)) = \min_{\hat{\theta}\in\Theta} V(\hat{\theta}(\tilde{T}_n))$, and a cardinality estimator for T_n is asymptotically efficient if and only if it is asymptotically efficient for $\tilde{T}_n$. Since the composite likelihood estimator for T_n is the true maximum likelihood estimator for $\tilde{T}_n$, it is asymptotically efficient. □

Appendix B

Supporting Proofs for Chapter 2

The proofs of the results from Section 2.5 closely follow the techniques of Lei and Rinaldo (2015). For both SBM and DCBM, we will require the following notation. Let $\texttt{eig}_k(\cdot) : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times k}$ denote a function that returns the eigenvectors of its argument corresponding to the k largest eigenvalues in absolute value.

Let $Y \sim \text{DCBM}(\theta, \psi, B)$, where $\psi = 1$ for SBM. Let P be the $n \times n$ matrix with entries $P_{ij} = \psi_i \psi_j B_{\theta_i \theta_j}$. Note that $E[Y] = P - \text{diag}(P)$, as Y has zeros on the diagonal by construction. Let $\hat{U} = \texttt{eig}_k(Y)$ denote the eigenvectors of the observed non-private adjacency matrix Y, and $U = \texttt{eig}_k(P)$ denote the eigenvectors of the expected adjacency matrix (plus a diagonal).

We will use a $\downarrow$ subscript to denote private analogs to these quantities. Assuming $\varepsilon < \infty$, let $Y_\downarrow = (d_\varepsilon \circ \mathcal{M}_\varepsilon)(Y)$ denote the edge-flipped and downshifted synthetic network. Since $E[Y_\downarrow] = \frac{e^\varepsilon - 1}{e^\varepsilon + 1} E[Y]$ (Corollary 2.9), we let $P_\downarrow = \frac{e^\varepsilon - 1}{e^\varepsilon + 1} P$, and let $\hat{U}_\downarrow = \texttt{eig}_k(Y_\downarrow)$ and $U_\downarrow = \texttt{eig}_k(P_\downarrow)$. If $\varepsilon = \infty$, we have no privacy, so we let $Y_\downarrow = Y, P_\downarrow = P, \hat{U}_\downarrow = \hat{U}, U_\downarrow = U$.

A critical fact for the proofs of the main theorems is that the matrix $U_\downarrow$ can be chosen to be precisely equal to U, even if $\varepsilon < \infty$, since if $V \Lambda V^T$ is an eigendecomposition of P, then $V \left(\frac{e^\varepsilon - 1}{e^\varepsilon + 1} \Lambda \right) V^T$ is an eigendecomposition of $P_\downarrow$. For this reason, we will assume below that $U_\downarrow = U$.

We begin with a lemma that captures the main technical distinction between our results and those of Lei and Rinaldo (2015).

Lemma B.1. *Suppose* $Y \sim \text{DCBM}(\theta, \psi, B)$ *with* $\max_{i \in C_j} \psi_i = 1$ *for* $j \in [k]$, B *full rank,* $\max B \geq \log n / n$, *and minimum absolute eigenvalue* $\lambda_B > 0$. *Let* $\varepsilon \in (0, \infty]$, *and let* $g_\varepsilon(B)$ *as defined in eq. (2.6). There exists a universal constant* c_0 *and a* $k \times k$ *orthogonal matrix* Q *such that:*

$$P\left(\|\hat{U}_\downarrow - U_\downarrow Q\|_F \leq c_0 \frac{2\sqrt{2kng_\varepsilon(B)}}{\tilde{n}_{min} \lambda_B} \right) \geq 1 - n^{-1}.$$

Proof. By Lemma 5.1 of Lei and Rinaldo (2015), there exists a $k \times k$ orthogonal matrix Q such that:

$$\begin{aligned}\|\hat{U}_\downarrow - U_\downarrow Q\|_F &\leq \frac{2\sqrt{2k}}{\lambda_{P_\downarrow}}\|Y_\downarrow - P_\downarrow\| \\ &= \frac{2\sqrt{2k}}{\lambda_{P_\downarrow}}\|\mathcal{M}_\varepsilon(Y) - E[\mathcal{M}_\varepsilon(Y)] - \operatorname{diag}(P_\downarrow)\| \\ &\leq \frac{2\sqrt{2k}}{\lambda_{P_\downarrow}}\left(\|\mathcal{M}_\varepsilon(Y) - E[\mathcal{M}_\varepsilon(Y)]\| + 1\right),\end{aligned}$$

where the last inequality follows from the fact that $\|\operatorname{diag}(P_\downarrow)\| \leq 1$, as every entry of $P_\downarrow$ is bounded in the unit interval. From here, it remains to find a lower bound for $\lambda_{P_\downarrow}$ and an upper bound for $\|\mathcal{M}_\varepsilon(Y) - E[\mathcal{M}_\varepsilon(Y)]\|$.

We begin with the simpler task of bounding $\lambda_{P_\downarrow}$. Consider first the case when $\varepsilon = \infty$. In this case, $P_\downarrow = P$. From the proof of Lei and Rinaldo (2015) Lemma 4.1, we know that the nonzero eigenvalues of P are precisely those of $\Psi B \Psi$, where $\Psi = \operatorname{diag}(\sqrt{\tilde{n}_1}, \ldots, \sqrt{\tilde{n}_k})$. Since both B and Ψ are symmetric and invertible, we can say that:

$$\begin{aligned}\|(\Psi B \Psi)^{-1}\| &\leq \|\Psi^{-1}\|\|B^{-1}\|\|\Psi^{-1}\| \\ &= \lambda_\Psi^{-1}\lambda_B^{-1}\lambda_\Psi^{-1} \\ &= \tilde{n}_{\min}^{-1}\lambda_B^{-1}.\end{aligned}$$

The fact that $\|(\Psi B \Psi)^{-1}\|$ is the largest absolute eigenvalue of $(\Psi B \Psi)^{-1}$ further implies that the smallest absolute eigenvalue of $\Psi B \Psi$ satisfies:

$$\lambda_P = \lambda_{\Psi B \Psi} \geq \tilde{n}_{\min}\lambda_B.$$

When $\varepsilon < \infty$, recall that $P_\downarrow = \frac{e^\varepsilon - 1}{e^\varepsilon + 1}P$, so:

$$\lambda_{P_\downarrow} \geq \begin{cases}\tilde{n}_{\min}\lambda_B, & \varepsilon = \infty \\ \frac{e^\varepsilon - 1}{e^\varepsilon + 1}\tilde{n}_{\min}\lambda_B, & \varepsilon < \infty\end{cases}.$$

Next, we can upper bound $\|\mathcal{M}_\varepsilon(Y) - E[\mathcal{M}_\varepsilon(Y)]\|$ using Theorem 5.2 of Lei and Rinaldo (2015). First, we establish necessary bounds on the entries of $E[\mathcal{M}_\varepsilon(Y)]$. Let:

$$\mu = \begin{cases}\max B, & \varepsilon = \infty \\ \frac{e^\varepsilon - 1}{e^\varepsilon + 1}(\max B) + \frac{2}{e^\varepsilon + 1}\left(\frac{1}{2}\right), & \varepsilon < \infty\end{cases}.$$

In both cases, we can see that μ is an upper bound for $\max E[\mathcal{M}_\varepsilon(Y)]$, per the definition of DCBM and Lemma 2.7. Additionally, we can view μ as an affine combination of $\max B$ and $\frac{1}{2}$, from which it is clear that $\mu \geq \min\{\frac{1}{2}, \max B\} \geq \log n/n$ (for $n \geq 1$). So by Lei and Rinaldo (2015) Theorem 5.2, with probability at least $1 - n^{-1}$:

$$\|\mathcal{M}_\varepsilon(Y) - E[\mathcal{M}_\varepsilon(Y)]\| \leq C\sqrt{n\mu},$$

where $C = C(1,1)$ for $C(\cdot,\cdot)$ defined in Lei and Rinaldo (2015). Combining some facts, we now have that with probability at least $1 - n^{-1}$:

$$\|\hat{U}_\downarrow - U_\downarrow Q\|_F \leq 2\sqrt{2k}\frac{C\sqrt{n\mu}+1}{\lambda_{P_\downarrow}}.$$

Since $\mu \geq \log n/n$, then $1 \leq (\log 2)^{-1/2}\sqrt{n\mu}$ (for $n > 1$). Thus if we let $c_0 = C + (\log 2)^{-1/2}$, we can simplify the above to:

$$\|\hat{U}_\downarrow - U_\downarrow Q\|_F \leq c_0\frac{2\sqrt{2kn\mu}}{\lambda_{P_\downarrow}}.$$

Finally, it remains to put these results in terms of $g_\varepsilon(B)$ and λ_B. When $\varepsilon = \infty$, it is clear that:

$$\frac{\sqrt{\mu}}{\lambda_{P_\downarrow}} \leq \frac{\sqrt{\max B}}{\tilde{n}_{\min}\lambda_B} = \frac{\sqrt{g_\varepsilon(B)}}{\tilde{n}_{\min}\lambda_B}.$$

When $\varepsilon < \infty$, this same inequality holds, albeit with different intermediate steps:

$$\begin{aligned}\frac{\sqrt{\mu}}{\lambda_{P_\downarrow}} &\leq \frac{1}{\tilde{n}_{\min}\lambda_B}\frac{e^\varepsilon+1}{e^\varepsilon-1}\sqrt{\frac{(e^\varepsilon-1)\max B+1}{e^\varepsilon+1}} \\ &= \frac{1}{\tilde{n}_{\min}\lambda_B}\sqrt{\frac{e^\varepsilon+1}{e^\varepsilon-1}\left(\max B + \frac{1}{e^\varepsilon-1}\right)} \\ &\leq \frac{\sqrt{g_\varepsilon(B)}}{\tilde{n}_{\min}\lambda_B}.\end{aligned}$$

Thus, we conclude that with probability at least $1 - n^{-1}$:

$$\|\hat{U}_\downarrow - U_\downarrow Q\|_F \leq c_0\frac{2\sqrt{2kng_\varepsilon(B)}}{\tilde{n}_{\min}\lambda_B}.$$

□

B.1 The k-means and k-medians Problems

Recall that Algorithm 1 requires solving an approximate k-means problem, while Algorithm 2 requires solving an approximate k-medians problem. The proofs of Theorems 2.10 and 2.11 rely on some additional notation and properties surrounding these problems.

Let $\mathbb{G}^{n\times k}$ denote the set of $n \times k$ membership matrices consisting of a single one in each row with zeroes elsewhere. In both the k-means and k-medians problems applied to the rows of the matrix $\hat{U}$, we seek a membership matrix $\Theta \in \mathbb{G}^{n\times k}$ and set of centroids $X \in \mathbb{R}^{k\times k}$ that minimize the distance $\|\Theta X - \hat{U}\|_*$ for a suitable norm $\|\cdot\|_*$. In k-means, the norm chosen is the Frobenius norm, while in the k-medians problem, the norm chosen is the (2,1) norm, $\|A\|_{2,1} = \sum_i \|A_{i*}\|_2$. Finding an exact solution for each of these problems is difficult, but efficient approximation algorithms exist. We will call a solution $\hat{\Theta}\hat{X}$ a $(1+\gamma)$-approximate solution to the k-means problem if:

$$\|\hat{\Theta}\hat{X} - \hat{U}\|_F^2 \leq (1+\gamma)\left[\inf_{\Theta' \in \mathbb{G}^{n,k}, X'_{k\times k}} \|\Theta' X' - \hat{U}\|_F^2\right].$$

Similarly, we will call $\hat{\Theta}\hat{X}$ a $(1+\gamma)$-approximate solution to the k-medians problem if:

$$\|\hat{\Theta}\hat{X} - \hat{U}\|_{2,1} \leq (1+\gamma)\left[\inf_{\Theta' \in \mathbb{G}^{n,k}, X'_{k\times k}} \|\Theta' X' - \hat{U}\|_{2,1}\right]. \tag{B.1}$$

In the main text, we denote the membership parameter θ as a vector. One can easily convert a membership matrix $\Theta \in \mathbb{G}^{n\times k}$ to a membership vector $\theta \in [k]^n$ by choosing $\theta_i = \arg\max_j \Theta_{ij}$.

B.2 SBM

Proof of Theorem 2.10. We begin by recalling that $\text{SBM}(\theta, B) \overset{D}{=} \text{DCBM}(\theta, \mathbf{1}_n, B)$ (2.1), and so by Lemma B.1, there exists a universal constant c_0 and orthogonal matrix Q such that with probability at least $1 - n^{-1}$:

$$\|\hat{U}_\downarrow - U_\downarrow Q\|_F \leq c_0 \frac{2\sqrt{2kng_\varepsilon(B)}}{n_{\min}\lambda_B}. \tag{B.2}$$

From here, the proof follows the same line of argument as Lei and Rinaldo (2015) Theorem 3.1 and Corollary 3.2. For completeness, and since our parameterization is a bit different, we include the remainder of the proof here.

Since $U_\downarrow = U$, we know from Lei and Rinaldo (2015) Lemma 3.1 that $U_\downarrow Q = \Theta X'$ for some

$\Theta \in \mathbb{G}^{n,k}$ and $X'_{k\times k}$ satisfying:

$$\|X'_{j*} - X'_{\ell *}\|_2 = \sqrt{n_j^{-1} + n_\ell^{-1}}$$

for $j \neq \ell$. Now we wish to apply Lemma 5.3 of Lei and Rinaldo (2015). For $j \in [k]$, choose $\delta_j = \sqrt{n_j^{-1} + (\max_{\ell \neq j} n_\ell)^{-1}}$, and define S_j as in Lemma 5.3 of Lei and Rinaldo (2015). We want to show that $(16+8\gamma)\|\hat{U}_\downarrow - U_\downarrow Q\|_F^2/\delta_j^2 < n_j$ for $j \in [k]$. Since $\delta_j^2 n_j > 1$ for all j, it suffices to show that $(16+8\gamma)\|\hat{U}_\downarrow - U_\downarrow Q\|_F^2 \leq 1$. Under the condition that (B.2) holds, we have that:

$$(16+8\gamma)\|\hat{U}_\downarrow - U_\downarrow Q\|_F^2 \leq \frac{64c_0^2(2+\gamma)kng_\varepsilon(B)}{n_{\min}^2\lambda_B^2}$$

Choosing $c_1 = 64c_0^2$, then (2.7) implies that the above is ≤ 1, as desired. Thus, for each community $j \in [k]$, the set of nodes that are possibly misclassified by Algorithm 1 must be a subset of S_j, and since $\delta_j^2 > n_j^{-1}$:

$$\begin{aligned}\sum_{j=1}^k \frac{|S_j|}{n_j} \leq \sum_{j=1}^k |S_j|\delta_j^2 &\leq 4(4+2\gamma)\|\hat{U}_\downarrow - U_\downarrow Q\|_F^2 \\ &\leq 64c_0^2\frac{(2+\gamma)kng_\varepsilon(B)}{n_{\min}^2\lambda_B^2} \\ &= c_1\frac{(2+\gamma)kng_\varepsilon(B)}{n_{\min}^2\lambda_B^2}.\end{aligned}$$

So then we have that:

$$\begin{aligned}\tilde{L}(\theta, \hat{\theta}_\varepsilon) &\leq \max_{j\in k}\frac{|S_j|}{n_j} \leq c_1\frac{(2+\gamma)kn}{n_{\min}^2\lambda_B^2}g_\varepsilon(B) \\ L(\theta, \hat{\theta}_\varepsilon) &\leq \frac{1}{n}\sum_{j=1}^k |S_j| \leq c_1\frac{(2+\gamma)kn'_{\max}}{n_{\min}^2\lambda_B^2}g_\varepsilon(B).\end{aligned}$$

□

B.3 DCBM

Proof of Theorem 2.11. We begin with the results of Lemma B.1, which states that there exists a universal constant c_0 and a $k \times k$ orthogonal matrix Q such that with probability at least $1 - n^{-1}$:

$$\|\hat{U}_\downarrow - U_\downarrow Q\|_F \leq c_0\frac{2\sqrt{2kng_\varepsilon(B)}}{\tilde{n}_{\min}\lambda_B}.$$

The remainder of the proof follows very closely from the proofs of Lemma A.1 and Theorem 4.2 from Lei and Rinaldo (2015). We define the sets I_0, I_+, and S as in the original proofs. Recall that Algorithm 2 requires separate handling of zero vs. non-zero rows of $\hat{U}_\downarrow$. The set $I_0 = \{i \in [n] \mid (\hat{U}_\downarrow)_{i*} = 0\}$ holds the set of nodes whose embeddings are zero, and $I_+ = [n] \setminus I_0$ holds the remainder. We allow that the nodes in I_0 will be misclassified, and we define a set $S \subseteq I_+$ in which we also allow misclassification. Our goal, then, is to bound $|I_0 \cup S|$. We start by bounding $|I_0|$. Note that:

$$\begin{aligned}
\|\hat{U}_\downarrow - U_\downarrow Q\|_F^2 &\geq \sum_{i \in I_0} \|(U_\downarrow Q)_{i*}\|_2^2 \\
&\geq \frac{|I_0|^2}{\sum_{i \in I_0} \|(U_\downarrow Q)_{i*}\|_2^{-2}} \quad \text{(Cauchy–Schwarz)} \\
&\geq \frac{|I_0|^2}{\sum_{i=1}^n \|U_{i*}\|_2^{-2}} \quad (U_\downarrow = U, Q \text{ orthogonal})
\end{aligned}$$

From Lei and Rinaldo (2015) Lemma 4.1, we know that $\|U_{i*}\|_2^{-2} = \tilde{n}_{\theta_i} \psi_i^{-2}$, so $\sum_{i=1}^n \|U_{i*}\|_2^{-2} = \sum_{j=1}^k n_j^2 \nu_j$. Thus:

$$|I_0| \leq \|\hat{U}_\downarrow - U_\downarrow Q\|_F \sqrt{\sum_{j=1}^k n_j^2 \nu_j}.$$

Moving on to I_+, we construct $\hat{U}'_\downarrow$ of size $|I_+| \times k$ to be the row-normalized version of $\hat{U}_\downarrow$, excluding zero rows, i.e., for $i = 1, \ldots, |I_+|$ and $j = (I_+)_i$:

$$(\hat{U}'_\downarrow)_{i*} = (\hat{U}_\downarrow)_{j*} / \|(\hat{U}_\downarrow)_{j*}\|_2.$$

Let $U'_\downarrow$ be the $n \times k$ row-normalized version of the expected embeddings $U_\downarrow$, where zero rows in $U_\downarrow$ are preserved as zero rows in $U'_\downarrow$. Then let $U''_\downarrow$ to be the $|I_+| \times k$ matrix constructed from the non-zero rows of $U'_\downarrow$, i.e., for $i = 1, \ldots, |I_+|$ and $j = (I_+)_i$:

$$(U''_\downarrow)_{i*} = (U'_\downarrow)_{j*}.$$

By (B.1), if $\hat{\Theta}_+ \hat{X}$ is a $(1+\gamma)$-approximate solution to the k-medians problem on $\hat{U}_\downarrow$, then $\|\hat{\Theta}_+ \hat{X} - \hat{U}'_\downarrow\|_{2,1} \leq (1+\gamma)\|\hat{U}'_\downarrow - U''_\downarrow Q\|_{2,1}$, and so:

$$\begin{aligned}
\|\hat{\Theta}_+ \hat{X} - U''_\downarrow Q\|_{2,1} &\leq \|\hat{\Theta}_+ \hat{X} - \hat{U}'_\downarrow\|_{2,1} + \|\hat{U}'_\downarrow - U''_\downarrow Q\|_{2,1} \\
&\leq (2+\gamma)\|\hat{U}'_\downarrow - U''_\downarrow Q\|_{2,1}
\end{aligned}$$

Using the fact that, for any vectors v_1, v_2 of equal dimension, $\|\frac{v_1}{\|v_1\|} - \frac{v_2}{\|v_2\|}\| \leq 2\frac{\|v_1 - v_2\|}{\|v_1\|}$ (Lei and Rinaldo, 2015), we can bound the above (2,1) norm as follows:

$$\begin{aligned}
\|\hat{U}'_\downarrow - U''_\downarrow Q\|_{2,1} &= \sum_{i \in I_+} \|(\hat{U}'_\downarrow)_{i*} - (U''_\downarrow Q)_{i*}\|_2 \quad \text{(norm definition)} \\
&\leq 2 \sum_{i=1}^{n} \frac{\|(\hat{U}_\downarrow)_{i*} - (U_\downarrow Q)_{i*}\|_2}{\|(U_\downarrow Q)_{i*}\|_2} \quad \text{(fact above)} \\
&= 2\sqrt{\left(\sum_{i=1}^{n} \|(\hat{U}_\downarrow)_{i*} - (U_\downarrow Q)_{i*}\|_2 \|(U_\downarrow)_{i*}\|_2^{-1}\right)^2} \\
&\leq 2\sqrt{\left(\sum_{i=1}^{n} \|(\hat{U}_\downarrow)_{i*} - (U_\downarrow Q)_{i*}\|_2^2\right)\left(\sum_{i=1}^{n} \|(U_\downarrow)_{i*}\|_2^{-2}\right)} \quad \text{(Cauchy–Schwarz)} \\
&= 2\sqrt{\|\hat{U}_\downarrow - U_\downarrow Q\|_F^2 \left(\sum_{i=1}^{n} \|U_{i*}\|_2^{-2}\right)} \quad (U_\downarrow = U) \\
&= 2\|\hat{U}_\downarrow - U_\downarrow Q\|_F \sqrt{\sum_{j=1}^{k} n_j^2 v_j}
\end{aligned}$$

Let $S = \{i \in I_+ : \|(\hat{\Theta}_+)_{i*}\hat{X} - (U''_\downarrow Q)_{i*}\|_2 \geq 2^{-1/2}\}$. These are the nodes in I_+ for which the corresponding fitted cluster centroids are located more than $2^{-1/2}$ from their expectation (up to the transformation Q). Observe:

$$2^{-1/2}|S| \leq \sum_{i \in I_+} \|(\hat{\Theta}_+)_{i*}\hat{X} - (U''_\downarrow Q)_{i*}\|_2 = \|\hat{\Theta}_+\hat{X} - U''_\downarrow Q\|_{2,1}$$

Thus we can bound $|S|$:

$$\begin{aligned}
|S| &\leq \sqrt{2}\|\hat{\Theta}_+\hat{X} - U''_\downarrow Q\|_{2,1} \\
&\leq \sqrt{2}(2+\gamma)\|\hat{U}'_\downarrow - U''_\downarrow Q\|_{2,1} \\
&\leq 2\sqrt{2}(2+\gamma)\|\hat{U}_\downarrow - U_\downarrow Q\|_F \sqrt{\sum_{j=1}^{k} n_j^2 v_j}
\end{aligned}$$

And so we can bound $|I_0 \cup S|$:

$$|I_0 \cup S| = |I_0| + |S| \leq \left[1 + 2\sqrt{2}(2+\gamma)\right] \|\hat{U}_\downarrow - U_\downarrow Q\|_F \sqrt{\sum_{j=1}^{k} n_j^2 v_j}$$

With probability $1 - n^{-1}$, this is further bounded:

$$|I_0 \cup S| \le [1 + 2\sqrt{2}(2+\gamma)]c_0 \frac{2\sqrt{2kng_\varepsilon(B)}}{\tilde{n}_{\min}\lambda_B}\sqrt{\sum_{j=1}^{k} n_j^2 v_j}$$
$$\le 8c_0(2.5+\gamma)\frac{\sqrt{kng_\varepsilon(B)}}{\tilde{n}_{\min}\lambda_B}\sqrt{\sum_{j=1}^{k} n_j^2 v_j}$$

Under the condition that the above is less than $n_{\min}$, which is implied by (2.9), then for each $j \in [k]$, there must exist $i \in (C_j \setminus (I_0 \cup S))$—i.e., every block has at least one node whose normalized spectral embedding is within $2^{-1/2}$ of its expectation (in ℓ_2 norm). Because the expected cluster centers U''_{i*} are orthogonal and have unit norm, we have $\|U''_{i*} - U''_{j*}\|_2 = \sqrt{2}$ whenever $\theta_i \neq \theta_j$ (or 0 otherwise). Thus for any two nodes $i, j \notin I_0 \cup S$, we have:

$$\begin{aligned}
(\hat{\Theta}_+)_{i*} = (\hat{\Theta}_+)_{j*} \implies & \|U''_{i*} - U''_{j*}\|_2 \\
& \le \|(\hat{\Theta}_+)_{i*}\hat{X} - U'_{i*}Q\|_2 + \|(\hat{\Theta}_+)_{j*}\hat{X} - U'_{j*}Q\|_2 \\
& < \sqrt{2} \\
\implies & U''_{i*} = U''_{j*} \\
\implies & \theta_i = \theta_j
\end{aligned}$$

In other words, the set of nodes that are misclassified must be a subset of $I_0 \cup S$, and so $L(\theta, \hat{\theta}_\varepsilon) \le \frac{|I_0 \cup S|}{n}$. The final theorem statement is obtained by choosing $c_2 = 8c_0$. □

Proof of Lemma 2.12. We begin by stating a few key facts under the assumed conditions:

$$\begin{aligned}
v_j &= \left(\frac{1}{n_j}\sum_{i\in C_j}\psi_i^2\right)\left(\frac{1}{n_j}\sum_{i\in C_j}\psi_i^{-2}\right) \in [a^2, a^{-2}] \\
\tilde{n}_{\min} &= \min_{j\in[k]}\sum_{i\in C_j}\psi_i^2 \in [a^2 n_{\min}, n_{\min}] \\
n_{\min} &= \Theta(n/k)
\end{aligned}$$

From here, we apply Theorem 2.11. First, we want to show that under the assumed conditions,

(2.9) is satisfied for large n. A lower bound for the RHS of (2.9) is:

$$c_2^{-1}\frac{n_{\min}}{\sqrt{\sum_{j=1}^k n_j^2 \nu_j}} = \Omega\left(\frac{n/k}{\sqrt{k(n/k)^2 a^{-2}}}\right) = \Omega\left(k^{-1/2}a\right)$$

Then looking at the LHS of (2.9), we can write:

$$\frac{(2.5+\gamma)\sqrt{kn\, g_\varepsilon(B)}}{\tilde{n}_{\min}\lambda_B} = O\left(\frac{\sqrt{kn\, g_\varepsilon(B)}}{a^2(n/k)\lambda(B)}\right) = O\left(\frac{k^{3/2}\sqrt{g_\varepsilon(B)}}{a^2\lambda(B)\sqrt{n}}\right)$$

Thus Theorem 2.11 is applicable for large n if:

$$\frac{k^{3/2}\sqrt{g_\varepsilon(B)}}{a^2\lambda(B)\sqrt{n}} = o\left(k^{-1/2}a\right)$$

which is satisfied by assumption. Thus we conclude that with probability $1 - n^{-1}$:

$$\begin{aligned} L(\theta, \hat{\theta}_\varepsilon) &\le c_2 \frac{(2.5+\gamma)}{\tilde{n}_{\min}\lambda_B}\sqrt{\frac{k}{n}\left(\sum_{j=1}^k n_j^2 \nu_j\right) g_\varepsilon(B)} \\ &= O\left(\frac{1}{a^2(n/k)\lambda_B}\sqrt{\frac{k}{n}(k(n/k)^2 a^{-2}) g_\varepsilon(B)}\right) \\ &= O\left(\frac{k\sqrt{g_\varepsilon(B)}}{a^3\lambda_B\sqrt{n}}\right). \end{aligned}$$

□

B.4 Miscellaneous

Below we prove a useful inequality for $g_\varepsilon(B)$ claimed in Section 2.5.

Fact B.2. *Let $k \in \mathbb{N}, \varepsilon \in (0,\infty), B \in [0,1]^{k\times k}$. Then:*

$$g_\infty(B) < g_\varepsilon(B) \le \max B + 3\zeta_\varepsilon^{-1} + 2\zeta_\varepsilon^{-2}$$

where $\zeta_\varepsilon = e^\varepsilon - 1$.

Proof. Note first that:

$$\frac{e^\varepsilon + 1}{e^\varepsilon - 1} = \frac{e^\varepsilon - 1}{e^\varepsilon - 1} + \frac{2}{e^\varepsilon - 1} = 1 + 2\zeta_\varepsilon^{-1}.$$

Thus:

$$\begin{aligned} g_\varepsilon(B) &= \frac{e^\varepsilon + 1}{e^\varepsilon - 1}\left(\max B + \frac{1}{e^\varepsilon - 1}\right) \\ &= (1 + 2\zeta_\varepsilon^{-1})(\max B + \zeta_\varepsilon^{-1}) \\ &= \max B + \zeta_\varepsilon^{-1} + 2\zeta_\varepsilon^{-1} \max B + 2\zeta_\varepsilon^{-2} \\ &\leq \max B + 3\zeta_\varepsilon^{-1} + 2\zeta_\varepsilon^{-2} \end{aligned}$$

where the last line follows from the fact that $\max B \leq 1$. The fact that $g_\infty(B) < g_\varepsilon(B)$ also follows from the above, as ζ_ε (and thus ζ_ε^{-1}, ζ_ε^{-2}) is strictly positive. □

B.5 Privacy Guarantee

Finally, for completeness, we conclude with a formal proof of the privacy guarantee.

Proof of Theorem 2.6. Let Y, Y' be two binary, undirected networks of n nodes differing on one edge, (i, j) with $1 \leq i < j \leq n$. Then for all $k \neq i$, we have that $\mathcal{M}_k(Y_{k*}) \overset{D}{=} \mathcal{M}_k(Y'_{k*})$. Thus, all that remains to show is that for any $a \in \{0, 1\}^n$:

$$P(\mathcal{M}_i(Y_{i*}) = a) \leq e^\varepsilon P(\mathcal{M}_i(Y'_{i*}) = a).$$

To simplify notation, we will write $\mathcal{M}_i$ in another form. Let $\mathcal{M}' : \{0, 1\} \to \{0, 1\}$ such that:

$$\mathcal{M}'(x) = \begin{cases} 1 - x & \text{w.p.} \quad \frac{1}{1+e^\varepsilon} \\ x & \text{w.p.} \quad \frac{e^\varepsilon}{1+e^\varepsilon} \end{cases}.$$

Then we can write:

$$\mathcal{M}_i(Y_{i*}) = \left(\underbrace{0, \ldots, 0}_{i \text{ entries}}, \ \mathcal{M}'(Y_{i,i+1}), \ldots, \mathcal{M}'(Y_{in})\right),$$

where the $\mathcal{M}'(\cdot)$ are taken independently. Since the first i entries of $\mathcal{M}_i(Y_{i*})$ are deterministic, we can safely restrict our attention to the case when $a_k = 0$ for $k \leq i$ (since otherwise we have

$P(\mathcal{M}_i(Y_{i*}) = a) = P(\mathcal{M}_i(Y'_{i*}) = a) = 0$). Thus:

$$\begin{aligned}
\frac{P(\mathcal{M}_i(Y_{i*}) = a)}{P(\mathcal{M}_i(Y'_{i*}) = a)} &= \frac{\prod_{\ell=i+1}^{n} P(\mathcal{M}'(Y_{i\ell}) = a_\ell)}{\prod_{\ell=i+1}^{n} P(\mathcal{M}'(Y_{i\ell}) = a_\ell)} \\
&= \frac{P(\mathcal{M}'(Y_{ij}) = a_j)}{P(\mathcal{M}'(Y'_{ij}) = a_j)} \\
&\leq \frac{P(\mathcal{M}'(1) = 1)}{P(\mathcal{M}'(0) = 1)} \\
&= e^{\varepsilon},
\end{aligned}$$

where the inequality is taken by considering all combinations of Y_{ij}, Y'_{ij}, and a_j. □

Appendix C

Supporting Proofs for Chapter 3

C.1 Preliminaries

We begin by defining the matrix absolute value and discussing some of its properties.

Definition C.1. *For a matrix $A \in \mathbb{R}^{m\times n}$, we define the matrix absolute value $|A| = \sqrt{A^TA}$. In particular, when $D = \operatorname{diag}(d_1, \ldots, d_n)$, we have $|D| = \operatorname{diag}(|d_1|, \ldots, |d_n|)$. For symmetric matrices $A = A^T$ with eigendecomposition $A = U\Lambda U^T$, we have $|A| = U|\Lambda|U^T$.*

Fact C.2. *$|A|$ is the unique positive semi-definite square root of A^TA.*

Proof. See Horn and Johnson (2012, Theorem 7.3.1). □

Fact C.3. *If $A = A^T$ and $A = U\Sigma V^T$ is a singular value decomposition of A, then $|A| = U\Sigma U^T$.*

Proof. We may write $A^TA = AA^T = U\Sigma V^TV\Sigma U^T = U\Sigma^2U^T$. Note that

$$U\Sigma U^T \succeq 0 \quad \text{and} \quad (U\Sigma U^T)(U\Sigma U^T) = A^2 = A^TA.$$

So by Fact C.2, $|A| = U\Sigma U^T$ is the unique positive semi-definite square root of A^TA. □

Fact C.4. *Suppose $A = XDX^T$, where X^TX is diagonal and D is a diagonal matrix with diagonal entries in $\{\pm 1\}$. Then $|A| = XX^T$.*

Proof. Write A^TA as follows:

$$\begin{aligned} A^TA &= XDX^TXDX^T \\ &= XD^2(X^TX)X^T \quad \text{(diagonals commute)} \\ &= XX^TXX^T \quad (D^2 = I) \\ &= (XX^T)^2. \end{aligned}$$

Since $XX^T \succeq 0$, $|A| = XX^T$ is the unique positive semi-definite square root of A^TA. □

Fact C.5. *If U is orthogonal, then $|UAU^T| = U|A|U^T$.*

Proof.

$$\begin{aligned}(U|A|U^T)^2 &= U|A||A|U^T \\ &= UA^TAU^T \quad (|A|^2 = A^TA) \\ &= UA^TU^TUAU^T \\ &= (UAU^T)^T(UAU^T).\end{aligned}$$

Since $U|A|U^T \succeq 0$, $U|A|U^T$ is the unique positive semi-definite square root of $(UAU^T)^T(UAU^T)$. □

Fact C.6. *Suppose $A = c\mathbf{1}_n\mathbf{1}_n^T + dI_n$. Then $|A| = c'\mathbf{1}_n\mathbf{1}_n^T + d'I_n$, where:*

$$c' = \frac{|cn + d| - |d|}{n}, \quad d' = |d|.$$

Proof. Let $U\Lambda U^T$ be an eigendecomposition of $\mathbf{1}_n\mathbf{1}_n^T$. Then $\Lambda = \text{diag}(n, 0, \ldots, 0)$. Now we write an eigendecomposition for A:

$$\begin{aligned}A &= c\mathbf{1}_n\mathbf{1}_n^T + dI_n \\ &= cU\Lambda U^T + dUU^T \\ &= U(c\Lambda + dI_n)U^T.\end{aligned} \tag{C.1}$$

By definition, then:

$$|A| = U|c\Lambda + dI_n|U^T,$$

which is of the same form as eq. (C.1), albeit with different constants. The result follows by solving the following for c' and d':

$$\text{diag}(|cn + d|, |d|, \ldots, |d|) = |c\Lambda + dI_n| = c'\Lambda + d'I_n = \text{diag}(c'n + d', d', \ldots, d').$$

□

Fact C.7. *Suppose $A = c\mathbf{1}_n\mathbf{1}_n^T + dI_n$, and $A_{ij} > 0$ for all $i, j \in [n]$. Then $|A|_{ij} > 0$ for all $i, j \in [n]$.*

Proof. We begin with the trivial cases: If $d \geq 0$, then $A \succeq 0$ and $A = |A|$. Also if $n = 1$, then A is scalar, and $|A|$ is the usual scalar absolute value.

Assume then that $d < 0$ and $n \geq 2$. Let $|A| = c'\mathbf{1}_n\mathbf{1}_n^T + d'I_n$ as defined in Fact C.6. Since all entries in A are positive, then $c > -d = |d|$. Consequently:

$$cn + d = cn - |d| > |d|n - |d| = |d|(n-1) \geq |d|$$

As a result, c' must be positive, since $|cn + d| = cn + d > |d|$. Since d' is also positive, every entry in $|A|$ is positive. □

Fact C.8. *For any two square matrices of equal dimension,* $\| \, |A| - |B| \, \|_F \leq \sqrt{2}\|A - B\|_F$.

Proof. See Bhatia (2013), Theorem VII.5.7 and eq. (VII.39). □

We recall our definition of the binary matrix operator ⊞.

Definition C.9. *Let* $A \in \mathbb{R}^{m\times m}, B \in \mathbb{R}^{n\times n}$. *Then:*

$$A \boxplus B = (A \otimes \mathbf{1}_n\mathbf{1}_n^T) + (\mathbf{1}_m\mathbf{1}_m^T \otimes B).$$

The operation ⊞ is similar to the more standard Kronecker sum $A \oplus B = (A \otimes I_n) + (I_m \otimes B)$, but with identity matrices replaced by $\mathbf{1}\mathbf{1}^T$. Fact C.10 below also resembles a property that the Kronecker sum satisfies, but replacing the matrix exponential with an element-wise exponential.

Fact C.10. *For two square matrices A and B,* $\exp(A \boxplus B) = \exp(A) \otimes \exp(B)$, *where* exp *is evaluated element-wise.*

Proof. Observe that the Kronecker product of two square matrices $A \in \mathbb{R}^{m\times m}$ and $B \in \mathbb{R}^{n\times n}$ may be written $A \otimes B = (A \otimes \mathbf{1}_n\mathbf{1}_n^T) \odot (\mathbf{1}_m\mathbf{1}_m^T \otimes B)$, where $\odot$ denotes the Hadamard product (i.e., element-wise multiplication). From here it follows that:

$$\begin{aligned}
\exp(A \boxplus B) &= \exp(A \otimes \mathbf{1}_n\mathbf{1}_n^T + \mathbf{1}_m\mathbf{1}_m^T \otimes B) \\
&= \exp(A \otimes \mathbf{1}_n\mathbf{1}_n^T) \odot \exp(\mathbf{1}_m\mathbf{1}_m^T \otimes B) \\
&= \left(\exp(A) \otimes \mathbf{1}_n\mathbf{1}_n^T\right) \odot \left(\mathbf{1}_m\mathbf{1}_m^T \otimes \exp(B)\right) \\
&= \exp(A) \otimes \exp(B).
\end{aligned}$$

□

In light of the Kronecker representation of $\exp(A \boxplus B)$, we review some facts about Kronecker products and inspect their matrix absolute values.

Fact C.11. *If $A = A^T$ and $B = B^T$, then $A \otimes B = (A \otimes B)^T$.*

Proof. By Horn and Johnson (1991, eq. 4.2.5), $(A \otimes B)^T = A^T \otimes B^T = A \otimes B$. □

Fact C.12. *Let $A = A^T, B = B^T$ with eigendecompositions $A = U\Lambda U^T, B = V\Psi V^T$. If $C = A \otimes B$, then:*

$$|C| = (U \otimes V)|\Lambda \otimes \Psi|(U \otimes V)^T = |A| \otimes |B|.$$

Proof. We begin by writing SVDs for A and B, namely:

$$A = U|\Lambda|(\text{sign}(\Lambda)U^T)$$
$$B = V|\Psi|(\text{sign}(\Psi)V^T),$$

where $\text{sign}(\cdot)$ is taken element-wise. It is easy to verify that $\text{sign}(\Lambda)U^T$ and $\text{sign}(\Psi)V^T$ are indeed orthogonal.

Armed with these decompositions, we may apply Horn and Johnson (1991, Theorem 4.2.15) to find an SVD for C:

$$\begin{aligned} C &= (U \otimes V)(|\Lambda| \otimes |\Psi|)(\text{sign}(\Lambda)U^T \otimes \text{sign}(\Psi)V^T) \\ &= (U \otimes V)|\Lambda \otimes \Psi|(\text{sign}(\Lambda)U^T \otimes \text{sign}(\Psi)V^T) \end{aligned}$$

Since $A = A^T$ and $B = B^T$, we have that $C = C^T$ (Fact C.11). Therefore:

$$\begin{aligned} |C| &= (U \otimes V)|\Lambda \otimes \Psi|(U \otimes V)^T \quad \text{(Fact C.3)} \\ &= (U \otimes V)(|\Lambda| \otimes |\Psi|)(U \otimes V)^T \\ &= (U|\Lambda| \otimes V|\Psi|) \otimes (U^T \otimes V^T) \\ &= (U|\Lambda|U^T) \otimes (V|\Psi|V^T) \\ &= |A| \otimes |B|. \end{aligned}$$

□

Finally, we give two useful facts about sums and permutations.

Fact C.13. *Let $x_1, \ldots, x_n \in \mathbb{R}$. Then for any $\sigma \in S_{[n]}$:*

$$\sum_{i=1}^{n} x_i x_{\sigma(i)} \leq \sum_{i=1}^{n} x_i^2.$$

Proof. This is an application of Cauchy–Schwarz in disguise:

$$\left(\sum_{i=1}^{n} x_i x_{\sigma(i)}\right)^2 \leq \left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} x_{\sigma(i)}^2\right) = \left(\sum_{i=1}^{n} x_i^2\right)^2.$$

The final statement comes by taking the square root of both sides. □

Fact C.14. *Let $A \in \mathbb{R}^{n\times n}$ such that $A \succeq 0$. Then for any $\sigma \in S_{[n]}$:*

$$\sum_{i=1}^{n} A_{i\sigma(i)} \leq \sum_{i=1}^{n} A_{ii}.$$

Moreover, if $\operatorname{rank}(A) = n$ *and* $\sigma \neq \mathrm{id}$*, the inequality is strict.*

Proof. Since $A \succeq 0$, let $A = XX^T$. Fix $\sigma \in S_{[n]}$. Then:

$$\begin{aligned}
\sum_{i=1}^{n} A_{i\sigma(i)} &= \sum_{i=1}^{n} e_i^T A e_{\sigma(i)} \\
&= \sum_{i=1}^{n} \langle X^T e_i, X^T e_{\sigma(i)} \rangle \\
&\overset{\text{ⓐ}}{\leq} \sum_{i=1}^{n} \|X^T e_i\| \|X^T e_{\sigma(i)}\| \quad \text{(Cauchy–Schwarz)} \\
&\leq \sum_{i=1}^{n} \|X^T e_i\|^2 \quad \text{(Fact C.13)} \\
&= \sum_{i=1}^{n} \langle X^T e_i, X^T e_i \rangle \\
&= \sum_{i=1}^{n} e_i^T A e_i = \sum_{i=1}^{n} A_{ii}.
\end{aligned}$$

If $\sigma \neq \mathrm{id}$, the inequality ⓐ is made strict when X has linearly independent rows, i.e., when A is full-rank. □

C.2 Proofs of Results

Representation Results. We prove that ACSBM can be represented as an SBM by explicitly constructing such a representation.

Proof of Proposition 3.3. Consider first the case when $M = 1$, i.e., $Z = Z_{*1}$. Every edge is an independent Bernoulli random variable whose probability depends on (θ_i, Z_{i1}) and (θ_j, Z_{j1}). It will be convenient to map these tuples to scalars. Let $\tau(k, \ell) = L_1(k-1) + \ell$, a bijection from $[K] \times [L_1]$ to $[KL_1]$. Let $\tilde{\theta}^{(1)} \in [KL_1]^n = (\tau(\theta_i, Z_{1i}))_{i=1}^n$. We will now write the edge probabilities in terms of these new scalar quantities. It can be shown (if a bit tediously) that:

$$\begin{aligned}\mathbf{P}(Y_{ij} = 1 \mid \tilde{\theta}_i^{(1)} = t_1, \tilde{\theta}_j^{(1)} = t_2) &= g^{-1}\left(\left[B \otimes \mathbf{1}_{L_1}\mathbf{1}_{L_1}^T + \mathbf{1}_K\mathbf{1}_K^T \otimes \beta_1 I_{L_1} \right]_{t_1 t_2} \right) \\ &= \left[g^{-1}(B \boxplus \beta_1 I_{L_1}) \right]_{t_1 t_2},\end{aligned}$$

where g^{-1} is taken element-wise in the final line. This is precisely the form of the SBM given in Definition 3.1. Thus when $M = 1$, we can say Y is equal to an SBM with $\tilde{L} = KL_1$ communities, $\tilde{\theta} = L_1(\theta - \mathbf{1}_n) + Z_{*1}$, and edge probabilities $\tilde{B} = g^{-1}(B \boxplus \beta_1 I_{L_1})$.

The case when $M \geq 2$ follows inductively. Let $Y_1 \sim \text{ACSBM}(\theta, B, Z_1, \beta_1, g) \overset{D}{=} \text{SBM}(\tilde{\theta}^{(1)}, \tilde{B}^{(1)})$. Define $Y_2 = \text{ACSBM}(\theta, B, [Z_1 \mid Z_2], (\beta_1, \beta_2)^T, g)$. This network is equal in distribution to $Y_2' \sim \text{ACSBM}(\tilde{\theta}^{(1)}, g(\tilde{B}^{(1)}), Z_2, \beta_2, g)$. By the $M = 1$ case above, these networks are equal in distribution to an SBM with KL_1L_2 communities:

$$\tilde{\theta}^{(2)} = L_2(\tilde{\theta}^{(1)} - \mathbf{1}_n) + Z_{*2} = L_2(L_1(\theta - \mathbf{1}_n) + Z_{*1} - \mathbf{1}_n) + Z_{*2}$$

and edge probabilities:

$$g^{-1}\left(g(\tilde{B}^{(1)}) \boxplus \beta_2 I_{L_2} \right) = g^{-1}(B \boxplus \beta_1 I_{L_1} \boxplus \beta_2 I_{L_2}),$$

where once again, g and g^{-1} are element-wise.

Proceed inductively to find the forms of $Y_3, \ldots, Y_M$, defined analogously to Y_2, so that $Y \overset{D}{=} Y_M$. □

The gRDPG representation now follows immediately as a corollary.

Proof of Proposition 3.8. By Proposition 3.3, we may represent Y as an SBM, i.e., $Y \overset{D}{=} \text{SBM}(\tilde{\theta}, \tilde{B})$. The ability to represent an SBM as a gRDPG using latent positions derived from spectral decom-

position is a well established practice in the gRDPG literature, e.g., Rubin-Delanchy et al. (2022, Section 2.1). Thus Proposition 3.8 follows as a corollary to Proposition 3.3. □

Consistency of Part 1. Consistency of Part 1 of the algorithm was stated in Theorem 3.11, proven here.

Proof of Theorem 3.11. By Lemma 3.10, we know that:

$$\max_{i\in[n]} \|Q\hat{X}_i - X_{\tilde{B}}(\theta_i, Z_i)\|_2 = O_P\left(\frac{\log^c n}{\sqrt{n}}\right)$$

for some sequence of matrices $Q \in \mathbb{O}(p,q)$. We might prefer a statement in terms of $\hat{X}_i$, rather than $Q\hat{X}_i$, which we can make as follows:

$$\max_{i\in[n]} \|\hat{X}_i - QX_{\tilde{B}}(\theta_i, Z_i)\|_2 \leq \|Q^{-1}\|_2 \left(\max_{i\in[n]} \|Q\hat{X}_i - X_{\tilde{B}}(\theta_i, Z_i)\|_2\right).$$

We have seemingly done little here but move the troublesome Q and impose an additional nuisance term. However, Rubin-Delanchy et al. (2022, Lemma 5) states a key result: $\|Q\|_2$ and $\|Q^{-1}\|_2$ are bounded almost surely. This allows us to eliminate the nuisance term:

$$\max_{i\in[n]} \|\hat{X}_i - QX_{\tilde{B}}(\theta_i, Z_i)\|_2 = O_P\left(\frac{\log^c n}{\sqrt{n}}\right).$$

We still have to grapple with $QX_{\tilde{B}}$. Observe that for z fixed, the canonical latent positions $X_{\tilde{B}}(1,z), \ldots, X_{\tilde{B}}(K,z)$ are distinct by construction. Since Q is full-rank, this also applies to $QX_{\tilde{B}}(1,z), \ldots, QX_{\tilde{B}}(K,z)$. Moreover, in light of the bounded spectral norms of Q and Q^{-1}, which bound the singular values of Q in an interval away from zero, the asymptotic distortion of distances is limited. In particular, $\|Q(X_{\tilde{B}}(k_1,z) - X_{\tilde{B}}(k_2,z))\|_2 = \Theta(\sqrt{\alpha_n})$ almost surely. Combining these facts yields the result, as follows.

Let $\mathcal{B}(x,r)$ denote a ball centered at x with radius r. From our argument above, there exists a sequence of radii $r = O_P(\log^c n/\sqrt{n})$ such that $\hat{X}_i \in \mathcal{B}(QX_{\tilde{B}}(\theta_i, z), r)$ for all $i \in \mathcal{I}_z$. Since $\|Q(X_{\tilde{B}}(k_1,z) - X_{\tilde{B}}(k_2,z))\|_2$ scales with $\sqrt{\alpha_n} = \omega(\log^{2c} n/\sqrt{n})$, these balls shrink in size faster than they converge to the origin. More concretely, let $\mathcal{B}_{k,z} = \mathcal{B}(QX_{\tilde{B}}(k,z), r)$ for $k \in [K]$. Then for any $k_1, k_2 \in [K]$:

$$\mathbf{P}(\mathcal{B}_{k_1,z} \cap \mathcal{B}_{k_2,z} = \emptyset) = \mathbf{P}\left(r < \frac{1}{2}\|QX_{\tilde{B}}(k_1,z) - QX_{\tilde{B}}(k_2,z)\|_2\right) \to 1,$$

since $\|QX_{\tilde{B}}(k_1, z) - QX_{\tilde{B}}(k_2, z)\|_2 = \Theta(\sqrt{\alpha_n})$ almost surely, and $r = o_P(\sqrt{\alpha_n})$. □

Consistency of Part 2. Consistency of Part 2 of the algorithm was stated in Theorem 3.12.

Proof of Theorem 3.12. Suppose $Y_{gen} \sim \text{SBM}(\tilde{\theta}, B_{gen})$ for some symmetric matrix $B_{gen} \in \mathbb{R}^{K\tilde{L} \times K\tilde{L}}$. This model is more general than $Y \sim \text{SBM}(\tilde{\theta}, \tilde{B})$. Suppose we have a perfect estimate of $\tilde{\theta}$ (up to a permutation), and we wish to estimate B_{gen}. In this case, the natural approach to estimating B_{gen} via the empirical density of each block is precisely the maximum likelihood estimator, which has been well-studied (e.g., Bickel et al., 2013).

Under the theorem hypothesis, we have indeed recovered $\tilde{\theta}$ up to a permutation of labels. This is true since $\tilde{\theta}((\tau_{z_i} \circ \hat{\theta}_{z_i})(i)), z_i) = \tilde{\theta}_i$ for all i, and the function $\tilde{\theta}(\cdot, \cdot)$ is a bijection. Let $\tau \in S_{[K\tilde{L}]}$ denote this permutation, and let T denote the corresponding permutation matrix. Then $T^{-1}\hat{\tilde{B}}T$ is the maximum likelihood estimator for a model $Y_{gen} \sim \text{SBM}(\tilde{\theta}, B_{gen})$, and so we may apply the maximum likelihood results of Bickel et al. (2013, Lemma 1) or, more conveniently, Tang et al. (2022, Theorem 1). Per these results, we can say that for any $k_1, k_2 \in [K\tilde{L}]$:

$$n\alpha_n^{-1/2}\left((T^{-1}\hat{\tilde{B}}T)_{k_1k_2} - \tilde{B}_{k_1k_2}\right) \xrightarrow{D} \mathcal{N}(0, v_{k_1k_2}),$$

where $\xrightarrow{D} \mathcal{N}(\cdot, \cdot)$ denotes convergence in distribution to the normal distribution, and $v_{k_1k_2} > 0$ is a constant depending on k_1 and k_2. In other words:

$$(T^{-1}\hat{\tilde{B}}T)_{k_1k_2} - \tilde{B}_{k_1k_2} = O_P\left(\frac{\sqrt{\alpha_n}}{n}\right).$$

Since $\tilde{B}$ scales with α_n, we rewrite this to be in terms of the constant quantity $\alpha_n^{-1}\tilde{B}$:

$$\alpha_n^{-1}\left((T^{-1}\hat{\tilde{B}}T)_{k_1k_2} - \tilde{B}_{k_1k_2}\right) = O_P\left(\frac{1}{n\sqrt{\alpha_n}}\right) = o_P\left(\frac{1}{\sqrt{n\log^c n}}\right).$$

Since K and $\tilde{L}$ are kept constant in n, these entrywise bounds may be taken as a bound for the Frobenius norm, $\|T^{-1}\hat{\tilde{B}}T - \tilde{B}\|_F$. Moreover, since the Frobenius norm is unitarily invariant, we may write:

$$\|\hat{\tilde{B}} - T\tilde{B}T^{-1}\|_F = o_P\left(\frac{1}{\sqrt{n\log^c n}}\right).$$

□

Consistency of Part 3. We first show that the matching problem selects the appropriate

permutations in the absence of estimation error, i.e., when applied to the true latent positions $X_{\tilde{B}}$. Note that the role of the permutation σ in Theorem C.15 below differs slightly from its role in Algorithm 3. In the algorithm, there is an unknown permutation that we are looking to reverse for each choice of z; in the theorem below, there is no such permutation, so the correct choice of σ is the identity permutation.

Theorem C.15. *Assume Y from the setting of Section 3.4. Let $X_{\tilde{B}}$ as in Proposition 3.3. For any fixed $z \in [L_1] \times \cdots \times [L_M]$:*

$$\arg\min_{\sigma \in S_{[K]}} \sum_{k=1}^{K} \|X_{\tilde{B}}(\sigma(k), z) - X_{\tilde{B}}(k, \mathbf{1}_M)\|_2^2 = \mathrm{id}. \tag{C.2}$$

Moreover, if $\exp(B)$ is full-rank, $\sigma = \mathrm{id}$ is the unique minimizer.

Proof. To simplify notation for the proof, let $x_{kz} = X_{\tilde{B}}(k, z)$. We begin by unpacking the squared norm:

$$\begin{aligned}
\sum_{k=1}^{K} \|x_{\sigma(k)z} - x_{k\mathbf{1}}\|_2^2 &= \sum_{k=1}^{K} \langle x_{\sigma(k)z} - x_{k\mathbf{1}}, x_{\sigma(k)z} - x_{k\mathbf{1}} \rangle \\
&= \sum_{k=1}^{K} \left(\langle x_{\sigma(k)z}, x_{\sigma(k)z} \rangle + \langle x_{k\mathbf{1}}, x_{k\mathbf{1}} \rangle - 2\langle x_{\sigma(k)z}, x_{k\mathbf{1}} \rangle \right) \\
&= \sum_{k=1}^{K} \langle x_{kz}, x_{kz} \rangle + \sum_{k=1}^{K} \langle x_{k\mathbf{1}}, x_{k\mathbf{1}} \rangle - 2 \sum_{k=1}^{K} \langle x_{\sigma(k)z}, x_{k\mathbf{1}} \rangle
\end{aligned}$$

Since only the final sum depends on σ, the optimization problem (C.2) is equivalent to finding:

$$\arg\max_{\sigma \in S_{[K]}} \sum_{k=1}^{K} \langle x_{\sigma(k)z}, x_{k\mathbf{1}} \rangle.$$

Fix $z \in [L_1] \times \cdots \times [L_M]$, and let $\tilde{B}$ as in Proposition 3.3. Next, we will assemble yet another matrix. For any $k_1, k_2 \in [K]$, let $Q_{k_1 k_2} = \langle x_{k_1 z}, x_{k_2 \mathbf{1}} \rangle$. If we can show that $Q \succeq 0$, the result will follow from Fact C.14. This is our plan. Observe that:

$$\langle x_{k_1 z}, x_{k_2 \mathbf{1}} \rangle_{pq} = \tilde{B}_{\tilde{\theta}(k_1, z), \tilde{\theta}(k_2, \mathbf{1})},$$

where (p, q) is the signature of the gRDPG corresponding to Y. Following from Fact C.4, the inner products that form the entries of Q can be found in $|\tilde{B}|$, i.e.:

$$Q_{k_1 k_2} = \langle x_{k_1 z}, x_{k_2 \mathbf{1}} \rangle = |\tilde{B}|_{\tilde{\theta}(k_1, z), \tilde{\theta}(k_2, \mathbf{1})}.$$

Since $g = \log$, by Fact C.10, we can write $\tilde{B}$ like so:

$$\tilde{B} = \exp(B) \otimes \exp(\beta_1 I_{L_1}) \otimes \cdots \otimes \exp(\beta_M I_{L_M}).$$

Lemma C.12 gives the convenient form of $|\tilde{B}|$:

$$|\tilde{B}| = |\exp(B)| \otimes |\exp(\beta_1 I_{L_1})| \otimes \cdots \otimes |\exp(\beta_M I_{L_M})|.$$

In particular, this means:

$$\begin{aligned} Q_{k_1 k_2} &= |\tilde{B}|_{\tilde{\theta}(k_1,z),\tilde{\theta}(k_2,\mathbf{1})} \\ &= |\exp(B)|_{k_1 k_2} \left[\, |\exp(\beta_1 I_{L_1})| \otimes \cdots \otimes |\exp(\beta_M I_{L_M})| \,\right]_{\tilde{\theta}(1,z),1} \\ &= c_z \, |\exp(B)|_{k_1 k_2}, \end{aligned}$$

where $c_z = \left[\, |\exp(\beta_1 I_{L_1})| \otimes \cdots \otimes |\exp(\beta_M I_{L_M})| \,\right]_{\tilde{\theta}(1,z),1}$ is a strictly positive constant. This follows from Fact C.7, which says that each of the $|\exp(\beta_m I_{L_m})|$ matrices have positive entries. Since $|\exp(B)| \geq 0$ by construction, we have then that $Q \succeq 0$. Moreover, when $\exp(B)$ is full-rank, $Q \succ 0$.

Applying Fact C.14, we have that $\sigma = \mathrm{id}$ is a solution to our optimization problem; moreover, it is the unique solution when $\exp(B)$ is full-rank. □

Next, we show that the estimation error due to use of $\hat{X}_{\tilde{B}}$ in place of $X_{\tilde{B}}$ vanishes asymptotically. Note that relabeling permutations appear here.

Lemma C.16. *Assume the conditions of Theorem 3.13 hold. Let $X_{\tilde{B}}$ as in Proposition 3.3 and $\hat{X}_{\tilde{B}}$ as in Algorithm 3. For any fixed $z \in [L_1] \times \cdots \times [L_M]$, let:*

$$\begin{aligned} \hat{L}_z(\sigma) &= \sum_{k=1}^{K} \|\hat{X}_{\tilde{B}}(\sigma(k), z) - \hat{X}_{\tilde{B}}(k, \mathbf{1}_M)\|_2^2 \\ L_z(\sigma) &= \sum_{k=1}^{K} \|X_{\tilde{B}}((\sigma \circ \tau_z)(k), z) - \hat{X}_{\tilde{B}}(\tau_{\mathbf{1}_M}(k), \mathbf{1}_M)\|_2^2. \end{aligned}$$

Then for any $\sigma_1, \sigma_2 \in S_{[K]}$:

$$\alpha_n^{-1}(\hat{L}_z(\sigma_1) - \hat{L}_z(\sigma_2)) = \alpha_n^{-1}(L_z(\sigma_1) - L_z(\sigma_2)) + o_P\left(\frac{1}{\sqrt{n \log^c n}}\right).$$

Proof. By an argument similar to the proof of Theorem C.15, we observe that:

$$\hat{L}_z(\sigma) = \hat{c}_z - 2\sum_{k=1}^{K}\langle \hat{X}_{\tilde{B}}(\sigma(k), z),\ \hat{X}_{\tilde{B}}(k, \mathbf{1}_M)\rangle$$
$$L_z(\sigma) = c_z - 2\sum_{k=1}^{K}\langle X_{\tilde{B}}((\sigma \circ \tau_z)(k), z),\ \hat{X}_{\tilde{B}}(\tau_{\mathbf{1}_M}(k), \mathbf{1}_M)\rangle$$

for some constants $\hat{c}_z$ and c_z. Moreover, continuing to extend the arguments from the proof of Theorem C.15, we have:

$$\begin{aligned}
\langle \hat{X}_{\tilde{B}}(\sigma(k), z),\ \hat{X}_{\tilde{B}}(k, \mathbf{1}_M)\rangle &= |\hat{\tilde{B}}|_{\tilde{\theta}(\sigma(k),z),\tilde{\theta}(k,\mathbf{1})} \\
\langle X_{\tilde{B}}((\sigma \circ \tau_z)(k), z),\ \hat{X}_{\tilde{B}}(\tau_{\mathbf{1}_M}(k), \mathbf{1}_M)\rangle &= |\tilde{B}|_{\tilde{\theta}((\sigma\circ\tau_z)(k),z),\tilde{\theta}(\tau_{\mathbf{1}_M}(k),\mathbf{1})} \\
&= (T|\tilde{B}|T^{-1})_{\tilde{\theta}(\sigma(k),z),\tilde{\theta}(k,\mathbf{1})} \\
&= |T\tilde{B}T^{-1}|_{\tilde{\theta}(\sigma(k),z),\tilde{\theta}(k,\mathbf{1})},
\end{aligned}$$

where T is the permutation matrix from Theorem 3.12. Note that the last line follows from Fact C.5. Therefore:

$$\begin{aligned}
&\hat{L}_z(\sigma_1) - \hat{L}_z(\sigma_2) - (L_z(\sigma_1) - L_z(\sigma_2)) \\
&\quad= -2\sum_{k=1}^{K}|\hat{\tilde{B}}|_{\tilde{\theta}(\sigma_1(k),z),\tilde{\theta}(k,\mathbf{1})} + 2\sum_{k=1}^{K}|\hat{\tilde{B}}|_{\tilde{\theta}(\sigma_2(k),z),\tilde{\theta}(k,\mathbf{1})} \\
&\qquad+ 2\sum_{k=1}^{K}|T\tilde{B}T^{-1}|_{\tilde{\theta}(\sigma_1(k),z),\tilde{\theta}(k,\mathbf{1})} - 2\sum_{k=1}^{K}|T\tilde{B}T^{-1}|_{\tilde{\theta}(\sigma_2(k),z),\tilde{\theta}(k,\mathbf{1})} \\
&\quad= 2\sum_{k=1}^{K}\left(|\hat{\tilde{B}}|_{\tilde{\theta}(\sigma_2(k),z),\tilde{\theta}(k,\mathbf{1})} - |T\tilde{B}T^{-1}|_{\tilde{\theta}(\sigma_2(k),z),\tilde{\theta}(k,\mathbf{1})}\right) \\
&\qquad- 2\sum_{k=1}^{K}\left(|\hat{\tilde{B}}|_{\tilde{\theta}(\sigma_1(k),z),\tilde{\theta}(k,\mathbf{1})} - |T\tilde{B}T^{-1}|_{\tilde{\theta}(\sigma_1(k),z),\tilde{\theta}(k,\mathbf{1})}\right).
\end{aligned}$$

Observe that the final expression consists of $2K$ terms of the form $2(|\hat{\tilde{B}}|_{ij} - |T\tilde{B}T^{-1}|_{ij})$. Combining Theorem 3.12 and Fact C.8, we know that:

$$\alpha_n^{-1}\|\ |\hat{\tilde{B}}| - |T\tilde{B}T^{-1}|\ \|_F = o_P\left(\frac{1}{\sqrt{n\log^c n}}\right),$$

from which we claim a bound on the entrywise error for any $i, j \in [K\tilde{L}]$:

$$\alpha_n^{-1}(|\hat{\tilde{B}}|_{ij} - |T\tilde{B}T^{-1}|_{ij}) = o_P\left(\frac{1}{\sqrt{n\log^c n}}\right).$$

Summarizing, then, we have:

$$\alpha_n^{-1}\left(\hat{L}_z(\sigma_1) - \hat{L}_z(\sigma_2) - (L_z(\sigma_1) - L_z(\sigma_2))\right) = 4K \cdot o_P\left(\frac{1}{\sqrt{n\log^c n}}\right).$$

Since K is constant, the final result follows by simple rearrangement. □

For completeness, we end with a formal proof of Theorem 3.13.

Proof of Theorem 3.13. Let $\hat{L}_z : S_{[K]} \to \mathbb{R}$ and $L_z : S_{[K]} \to \mathbb{R}$ as in the statement of Lemma C.16. We first rewrite the result of Theorem C.15 in a permuted order. For any fixed z:

$$\begin{aligned}
&\arg\min_{\sigma\in S_{[K]}} L_z(\sigma) \\
&= \arg\min_{\sigma\in S_{[K]}} \sum_{k=1}^{K} \|X_{\tilde{B}}\left((\sigma\circ\tau_z)(k), z\right) - X_{\tilde{B}}\left(\tau_{\mathbf{1}_M}(k), \mathbf{1}_M\right)\|_2^2 \\
&= \tau_{\mathbf{1}_M}\circ\tau_z^{-1}.
\end{aligned}$$

This follows from the commutativity of the sum and the fact that $S_{[K]}$ is closed under composition. In other words, we may think of the sum as going in order of $\tau_{\mathbf{1}_M}(1), \ldots, \tau_{\mathbf{1}_M}(K)$ and minimizing over $\sigma \circ \tau_z \in S_{[K]}$ instead, if we prefer, in which case recovering the identity permutation is equivalent to recovering $\sigma \circ \tau_z = \tau_{\mathbf{1}_M}$.

For each z, let $\sigma_z^* = \tau_{\mathbf{1}_M} \circ \tau_z^{-1}$ denote the optimal permutation, and let:

$$\begin{aligned}
a_z &= L_z(\sigma_z^*), \\
b_z &= \arg\min_{\sigma\neq\sigma_z^*} L_z(\sigma), \text{ and} \\
\Delta_z &= b_z - a_z,
\end{aligned}$$

so that Δ_z denotes the gap between the optimal and second-best permutation. Let $\Delta_0 = \min_z \Delta_z$. Since $X_{\tilde{B}}$ scales with $\sqrt{\alpha_n}$, $L_z(\cdot)$ scales with α_n, and the quantity $\alpha_n^{-1}\Delta_0$ is constant. By assumption ***(A2)***, we may further assume $\Delta_0 > 0$.

By Lemma C.16, we have that for any permutation $\sigma \in S_{[K]}$:

$$\alpha_n^{-1}(\hat{L}_z(\sigma) - \hat{L}_z(\sigma_z^*)) = \alpha_n^{-1}(L_z(\sigma) - L_z(\sigma_z^*)) + o_P\left(\frac{1}{\sqrt{n \log^c n}}\right).$$

We would like these error terms to be less than $\alpha_n^{-1}\Delta_0/2$ for all z. Since $\alpha_n^{-1}\Delta_0/2$ is constant, this happens with high probability for sufficiently large n. In this case, we have:

$$\hat{\sigma}_z = \arg\min_{\sigma \in S_{[K]}} \hat{L}_z(\sigma) = \arg\min_{\sigma \in S_{[K]}} L_z(\sigma) = \sigma_z^* = \tau_{\mathbf{1}_M} \circ \tau_z^{-1}.$$

Consequently, for all $i \in \mathcal{I}_z$, since $\hat{\theta}_z(i) = \tau_z(\theta_i)$, we have our desired result:

$$\hat{\sigma}_z(\hat{\theta}_z(i)) = \tau_{\mathbf{1}_M}(\tau_z^{-1}(\tau_z(\theta_i))) = \tau_{\mathbf{1}_M}(\theta_i).$$

□

Bibliography

Abbe, Emmanuel (2018). "Community Detection and Stochastic Block Models: Recent Developments". In: *Journal of Machine Learning Research* 18.177, pp. 1–86.

Abbe, Emmanuel, Jianqing Fan, and Kaizheng Wang (2022). *An ℓ_p theory of PCA and spectral clustering*. DOI: 10.1214/22-AOS2196.

Abbe, Emmanuel, Jianqing Fan, Kaizheng Wang, and Yiqiao Zhong (2020). "Entrywise eigenvector analysis of random matrices with low expected rank". In: *Annals of Statistics* 48.3, pp. 1452–1474.

Adamic, Lada A and Natalie Glance (2005). "The political blogosphere and the 2004 US election: divided they blog". In: *Proceedings of the 3rd international workshop on Link discovery*, pp. 36–43.

Agterberg, Joshua, Minh Tang, and Carey E Priebe (2020). "On two distinct sources of nonidentifiability in latent position random graph models". In: *arXiv preprint arXiv:2003.14250*.

Airoldi, Edoardo Maria, David M Blei, Stephen E Fienberg, and Eric P Xing (2008). "Mixed membership stochastic blockmodels". In: *Journal of Machine Learning Research*.

Akcora, Cuneyt G., Yitao Li, Yulia R. Gel, and Murat Kantarcioglu (July 2020). "BitcoinHeist: Topological Data Analysis for Ransomware Prediction on the Bitcoin Blockchain". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. Ed. by Christian Bessiere. Special Track on AI in FinTech. International Joint Conferences on Artificial Intelligence Organization, pp. 4439–4445. DOI: 10.24963/ijcai.2020/612.

Andris, Clio, David Lee, Marcus J Hamilton, Mauro Martino, Christian E Gunning, and John Armistead Selden (2015). "The rise of partisanship and super-cooperators in the US House of Representatives". In: *PloS one* 10.4, e0123507.

Athreya, Avanti, Donniell E Fishkind, et al. (2017). "Statistical inference on random dot product graphs: a survey". In: *The Journal of Machine Learning Research* 18.1, pp. 8393–8484.

Athreya, Avanti, Carey E Priebe, Minh Tang, Vince Lyzinski, David J Marchette, and Daniel L Sussman (2016). "A limit theorem for scaled eigenvectors of random dot product graphs". In: *Sankhya A* 78.1, pp. 1–18.

Bar-Yossef, Ziv, TS Jayram, Ravi Kumar, D Sivakumar, and Luca Trevisan (2002). "Counting distinct elements in a data stream". In: *International Workshop on Randomization and Approximation Techniques in Computer Science*. Springer, pp. 1–10.

Bhatia, Rajendra (2013). *Matrix analysis*. Vol. 169. Springer Science & Business Media.

Bickel, Peter, David Choi, Xiangyu Chang, and Hai Zhang (2013). "Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels". In: *The Annals of Statistics* 41.4, pp. 1922–1943.

Binkiewicz, Norbert, Joshua T Vogelstein, and Karl Rohe (2017). "Covariate-assisted spectral clustering". In: *Biometrika* 104.2, pp. 361–377.

Broder, Andrei and Michael Mitzenmacher (2004). "Network applications of bloom filters: A survey". In: *Internet mathematics* 1.4, pp. 485–509.

Chen, Hongjie et al. (2023). "Private estimation algorithms for stochastic block models and mixture models". In: *arXiv preprint arXiv:2301.04822*.

Chen, Yudong, Xiaodong Li, and Jiaming Xu (2018). "Convexified modularity maximization for degree-corrected stochastic block models". In: *The Annals of Statistics* 46.4, pp. 1573–1602.

Choi, David S, Patrick J Wolfe, and Edoardo M Airoldi (2012). "Stochastic blockmodels with a growing number of classes". In: *Biometrika* 99.2, pp. 273–284.

Choi, Seung Geol, Dana Dachman-Soled, Mukul Kulkarni, and Arkady Yerukhimovich (2020). "Differentially-private multi-party sketching for large-scale statistics". In: *Cryptology ePrint Archive*.

Clifford, Peter and Ioana A Cosma (2012). "A statistical analysis of probabilistic counting algorithms". In: *Scandinavian Journal of Statistics* 39.1, pp. 1–14.

Collet, Yann (2022). *xxHash - Extremely fast non-cryptographic hash algorithm*. `https://cyan4973.github.io/xxHash/`. Accessed: 2022-11-16.

Cormode, Graham and Ke Yi (2020). *Small summaries for big data*. Cambridge University Press.

Desfontaines, Damien, Andreas Lochbihler, and David Basin (Apr. 2019). "Cardinality Estimators do not Preserve Privacy". In: *Proceedings on Privacy Enhancing Technologies* 2019.2, pp. 26–46. DOI: 10.2478/popets-2019-0018.

Deshpande, Yash, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel (2018). "Contextual stochastic block models". In: *Advances in Neural Information Processing Systems* 31.

Dickens, Charlie, Justin Thaler, and Daniel Ting (2022). "Order-Invariant Cardinality Estimators Are Differentially Private". In: *Advances in Neural Information Processing Systems*.

Duchi, John C, Michael I Jordan, and Martin J Wainwright (2013). "Local privacy and statistical minimax rates". In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, pp. 429–438.

Durand, Marianne and Philippe Flajolet (2003). "Loglog counting of large cardinalities". In: *European Symposium on Algorithms*. Springer, pp. 605–617.

Dwork, Cynthia (2006). "Differential Privacy". In: *Automata, Languages and Programming*. Ed. by David Hutchison et al. Vol. 4052. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–12. (Visited on 12/12/2019).

— (2008). "Differential privacy: A survey of results". In: *International conference on theory and applications of models of computation*. Springer, pp. 1–19.

Dwork, Cynthia, Frank McSherry, Kobbi Nissim, and Adam Smith (2006). "Calibrating noise to sensitivity in private data analysis". In: *Theory of cryptography conference*. Springer, pp. 265–284.

Edmonds, Jack and Richard M Karp (1972). "Theoretical improvements in algorithmic efficiency for network flow problems". In: *Journal of the ACM (JACM)* 19.2, pp. 248–264.

Erlingsson, Úlfar, Vasyl Pihur, and Aleksandra Korolova (2014). "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response". In: *ACM SIGSAC Conference on Computer*

and Communications Security. ACM, pp. 1054–1067. DOI: 10.1145/2660267.2660348. URL: https://doi.org/10.1145/2660267.2660348.

Ertl, Otmar (2017). "New cardinality estimation algorithms for HyperLogLog sketches". In: *arXiv preprint arXiv:1702.01284*.

Flajolet, Philippe, Éric Fusy, Olivier Gandouet, and Frédéric Meunier (2007). "Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm". In: *Discrete Mathematics and Theoretical Computer Science*. Discrete Mathematics and Theoretical Computer Science, pp. 137–156.

Flajolet, Philippe and G Nigel Martin (1985). "Probabilistic counting algorithms for data base applications". In: *Journal of computer and system sciences* 31.2, pp. 182–209.

Giroire, Frédéric (2009). "Order statistics and estimating cardinalities of massive data sets". In: *Discrete Applied Mathematics* 157.2, pp. 406–427.

Goldsmith-Pinkham, Paul and Guido W Imbens (2013). "Social networks and the identification of peer effects". In: *Journal of Business & Economic Statistics* 31.3, pp. 253–264.

Goodreau, Steven M, James A Kitts, and Martina Morris (2009). "Birds of a feather, or friend of a friend? Using exponential random graph models to investigate adolescent social networks". In: *Demography* 46.1, pp. 103–125.

Handcock, Mark S, Adrian E Raftery, and Jeremy M Tantrum (2007). "Model-based clustering for social networks". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 170.2, pp. 301–354.

Hansell, Stephen (1984). "Cooperative groups, weak ties, and the integration of peer friendships". In: *Social Psychology Quarterly*, pp. 316–328.

Hehir, Jonathan (Jan. 2022a). *jonhehir/congress-voting-networks: v2022.01*. Version v2022.01. DOI: 10.5281/zenodo.5838420. URL: https://zenodo.org/record/5838420.

— (Sept. 2022b). *private-spectral-clustering: Spectral Clustering with Edge-Flip Differential Privacy*. Version v20220920-jpc. DOI: 10.5281/zenodo.7098162. URL: https://zenodo.org/record/7098162.

— (Feb. 2023). *acsbm: Spectral Clustering on the Additive-Covariate Stochastic Block Model*. Version v2023.02. DOI: 10.5281/zenodo.7601055. URL: https://zenodo.org/record/7601055.

Hehir, Jonathan, Xiaoyue Niu, and Aleksandra Slavkovic (2022). "Perfect Spectral Clustering with Discrete Covariates". In: *arXiv preprint arXiv:2205.08047*.

Hehir, Jonathan, Aleksandra Slavkovic, and Xiaoyue Niu (Nov. 2022). "Consistent Spectral Clustering of Network Block Models under Local Differential Privacy". In: *Journal of Privacy and Confidentiality* 12.2. DOI: 10.29012/jpc.811.

Hehir, Jonathan, Daniel Ting, and Graham Cormode (2023). "Sketch-Flip-Merge: Mergeable Sketches for Private Distinct Counting". In: *arXiv preprint arXiv:2302.02056*.

Henry, Adam Douglas, Paweł Prałat, and Cun-Quan Zhang (2011). "Emergence of segregation in evolving social networks". In: *Proceedings of the National Academy of Sciences* 108.21, pp. 8605–8610.

Heule, Stefan, Marc Nunkesser, and Alexander Hall (2013). "Hyperloglog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm". In: *Proceedings of the 16th International Conference on Extending Database Technology*, pp. 683–692.

Hoff, Peter (2007). "Modeling homophily and stochastic equivalence in symmetric relational data". In: *Advances in neural information processing systems* 20.

Holland, Paul W, Kathryn Blackmond Laskey, and Samuel Leinhardt (1983). "Stochastic blockmodels: First steps". In: *Social networks* 5.2, pp. 109–137.

Horn, Roger A. and Charles R. Johnson (1991). *Topics in Matrix Analysis*. Cambridge University Press.

— (2012). *Matrix Analysis*. Cambridge University Press.

Huang, Sihan and Yang Feng (2018). "Pairwise covariates-adjusted block model for community detection". In: *arXiv preprint arXiv:1807.03469*.

Huber, Gregory A and Neil Malhotra (2017). "Political homophily in social relationships: Evidence from online dating behavior". In: *The Journal of Politics* 79.1, pp. 269–283.

Ibarra, Herminia (1992). "Homophily and differential returns: Sex differences in network structure and access in an advertising firm". In: *Administrative science quarterly*, pp. 422–447.

Imola, Jacob, Takao Murakami, and Kamalika Chaudhuri (2021). "Locally differentially private analysis of graph statistics". In: *30th USENIX Security Symposium (USENIX Security 21)*.

Jin, Jiashun (2015). "Fast community detection by SCORE". In: *The Annals of Statistics* 43.1, pp. 57–89.

Joseph, Antony and Bin Yu (2016). "Impact of regularization on spectral clustering". In: *The Annals of Statistics* 44.4, pp. 1765–1791.

Karrer, Brian and Mark EJ Newman (2011). "Stochastic blockmodels and community structure in networks". In: *Physical review E* 83.1, p. 016107.

Karwa, Vishesh, Pavel N. Krivitsky, and Aleksandra B. Slavković (Apr. 2017). "Sharing social network data: differentially private estimation of exponential family random-graph models". en. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 66.3, pp. 481–500. (Visited on 10/02/2019).

Karwa, Vishesh and Aleksandra Slavković (Feb. 2016). "Inference using noisy degrees: Differentially private β-model and synthetic graphs". en. In: *The Annals of Statistics* 44.1, pp. 87–112. (Visited on 10/02/2019).

Kasiviswanathan, Shiva Prasad, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith (2013). "Analyzing Graphs with Node Differential Privacy". In: *Theory of Cryptography*. Ed. by Amit Sahai. Vol. 7785. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 457–476. (Visited on 11/12/2019).

Kolchin, V.F., B.A. Sevast'janov, V.P. Christ'Yakov, V.P. Čistjakov, and A.V. Balakrishnan (1978). *Random Allocations*. A Halsted Press book. V. H. Winston. ISBN: 9780470993941. URL: https://books.google.com/books?id=jxjvAAAAMAAJ.

Kreuter, Benjamin, Craig William Wright, Evgeny Sergeevich Skvortsov, Raimundo Mirisola, and Yao Wang (2020). "Privacy-preserving secure cardinality and frequency estimation". In.

Lang, Kevin J (2017). "Back to the future: an even more nearly optimal cardinality estimation algorithm". In: *arXiv preprint arXiv:1708.06839*.

Lee, Youjin and Elizabeth L Ogburn (2021). "Network dependence can lead to spurious associations and invalid inference". In: *Journal of the American Statistical Association* 116.535, pp. 1060–1074.

Lei, Jing and Alessandro Rinaldo (2015). "Consistency of spectral clustering in stochastic block models". In: *Annals of Statistics* 43.1, pp. 215–237.

Lewis, Jeffrey B, Keith Poole, Howard Rosenthal, Adam Boche, Aaron Rudkin, and Luke Sonnet (2021). *Voteview: Congressional Roll-Call Votes Database*. URL: https://voteview.com/.

Lindsay, Bruce G (1988). "Composite likelihood methods". In: *Contemporary mathematics* 80.1, pp. 221–239.

Lyzinski, Vince, Daniel L Sussman, Minh Tang, Avanti Athreya, and Carey E Priebe (2014). "Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding". In: *Electronic journal of statistics* 8.2, pp. 2905–2922.

Ma, Zhuang, Zongming Ma, and Hongsong Yuan (2020). "Universal Latent Space Model Fitting for Large Networks with Edge Covariates." In: *J. Mach. Learn. Res.* 21, pp. 4–1.

McPherson, Miller, Lynn Smith-Lovin, and James M Cook (2001). "Birds of a feather: Homophily in social networks". In: *Annual review of sociology* 27.1, pp. 415–444.

McSherry, Frank (2001). "Spectral partitioning of random graphs". In: *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. IEEE, pp. 529–537.

Mele, Angelo, Lingxin Hao, Joshua Cape, and Carey E Priebe (2022). "Spectral estimation of large stochastic blockmodels with discrete nodal covariates". In: *Journal of Business & Economic Statistics*, pp. 1–13.

Mohamed, Mohamed S, Dung Nguyen, Anil Vullikanti, and Ravi Tandon (2022). "Differentially private community detection for stochastic block models". In: *International Conference on Machine Learning*. PMLR, pp. 15858–15894.

Mu, Cong, Angelo Mele, Lingxin Hao, Joshua Cape, Avanti Athreya, and Carey E. Priebe (2022). "On Spectral Algorithms for Community Detection in Stochastic Blockmodel Graphs With Vertex Covariates". In: *IEEE Transactions on Network Science and Engineering* 9.5, pp. 3373–3384. DOI: 10.1109/TNSE.2022.3177708.

Mülle, Yvonne, Chris Clifton, and Klemens Böhm (2015). "Privacy-Integrated Graph Clustering Through Differential Privacy." In: *EDBT/ICDT Workshops*, pp. 247–254.

Newman, Mark EJ and Aaron Clauset (2016). "Structure and inference in annotated networks". In: *Nature communications* 7.1, pp. 1–11.

Ng, Andrew Y, Michael I Jordan, and Yair Weiss (2002). "On spectral clustering: Analysis and an algorithm". In: *Advances in neural information processing systems*, pp. 849–856.

Nguyen, Hiep H, Abdessamad Imine, and Michaël Rusinowitch (2016). "Detecting communities under differential privacy". In: *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, pp. 83–93.

Nuñez von Voigt, Saskia and Florian Tschorsch (2019). "RRTxFM: Probabilistic counting for differentially private statistics". In: *Conference on e-Business, e-Services and e-Society*. Springer, pp. 86–98.

Pagh, Rasmus and Nina Mesing Stausholm (2021). "Efficient Differentially Private F_0 Linear Sketching". In: *24th International Conference on Database Theory (ICDT 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Peel, Leto, Daniel B Larremore, and Aaron Clauset (2017). "The ground truth about metadata and community detection in networks". In: *Science advances* 3.5, e1602548.

Qin, Tai and Karl Rohe (2013). "Regularized spectral clustering under the degree-corrected stochastic blockmodel". In: *Advances in Neural Information Processing Systems*, pp. 3120–3128.

Qin, Zhan, Ting Yu, Yin Yang, Issa Khalil, Xiaokui Xiao, and Kui Ren (2017). "Generating Synthetic Decentralized Social Graphs with Local Differential Privacy". en. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17*. Dallas, Texas, USA: ACM Press, pp. 425–438. (Visited on 10/02/2019).

Rohe, Karl, Sourav Chatterjee, and Bin Yu (2011). "Spectral clustering and the high-dimensional stochastic blockmodel". In: *The Annals of Statistics* 39.4, pp. 1878–1915.

Roy, Sandipan, Yves Atchadé, and George Michailidis (2019). "Likelihood inference for large scale stochastic blockmodels with covariates based on a divide-and-conquer parallelizable algorithm with communication". In: *Journal of Computational and Graphical Statistics* 28.3, pp. 609–619.

Rubin-Delanchy, Patrick, Joshua Cape, Minh Tang, and Carey E. Priebe (2022). "A statistical interpretation of spectral embedding: The generalised random dot product graph". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 84.4, pp. 1446–1473. DOI: 10.1111/rssb.12509.

Shalizi, Cosma Rohilla and Andrew C Thomas (2011). "Homophily and contagion are generically confounded in observational social network studies". In: *Sociological methods & research* 40.2, pp. 211–239.

Shrum, Wesley, Neil H Cheek Jr, and Saundra MacD (1988). "Friendship in school: Gender and racial homophily". In: *Sociology of Education*, pp. 227–239.

Smith, Adam, Shuang Song, and Abhradeep Guha Thakurta (2020). "The flajolet-martin sketch itself preserves differential privacy: Private counting with minimal space". In: *Advances in Neural Information Processing Systems* 33, pp. 19561–19572.

Smith, Kirsten P and Nicholas A Christakis (2008). "Social networks and health". In: *Annu. Rev. Sociol* 34, pp. 405–429.

Snijders, Tom A. B. and Krzysztof Nowicki (1997). "Estimation and prediction for stochastic blockmodels for graphs with latent block structure". In: *Journal of classification* 14.1, pp. 75–100.

Stanojevic, Rade, Mohamed Nabeel, and Ting Yu (2017). "Distributed cardinality estimation of set operations with differential privacy". In: *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE, pp. 37–48.

Su, Liangjun, Wuyi Wang, and Yichong Zhang (2019). "Strong consistency of spectral clustering for stochastic block models". In: *IEEE Transactions on Information Theory* 66.1, pp. 324–338.

Sweet, Tracy M (2015). "Incorporating covariates into stochastic blockmodels". In: *Journal of Educational and Behavioral Statistics* 40.6, pp. 635–664.

Tallberg, Christian (2004). "A Bayesian approach to modeling stochastic blockstructures with covariates". In: *Journal of Mathematical Sociology* 29.1, pp. 1–23.

Tang, Minh, Joshua Cape, and Carey E Priebe (2022). "Asymptotically efficient estimators for stochastic blockmodels: The naive MLE, the rank-constrained MLE, and the spectral estimator". In: *Bernoulli* 28.2, pp. 1049–1073.

Ting, Daniel (2019). "Approximate distinct counts for billions of datasets". In: *Proceedings of the 2019 International Conference on Management of Data*, pp. 69–86.

Traud, Amanda L, Peter J Mucha, and Mason A Porter (2012). "Social structure of Facebook networks". In: *Physica A: Statistical Mechanics and its Applications* 391.16, pp. 4165–4180.

Tschorsch, Florian and Björn Scheuermann (2013). "An algorithm for privacy-preserving distributed user statistics". In: *Computer Networks* 57.14, pp. 2775–2787.

Varin, Cristiano, Nancy Reid, and David Firth (2011). "An overview of composite likelihood methods". In: *Statistica Sinica*, pp. 5–42.

Vishwanath, Siddharth and Jonathan Hehir (2022). "Topological Inference for Random Dot-Product Graphs under Local Differential Privacy". In: *Forthcoming*.

Von Luxburg, Ulrike (2007). "A tutorial on spectral clustering". In: *Statistics and computing* 17.4, pp. 395–416.

Vu, Duy Q, David R Hunter, and Michael Schweinberger (2013). "Model-based clustering of large networks". In: *The annals of applied statistics* 7.2, p. 1010.

Wang, Yuchung J and George Y Wong (1987). "Stochastic blockmodels for directed graphs". In: *Journal of the American Statistical Association* 82.397, pp. 8–19.

Warner, Stanley L (1965). "Randomized response: A survey technique for eliminating evasive answer bias". In: *Journal of the American Statistical Association* 60.309, pp. 63–69.

Wasserman, Larry and Shuheng Zhou (Mar. 2010). "A Statistical Framework for Differential Privacy". en. In: *Journal of the American Statistical Association* 105.489, pp. 375–389. URL: http://www.tandfonline.com/doi/abs/10.1198/jasa.2009.tm08651 (visited on 10/02/2019).

Weng, Haolei and Yang Feng (2021). "Community detection with nodal information: likelihood and its variational approximation". In: *Stat*, e428.

Whang, Kyu-Young, Brad T Vander-Zanden, and Howard M Taylor (1990). "A linear-time probabilistic counting algorithm for database applications". In: *ACM Transactions on Database Systems (TODS)* 15.2, pp. 208–229.

Yang, Jaewon, Julian McAuley, and Jure Leskovec (2013). "Community detection in networks with node attributes". In: *2013 IEEE 13th international conference on data mining*. IEEE, pp. 1151–1156.

Young, Stephen J and Edward R Scheinerman (2007). "Random dot product graph models for social networks". In: *International Workshop on Algorithms and Models for the Web-Graph*. Springer, pp. 138–149.

Zhang, Yun, Kehui Chen, Allan Sampson, Kai Hwang, and Beatriz Luna (2019). "Node features adjusted stochastic block model". In: *Journal of Computational and Graphical Statistics* 28.2, pp. 362–373.

Vita

Jonathan Hehir

Jonathan holds a Ph.D. in Statistics from The Pennsylvania State University, a B.A. in Mathematics from the University at Albany, SUNY, and an A.A.S. in Computer Information Systems from Hudson Valley Community College. Concurrent to the Ph.D., Jonathan held internships and part-time software engineering roles at Meta, Asana, and BuySellAds.com. While pursing his undergraduate degrees, he held full-time software engineering and development roles at BuySellAds.com, New York State Teachers' Retirement System, and the Asssociation for the Advancement of Sustainability in Higher Education.