Journal of Machine Learning Research 22 (2021) 1-29 Submitted 4/18; Revised 8/20; Published 5/21

Edge Sampling Using Local Network Information

Can M. Le CANLE@UCDAVIS.EDU
Department of Statistics
University of California, Davis
Davis, CA 95616-5270, USA

Editor: Vahab Mirrokni

Abstract

Edge sampling is an important topic in network analysis. It provides a natural way to reduce network size while retaining desired features of the original network. Sampling methods that only use local information are common in practice as they do not require access to the entire network and can be easily parallelized. Despite promising empirical performances, most of these methods are derived from heuristic considerations and lack theoretical justification. In this paper, we study a simple and efficient edge sampling method that uses local network information. We show that when the local connectivity is sufficiently strong, the sampled network satisfies a strong spectral property. We measure the strength of local connectivity by a global parameter and relate it to more common network statistics such as the clustering coefficient and network curvature. Based on this result, we also provide sufficient conditions under which random networks and hypergraphs can be efficiently sampled.

Keywords: Network analysis, edge sampling, local information, network curvature, clustering coefficient.

1. Introduction

Network analysis has become an important area in many research domains. It provides a natural way to model and analyze data with a complex interdependence among entities. A network typically consists of nodes representing the entities of interest and edges between nodes encoding the relations between the nodes. For example, in a social network such as Facebook or Twitter, nodes are users, and there is an edge between two users if they are friends. Studying the structure of a network provides valuable information about how entities interact and may help predict the formation of different groups Goldenberg et al. (2010); Fortunato (2010).

As real-world networks are often very large, it is difficult and often impossible to store or even access the entire data set. Therefore, it is desirable to preprocess the data to reduce the network size before performing any analysis. A natural method for this task is graph sparsification, a well-known edge sampling method in network literature Benczúr and Karger (1996); Spielman and Teng (2004); Spielman and Srivastava (2011). For a network of n nodes, one samples edges independently with probabilities proportional to their effective resistances, that is, the electrical resistances between the same nodes in the resistor network obtained from the original network by replacing edges with resistors of unit conductance Ghosh et al. (2008). It has been shown that sampling and storing $O(n \log n)$ weighted

edges is sufficient for approximately preserving the important topological structure of the original network Spielman and Srivastava (2011). Specifically, for an undirected network $G = (V, E)$ with the set of nodes $V = \{1, 2, ..., n\}$ and the set of edges $E \subseteq V \times V$, let A be the adjacency matrix with $A_{ij} = 1$ if $(i, j) \in E$ and $A_{ij} = 0$ otherwise. Let $L_G = D - A$ be the Laplacian, where D is the diagonal matrix with node degrees $d_i = \sum_{j \in V} A_{ij}$ on the diagonal, and define the Laplacian L_H for the weighted network H output by graph sparsification in a similar way. Then H satisfies the following inequality for every $x \in \mathbb{R}^n$, known as the strong spectral property:

$$(1 - \varepsilon)x^{\mathsf{T}} L_G x \leq x^{\mathsf{T}} L_H x \leq (1 + \varepsilon)x^{\mathsf{T}} L_G x. \tag{1.1}$$

Although this method has a strong theoretical guarantee, a severe drawback it suffers from, especially when applied to very large networks, is that it requires access to the entire network for computing effective resistances of all edges. Also, the computation involves a complicated linear system solver of Spielman and Teng, not easy to implement in practice. Although some improvements of Spielman and Srivastava (2011) have been proposed, they still rely on complicated linear system solvers Kelner and Levin (2013); Kapralov et al. (2014).

To avoid this problem, several fast and simple edge sampling methods have been developed with more emphasis on preserving certain network features such as the number of connected components, network diameter, homophily, node centrality measures, or community structure Newman (2010). One of the simplest sampling methods is uniform sampling, which samples edges independently and uniformly at random Sadhanala et al. (2016); Li et al. (2020). More adaptive methods leverage the strong local connectivity of networks widely observed in practice: the network neighborhoods of most of the nodes are surprisingly dense Watts and Strogatz (1998); Ugander et al. (2011). They sample edges according to certain edge scores that can be calculated locally without access to the entire network, such as the Jaccard similarity score Satuluri et al. (2011), the number of triangles Hamann et al. (2016), or the number of quadrangles containing the edges under consideration Nocaj et al. (2014); see also Hamann et al. (2016) for methods based on other local measures. Although these methods have been empirically shown to perform well and can be parallelized easily, to our best knowledge, there is still no theoretical guarantee for their performances. It is also unclear if other features of networks (besides the targeting features considered) are preserved.

In an attempt to understand the theoretical properties of these methods, in this paper, we study a fast and simple edge sampling scheme similar to methods that use Jaccard similarity, or numbers of triangles Satuluri et al. (2011); Hamann et al. (2016). Specifically, for an undirected network $G = (V, E)$, we sample each edge $(i, j) \in E$ with probability inversely proportional to the number of common neighbors of i and j (those nodes that are connected to both i and j). The numbers of common neighbors have been used in network literature, for example, in the context of community detection Rohe and Qin (2013) and network embedding Papadopoulos et al. (2015).

We observe that when the numbers of common neighbors are sufficiently large compared to node degrees, our sampling method satisfies the same strong spectral property (1.1) that the graph sparsification does while avoiding the complicated calculation of effective resistances. This result also provides theoretical evidence supporting edge sampling methods

based on local statistics Satuluri et al. (2011); Nocaj et al. (2014); Hamann et al. (2016). Qualitatively, as the number of common neighbors increases, the local network connectivity gets stronger, and our sampling method becomes more similar to graph sparsification using effective resistances. In contrast, as the numbers of common neighbors decrease, the method becomes more similar to uniform sampling. We measure the strength of the local network connectivity by the following parameter

$$\alpha = \frac{1}{n} \sum_{(i,j)\in E} \frac{2}{t_{ij}+2}, \tag{1.2}$$

where t_{ij} denotes the number of common neighbors of node i and node j. As we will show, α is closely related to other well-known and similar in nature statistics such as the clustering coefficient Watts and Strogatz (1998) and network curvature Bauer et al. (2012); see Section 3 for the definition. More importantly, it determines the sample size (the number of sampled edges) needed for the strong spectral property to hold.

1.1 Our Contributions

We make the following contributions in this paper. First, in Section 2 we propose a simple sampling method by leveraging the strong local connectivity widely observed for real-world networks and show that it satisfies the strong spectral property (1.1) if we sample $O(\alpha n \log n)$ edges, where α is defined by (1.2). Consequently, we show that uniform sampling with replacement also satisfies (1.1) if the sample size is sufficiently large; the exact value is given by (2.4). This requirement can be relaxed if a hybrid sampling method that combines uniform sampling and sampling according to the number of common neighbors is used. Second, we provide lower and upper bounds on α for generic networks in terms of the clustering coefficient and network curvature (Section 3). Since α directly determines the sample size required for the strong spectral property, these bounds provide useful information about when our sampling method can be efficiently used. They also show a connection with other sampling methods that use different local statistics Satuluri et al. (2011); Nocaj et al. (2014); Hamann et al. (2016) for which the theory developed in this paper may potentially be applied. Third, in Section 4 we provide an upper bound on α for the general inhomogeneous Erdős-Rényi random graph model Bollobas et al. (2007). Since this model is very popular in network literature, the bound provides a rich class of examples for which our sampling method can be used for reducing the network size. We discuss in Section 5 another natural class of examples, the hypergraphs, for which our method can be found useful. Lastly, in Section 7 we show that α is small for many real-world networks and perform a thorough numerical study to evaluate our sampling method.

1.2 Related Work

The simplest sampling method is bond percolation, which independently selects edges with a fixed probability ε Alon et al. (2004); Nachmias (2010); Bollobás et al. (2010). If ε is sufficiently large so that $\Omega(n \log n)$ edges are selected, then with high probability, the adjacency matrix of the sparsified network concentrates around εA by a standard matrix concentration result Oliveira (2010). The advantage of this method is that it is fast and only requires the total number of edges in the network as a global input parameter. However,

it satisfies a much weaker property than the strong spectral property Spielman and Teng (2004). A closely related method is uniform sampling, for which Sadhanala et al. (2016) shows that (1.1) holds with high probability, but only for smooth vectors x.

In semi-streaming setting, Benczúr and Karger (1996) and Goel et al. (2010) show that local network structure can be used to design sampling methods that approximately preserve all cuts of the original network; here, the cut of a set of nodes is the number of edges between that set and its complement in V. However, this property is strictly weaker than the strong spectral property that our method satisfies Kelner and Levin (2013).

2. Edge Sampling

For an undirected network $G = (V, E)$ and $(i, j) \in E$, let t_{ij} be the number of common neighbors of i and j. For simplicity of presentation, we first discuss the case when t_{ij} are known for all edges. In practice, they can be either exactly calculated in a parallel manner or approximated by neighbor sampling; see Section 6 for a more detailed discussion. To form a sparsifier H, we sample m edges of G independently according to a multinomial distribution with probabilities

$$p_{ij} = \frac{\frac{2}{t_{ij}+2}}{\sum_{(i,j)\in E} \frac{2}{t_{ij}+2}}. \tag{2.1}$$

If an edge $(i, j) \in E$ is selected $k \geq 1$ times then we add it to H and assign the weight $k(mp_{ij})^{-1}$ to it.

Note that $2/(t_{ij}+2)$ is the effective resistance of the edge between i and j in a subgraph of G consisting of the edge (i, j) and t_{ij} paths of length two between i and j. Therefore, it is an upper bound of the effective resistance of the edge between i and j in G; for a detailed explanation, see the proof of Theorem 2 in Appendix A.

The following theorem shows that our sampling method satisfies the strong spectral property.

Theorem 1 (Exact numbers of common neighbors) *Consider an undirected and connected network $G = (V, E)$. Let $\varepsilon \in (0, 1)$ and α be the parameter of G defined by* (1.2)*. Form a weighted network H by sampling $8\alpha n \log n/\varepsilon^2$ edges of G as described above. Then H satisfies the strong spectral property* (1.1) *with probability at least $1 - 1/n$.*

Parameter α measures the average strength of local network connectivity. To better understand α, consider a special case when $d_i = d$ for all vertices i and $t_{ij} = t$ for all edges $(i, j) \in E$. Then $\alpha \approx 2|E|/(nt) = d/t$, where here and after we use $|\mathcal{M}|$ to denote the number of elements of the set $\mathcal{M}$. Thus, if $(i, j) \in E$ then the number of common neighbors of i and j is approximately d/α. In other words, i and j share a fraction of $1/\alpha$ of their neighbors.

When the local connectivity is strong, that is, $\alpha = O(1)$, Theorem 1 shows that we can approximately preserve the network topology if we locally sample and retain $O(n \log n)$ edges. In contrast, if the local connectivity is weak (for example, when $t_{ij} = O(1)$), then p_{ij} are of the same order, resulting in a sampling scheme similar to uniform sampling. Table 2 shows the value of α and the clustering coefficient (see Section 3.1 for the definition) for

several well-known real-world networks. Note that while these networks are relatively sparse, the values of α are quite small, which suggests that real-world networks have strong local connectivity.

The above sampling method requires access to the number of common neighbors t_{ij} for all pairs of incident nodes. If t_{ij} are readily available, which is the case for some social networks such as Facebook, then the computational complexity of this sampling method is linear in the total number of edges $|E|$. When t_{ij} are not available, we can calculate them in parallel fashion or estimate them by neighbor sampling; see Section 6 for more detail. The following theorem shows that the strong spectral property still holds if we use estimates of t_{ij} and increase the sample size by a factor depending on the accuracy of the estimates.

Theorem 2 (Estimated numbers of common neighbors) *Consider an undirected and connected network $G = (V, E)$ and let $\hat{t}_{ij}$ be nonnegative estimates of t_{ij} such that*

$$\hat{t}_{ij} + 2 \le C(t_{ij} + 2) \tag{2.2}$$

for all edges $(i, j) \in E$ and some constant C. Let $\varepsilon \in (0, 1)$ and denote

$$\hat{\alpha} = \frac{1}{n} \sum_{(i,j)\in E_G} \frac{2}{\hat{t}_{ij} + 2}. \tag{2.3}$$

Form a weighted graph H by sampling $8C\hat{\alpha}n \log n/\varepsilon^2$ edges of G as described in Theorem 1 but using $\hat{t}_{ij}$ instead of t_{ij}. Then H satisfies the spectral property (1.1) *with probability at least $1 - 1/n$.*

The proof of Theorem 2 depends crucially on condition (2.2). It implies that $2/(t_{ij} + 2) \le 2C/(\hat{t}_{ij} + 2)$ and consequently the effective resistance of the edge (i, j) is bounded by $2C/(\hat{t}_{ij} + 2)$. This observation allows us to express the Laplacian of the sparsified network as a sum of independent matrices with spectral norms bounded by $C\hat{\alpha}n$ up to a scaling matrix factor. A standard matrix concentration result is then used to show the strong spectral property; see the proof in Appendix A for more detail. Note that Theorem 1 follows directly from Theorem 2 by setting $\hat{t}_{ij} = t_{ij}$ and $C = 1$.

One may wonder how many edges must be sampled so that the uniform sampling (which samples edges with probabilities $p_{ij} = 1/|E|$) satisfies the spectral property (1.1). The uniform sampling is obtained by setting $\hat{t}_{ij} = t$ for all edges of G in Theorem 2. The constant C can be taken to be

$$C = \frac{t + 2}{\min_{(i,j)\in E} t_{ij} + 2} \quad \text{and} \quad \hat{\alpha} = \frac{2|E|}{n(t + 2)}.$$

Therefore by Theorem 2, the uniform sampling satisfies (1.1) with high probability if the sample size is

$$m = \frac{16\varepsilon^{-2}|E| \log n}{\min_{(i,j)\in E} t_{ij} + 2}. \tag{2.4}$$

If $\min_{(i,j)\in E} t_{ij}$ is of order $|E|/n$, that is, the numbers of common neighbors are at least a constant fraction of the average degree, then $m = O(n \log n)$.

In general, the sample size requirement (2.4) is optimal up to the logarithm and constant factors. That is, (1.1) needs not hold if $m = o(|E|/(\min_{(i,j)\in E} t_{ij} + 2))$. To see this, consider an example of a graph G consisting of a complete graph of $n-1$ nodes and a node i of degree $k = o(n)$. Then $\min_{(i,j)\in E} t_{ij} = k - 1$. If $m = o(|E|/(k+1))$ and edges of G are sampled uniformly then the probability that no edges incident to i is selected is

$$\left(1 - \frac{k}{|E|}\right)^m = \left(1 - \frac{k}{|E|}\right)^{\frac{|E|}{k}\cdot\frac{mk}{|E|}} \approx \exp\left(-\frac{mk}{|E|}\right) \approx 1.$$

With probability close to one, i is an isolated node in the (weighted) sparsified graph. Therefore the degree of i cannot be approximately preserved, which implies that the spectral property (1.1) does not hold.

For graphs with a small value of $\min_{(i,j)\in E} t_{ij}$, the sample size m in (2.4) for the uniform sampling scheme may get as large as the total number of edges $|E|$, which defies the purpose of graph sparsification. By choosing $\hat{t}_{ij} = t$ only if $t_{ij} > t$ for some large threshold t and $\hat{t}_{ij} = t_{ij}$ if $t_{ij} \le t$, we obtain a hybrid of uniform sampling and sampling using common neighbors that may require smaller sample size than (2.4). Indeed, in Theorem 2, we can choose $C = 1$ and

$$\hat{\alpha} = \frac{1}{n}\sum_{(i,j)\in E: t_{ij}\le t} \frac{2}{t_{ij}+2} + \frac{1}{n}\sum_{(i,j)\in E: t_{ij} > t} \frac{2}{t+2} \le \alpha + \frac{2|E|}{n(t+2)}.$$

The required sample size for the hybrid method to obtain the spectral property (1.1) with high probability is then

$$8\varepsilon^{-2}C\hat{\alpha} n \log n \le 8\varepsilon^{-2}\left(\alpha + \frac{2|E|}{n(t+2)}\right) n\log n = 8\varepsilon^{-2}\alpha n \log n + \frac{16|E|\log n}{\varepsilon^2(t+2)},$$

which is smaller than the sample size in (2.4) if $\alpha n = o(|E|/(\min_{(i,j)\in E} t_{ij} + 2))$ and $\min_{(i,j)\in E} t_{ij} = o(t)$. This hybrid method illustrates an interesting application of Theorem 2 and may also be useful when uniform sampling is desirable, for example for controlling the variance of the sparsified graph.

3. Local Network Statistics

In this section, we draw the connection between α and two of the most common local network statistics, the clustering coefficient Watts and Strogatz (1998) and the network curvature Bauer et al. (2012).

3.1 Clustering Coefficient

It has been observed that for many real-world networks, the neighborhoods of most of the nodes are surprisingly dense Watts and Strogatz (1998); Ugander et al. (2011). This reflects the belief that incident nodes exhibit the transitivity property: if i and j are connected and j and k are connected then it is likely that i and k are also connected. One way to measure the transitivity is via the clustering coefficient Watts and Strogatz (1998). For an undirected network $G = (V, E)$, the local clustering coefficient of node $i \in V$ is defined as the ratio

between the number of triangles containing i and the maximum number of triangles it can form with incident nodes

$$c_i = \frac{|\{(j,k) \in E : (i,j) \in E, (i,k) \in E\}|}{d_i(d_i-1)/2}.$$

The clustering coefficient of a network G is the average of all local clustering coefficients

$$c = \frac{1}{n}\sum_{i=1}^{n} c_i.$$

The following theorem provides a lower bound on parameter α in terms of the clustering coefficient c and node degrees d_i. It shows that if node degrees are large and c is small, then α is large, and therefore a large sample size is required for our method to obtain the spectral property (1.1). On the other hand, Table 2 suggests that α is small when c is large. Since the clustering coefficient is a very popular statistic and has been calculated for most available real-world networks, the connection to the clustering coefficient provides valuable information about α before the sampling procedure is performed.

Theorem 3 (Lower bound on α) *For any undirected and connected network we have*

$$\alpha \geq \frac{1}{4c + \frac{2}{n}\sum_{i=1}^{n}\frac{1}{d_i}}. \tag{3.1}$$

According to Theorem 3, if $c \gtrsim 1/n\sum_{i\in V} 1/d_i$ then α satisfies $\alpha \gtrsim 1/c$ (for two sequences a_n and b_n, we write $a_n \gtrsim b_n$ if $a_n \geq Cb_n$ for some constant C and sufficiently large n). The geometric random graph model described in Corollary 6 below provides examples for which the upper bound $\alpha \lesssim 1/c$ also holds; for more detail, see the discussion following Corollary 6. In addition, Table 2 gives examples of real networks for which α and $1/c$ are of similar order.

There exist graphs for which the two sides of (3.1) are of different orders. For example, let $G = K_n \cup E_n$ be the union of a complete graph K_n of size n and an Erdős-Rényi random graph E_n, also of size n, for which edges are formed independently between each pair of nodes with probability d/n; we connect K_n and E_n by an arbitrary edge to make G a connected graph. If $\sqrt{n} \lesssim d = o(n)$ then an easy calculation shows that with high probability, the left hand-side of (3.1) is of order n/d while the right hand-side is bounded.

3.2 Network Curvature

Another measure of network transitivity that has recently attracted much attention is the network curvature Bauer et al. (2012); Jost and Liu (2014); Lin et al. (2014); Bhattacharya and Mukherjee (2015). In this section, we recall the definition of network curvature and show that if it is bounded from below by some constant $\kappa_0 > 0$ then $\alpha \leq 1/\kappa_0$.

Denote by $d(i,j)$ the length of a shortest path connecting nodes i and j. For each node i, consider a uniform measure m_i with support being the set N_i of neighbors of i:

$$m_i(k) = \begin{cases} \frac{1}{d_i}, & \text{if } k \in N_i \\ 0, & \text{otherwise.} \end{cases}$$

The optimal transportation distance between m_i and m_j is defined as follows:

$$W_1(m_i, m_j) = \inf_{\xi \in \Pi(m_i, m_j)} \sum_{(k,k') \in V \times V} d(k, k')\xi(k, k'),$$

where $\Pi(m_i, m_j)$ is the set of all probability measures on $V \times V$ with marginals m_i and m_j. Intuitively, $\xi(k, k')$ represents the mass transported from k to k', and $W_1(m_i, m_j)$ is the optimal cost for moving a unit mass distributed evenly among neighbors of i to neighbors of j. With this notion of distance between probability measures on G, the curvature κ defined for every pair of nodes i and j is

$$\kappa(i, j) = 1 - \frac{W_1(m_i, m_j)}{d(i, j)}.$$

To illustrate, in Figure 1 we show Zachary's karate club network Zachary (1977) together with the information of its curvatures for incident nodes. In particular, edges with negative curvatures are in blue, positive curvatures – in red and zero curvatures – in black; widths of edges are proportional to magnitudes of curvatures.

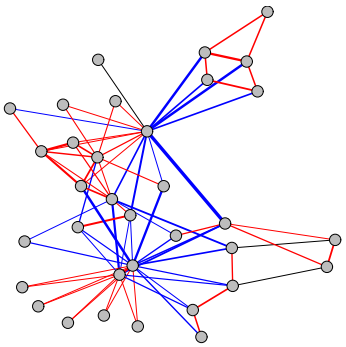

Figure 1: Zachary's karate club network Zachary (1977). The edges with negative curvatures are in blue, with positive curvatures – in red and with zero curvatures – in black; the widths of the edges are proportional to the magnitudes of their curvatures.

We say $\kappa \geq \kappa_0$ for some constant κ_0 if $\kappa(i, j) \geq \kappa_0$ for every pair of nodes i and j. If $\kappa \geq \kappa_0$ then by definition $W_1(m_i, m_j) \leq (1 - \kappa_0)d(i, j)$ for all $(i, j) \in V \times V$. In particular, if i and j are connected then $W_1(m_i, m_j) \leq 1 - \kappa_0$. Note that if G is a connected graph then the inverse is also true: If $W_1(m_i, m_j) \leq 1 - \kappa_0$ holds for all pairs of connected nodes i and j then $W_1(m_i, m_j) \leq (1 - \kappa_0)d(i, j)$ holds for all $(i, j) \in V \times V$ by a triangle inequality.

This notion of curvature is closely related to the simple random walk on a network. If $\kappa \geq \kappa_0 > 0$ then Ollivier (2009) shows that the spectral gap between the two largest eigenvalues of the transition matrix $D^{-1}A$ is bounded from below by κ_0 (see also Bauer et al. (2012) for an improvement of the bound). Thus, the curvature of a graph controls how fast a simple random walk on that network mixes.

The following theorem provides a simple upper bound on α in terms of the curvature.

Theorem 4 (Upper bound on α) *Let G be an undirected and connected network. Assume there exist constants $\kappa_0 > 0$ and $C > 0$ such that $\kappa(i,j) \geq \kappa_0$ for all but at most Cn edges of G. Then $\alpha \leq 1/\kappa_0 + C$.*

4. Random Networks

In this section, we provide a high probability bound on α for inhomogeneous Erdős-Rényi random networks Bollobas et al. (2007) satisfying some mild conditions. As a corollary, we give an example of a geometric random network model for which α is bounded.

Theorem 5 (Inhomogeneous Erdős-Rényi networks) *Consider a random graph with adjacency matrix A such that the upper diagonal elements of A are independent Bernoulli random variables. Denote $P = \mathbb{E}\, A$ and $\Delta = \max_i \sum_{j=1}^n P_{ij}$. Assume that there exists a sufficiently large constant C such that*

$$\Delta \geq C \log n \qquad \text{and} \qquad \Delta \cdot \left[1 + \max_{i,j} \left(P^2\right)_{ij}\right] \leq \frac{1}{C \log n} \sum_{i<j} P_{ij}. \tag{4.1}$$

Then with probability at least $1 - 1/n$,

$$\alpha \leq \frac{1}{n} \sum_{i<j} \frac{10 P_{ij}}{\mathbb{E}\, t_{ij} + 2}. \tag{4.2}$$

In particular, if the right-hand side of (4.2) is bounded, then α is also bounded.

The first inequality of (4.1) requires that the maximal expected node degree grow at least as $\log n$; this is a natural condition because otherwise, the network would already be sparse, and no sampling would be needed. The second inequality of (4.1) is a condition on the maximal expected degree Δ, the maximal expected number of common neighbors $\max_{ij}(P^2)_{ij}$ and the expected number of edges $1/2 \sum_{i<j} P_{ij}$. If all nodes in the graph are of similar expected degree then $\sum_{i<j} P_{ij} \approx n\Delta$. Therefore, using the crude bound $(P^2)_{ij} \leq \Delta$, the second inequality of (4.1) is satisfied if Δ is at most of order $n/\log n$.

By Jensen's inequality and the independence between A_{ij} and t_{ij}, we have

$$\mathbb{E}\, \alpha = \frac{1}{n} \sum_{i<j} \mathbb{E}\, \frac{2A_{ij}}{t_{ij} + 2} = \frac{1}{n} \sum_{i<j} \mathbb{E}\, \frac{2P_{ij}}{t_{ij} + 2} \geq \frac{1}{n} \sum_{i<j} \frac{2P_{ij}}{\mathbb{E}\, t_{ij} + 2}.$$

It then follows from (4.2) that $\alpha \leq 5\, \mathbb{E}\, \alpha$ with probability at least $1 - 1/n$, while naively applying Markov's inequality gives the same inequality with probability at least $4/5$. Note, however, that the upper bound of (4.2) is much easier to calculate than $\mathbb{E}\, \alpha$.

As a direct consequence of Theorem 5, the following corollary shows that α is bounded for a simple geometric random network model.

Corollary 6 (Geometric random networks) *Let $X = \{x_1, x_2, ..., x_n\} \subseteq K$ be a set of points in a bounded set $K \subseteq \mathbb{R}^d$ with unit volume. For each pair of nodes (i,j), let*

$$P_{ij} = \begin{cases} \delta, & \text{if } \|x_i - x_j\| \leq r_n, \\ 0, & \text{otherwise.} \end{cases}$$

Denote by n_i the number of points of X of distance at most R_n from x_i; similarly, denote by n_{ij} the number of points of X of distance at most R_n from x_i and x_j. Assume that there exists a constant $C = C(d, K)$ depending only on d and K such that for every node i and every node j with $\|x_j - x_i\| \leq r_n$,

$$C^{-1}nr_n^d \leq n_i, n_{ij} \leq Cnr_n^d \quad \text{and} \quad C^2\delta^{-1}\log n \leq nr_n^d \leq \frac{n}{4C^3\delta^2 \log n}. \tag{4.3}$$

Then $\alpha \leq 5C^2/\delta$ with probability at least $1 - 1/n$.

The first condition of (4.3) holds with high probability if $x_1, ..., x_n$ are independently drawn from a uniform distribution on K. Indeed, for every node i, n_i/n is approximately the volume of the ball of radius r_n and center x_i, which is proportional to r_n^d up to a constant depending on d and K; a similar argument holds for n_{ij} with $\|x_j - x_i\| \leq r_n$. The second condition of (4.3) requires that the average degree of the graph be roughly between $\log n$ and $n/\log n$. If these conditions are satisfied then α is bounded with high probability.

Theorem 3 shows that $\alpha \gtrsim 1/c$ if the clustering coefficient c is at least of the same order as $1/n\sum_{i\in V} 1/d_i$. Corollary 6 provides examples for which the reverse bound also holds. Indeed, since $\alpha \lesssim 1/\delta$ with high probability by Corollary 6, the bound $\alpha \lesssim 1/c$ holds if $c \gtrsim \delta$ with high probability. To see why that is the case, for every node i let N_i be the set of all neighbors of i. Then conditioned on N_i, the probability that two neighbors of i are connected is at most δ. Therefore the number of triangles containing i is stochastically bounded by a sum of $d_i(d_i - 1)/2$ independent Bernoulli random variables with success probability δ. Using a standard concentration result and union bounds, we see that the local clustering coefficients satisfy $c_i \lesssim \delta$ for all i with high probability. Since c is the average of c_i, this implies $c \gtrsim \delta$ with high probability.

5. Sampling Hypergraphs

Theorem 3 shows that $\alpha \gtrsim 1/c$ if the clustering coefficient c is at least of the same order as Strong local connectivity of a network is often caused by the fact that each node belongs to one or several tightly connected small groups Gupta et al. (2014). To simplify the analysis, we assume that within each small group, all nodes are connected. Under this assumption, a network can be modeled by a hypergraph $\mathcal{G} = (V, \mathcal{E})$ which consists of a set of nodes V and a set of hyperedges $\mathcal{E}$ where each hyperedge is a subset of V. In this section, we derive a condition under which a hypergraph can be sampled and reduced to a weighted network. This provides another example for which our sampling scheme works well and may be useful in practice as a computational acceleration technique.

The Laplacian previously defined for networks can be naturally extended to hypergraphs through clique expansion Rodríguez (2002); Agarwal et al. (2006). For a hypergraph $\mathcal{G} = (V, \mathcal{E})$, the evaluation of the Laplacian $L_\mathcal{G}$ at a vector x is defined by

$$L_\mathcal{G}(x) = \sum_{e\in\mathcal{E}} \sum_{i,j\in e} (x_i - x_j)^2.$$

If we view x as a function from V to $\mathbb{R}$ then $L_\mathcal{G}(x)$ measures the smoothness of x and it occurs naturally in many problems of estimating smooth functions Smola and Kondor

(2003); Belkin et al. (2004); Huang et al. (2011); Kirichenko and van Zanten (2017); Li et al. (2020); Le and Li (2020).

Let $G = (V, E, W)$ be a weighted network such that $(i, j) \in E$ if and only if both i and j belong to at least one hyperedge of $\mathcal{G}$, and W denotes the weight matrix with entries W_{ij} being the number of hyperedges that both i and j belong to. It is easy to see that $L_{\mathcal{G}}(x) = x^{\mathsf{T}} L_G x$ for every x, where L_G is the Laplacian of the weighted network G defined by

$$x^{\mathsf{T}} L_G x = \sum_{(i,j)\in E} W_{ij}(x_i - x_j)^2.$$

Thus, if we are mainly interested in the smoothness of functions determined by $\mathcal{G}$, we can replace $\mathcal{G}$ with G. We call G the weighted network induced by $\mathcal{G}$.

To form a sparsifier $H = (V, E_H, W_H)$ of G, we sample with replacement m edges of G with probability

$$\mathcal{P}_{ij} = \frac{\tilde{t}_{ij}^{-1}}{\sum_{(i,j)\in E} \tilde{t}_{ij}^{-1}}, \quad \text{where} \quad \tilde{t}_{ij} = \sum_{e\in\mathcal{E}:\{i,j\}\in e} |e|.$$

If an edge $(i, j) \in E$ is selected $k \geq 1$ times then we add (i, j) to E_H and assign the weight $k(m\mathcal{P}_{ij})^{-1}$ to it. Similar to the parameter α for unweighted graphs, let

$$\tilde{\alpha} = \frac{1}{n} \sum_{(i,j)\in E} \tilde{t}_{ij}^{-1}.$$

Lemma 7 (Upper bound on $\tilde{\alpha}$) *Let $\mathcal{G} = (V, \mathcal{E})$ be a hypergraph. If each node of $\mathcal{G}$ belongs to at most d hyperedges then $\tilde{\alpha} \leq d/2$.*

Without further assumptions on $\mathcal{G}$, the bound $\tilde{\alpha} \leq d/2$ is nearly optimal. To see this, consider the following example. Let $k > 0$ be an integer, $n = k^2$ and $V_1, ..., V_k$ be a partition of $V = \{1, ..., n\}$ such that each V_i contains exactly k elements $V_{i1}, ..., V_{ik}$. For each $1 \leq i \leq k$, let σ_i be a permutation of $\{1, 2, ..., k\}$ given by $\sigma_i(j) = i + j \pmod{k}$. Define the set of hyperedges of $\mathcal{G}$ as a collection of subsets of the form

$$\left\{V_{1j}, V_{2\sigma_i(j)}, ..., V_{k\sigma_i^{k-1}(j)}\right\}, \quad 1 \leq i, j \leq k.$$

It is easy to see that every node of $\mathcal{G}$ is contained in exactly $d = k$ hyperedges and every pair of nodes of $\mathcal{G}$ is contained in at most one hyperedge. A simple calculation shows that $\tilde{\alpha} = (d - 1)/2$.

The following theorem shows that the sparsified network obtained from a hypergraph satisfies the strong spectral property.

Theorem 8 (Sampling hypergraphs) *Let $\mathcal{G} = (V, \mathcal{E})$ be a hypergraph and $G = (V, E, W)$ be the weighted network induced by $\mathcal{G}$. Let $\varepsilon \in (0, 1)$ and assume that each node of $\mathcal{G}$ belongs to at most d hyperedges of $\mathcal{G}$. Form a weighted graph H by sampling $4dn \log n/\varepsilon^2$ edges of G as described above. Then H satisfies the strong spectral property (1.1) with probability at least $1 - 1/n$.*

6. Calculating the Numbers of Common Neighbors

In this section, we discuss the problem of calculating t_{ij} (either exactly or approximately), especially when the network is too large to be stored in a single computer. Once t_{ij} are all computed, the sampling method can be performed easily by sampling edges according to t_{ij} and aggregating over all sampled edges.

6.1 Exact Calculation

The number of common neighbors t_{ij} can be calculated efficiently by using an MPI-based distributed memory parallel algorithm in Arifuzzaman et al. (2019), with very little modification. The algorithm first carefully partitions the graph into smaller overlapping subgraphs and stores them separately in local machines. A sequential algorithm then finds all triangles in every subgraph and counts the number of common neighbors for every connected pair of nodes in that subgraph. Since an edge of the original graph may belong to different overlapping subgraphs, the counts from all local machines are then aggregated before the final result is output. The authors of Arifuzzaman et al. (2019) show that their algorithm scales almost linearly in the number of local machines and can handle very large graphs with billions of edges.

6.2 Estimation

Depending on the strength of the local connectivity of a graph, the computational complexity of our sampling method can be further improved by approximating t_{ij} instead of calculating them exactly. This section describes a simple method for estimating t_{ij} by sampling the neighbors of either i or j. A similar idea has been used in minwise hashing, a popular technique for efficiently estimating the Jaccard similarity between two sets Broder (1997); Broder et al. (1997); Becchetti et al. (2008); Satuluri et al. (2011); Shrivastava and Li (2014, 2015). Although the method described here is sequential, we can easily turn it into a parallel algorithm by adapting the method of Arifuzzaman et al. (2019) discussed in Section 6.1.

For each pair of connected nodes (i, j), denote by N_i the set of neighbors of i and by N_j the set of neighbors of j. The asymmetric Jaccard similarity between N_i and N_j is defined by

$$\theta_{ij} = \frac{|N_i \cap N_j|}{\min\{|N_i|, |N_j|\}} = \frac{t_{ij}}{\min\{d_i, d_j\}}.$$

Fix a sample size $k \geq 1$ and assume that $d_i \leq d_j$. If $k \geq d_i$ then simply counting the number of neighbors of i that are also neighbors of j gives us exactly t_{ij}. If $k < d_i$, let $Z_1, ..., Z_k$ be k random neighbors of i drawn independently and uniformly from N_i. We estimate t_{ij} by

$$\hat{t}_{ij} = \frac{d_i}{k} \sum_{\ell=1}^{k} \mathbf{1}(Z_\ell \in N_j),$$

where $\mathbf{1}(Z_\ell \in N_j)$ is the indicator of the event $Z_\ell \in N_j$. It is easy to see that $\hat{t}_{ij}$ is an unbiased estimate of t_{ij} because $\mathbf{1}(Z_\ell \in N_j)$, $1 \leq \ell \leq k$, are Bernoulli random variables with success probability θ_{ij}.

In order to apply Theorem 2, condition (2.2) must be satisfied for all edges $(i,j) \in E_G$. For those edges such that $\theta_{ij} \geq \varepsilon$ for some constant ε, we will show that (2.2) holds with high probability if k is chosen to be of order $\log n$. For those edges with $\theta_{ij} = o(1)$, $\hat{t}_{ij}$ may not satisfy (2.2) if $k = O(\log n)$, therefore we calculate t_{ij} directly. We use $\frac{1}{k}\sum_{\ell=1}^{k} \mathbf{1}(Z_\ell \in N_j)$ to check whether θ_{ij} is sufficiently large. The estimation procedure is summarized in the following algorithm.

Algorithm 1 *(Estimating the numbers of common neighbors) Choose $\varepsilon \in (0,1)$ and $k \geq 1$. For each edge (i,j), let i be the node with $d_i \leq d_j$. If $d_i \leq k$, calculate t_{ij} directly by counting the number of elements of $N_i \cap N_j$. If $d_i > k$, sample k neighbors $Z_1, ..., Z_k$ of i independently and uniformly from N_i, calculate $\hat{\theta}_{ij} = \frac{1}{k}\sum_{\ell=1}^{k} \mathbf{1}(Z_\ell \in N_j)$ and proceed as follows:*

- *If $\hat{\theta}_{ij} < \varepsilon$, calculate t_{ij} directly by counting the number of elements of $N_i \cap N_j$.*
- *If $\hat{\theta}_{ij} \geq \varepsilon$, estimate t_{ij} by $\hat{t}_{ij} = d_i \hat{\theta}_{ij}$.*

The following theorem provides the spectral guarantee (1.1) for the sparsified network when Algorithm 1 is used.

Theorem 9 (Minwise hashing) *Let $\varepsilon \in (0,1)$, $k = 100 \log n/\varepsilon$ and estimate the numbers of common neighbors using Algorithm 1. Form a weighted graph H by sampling $24\alpha n \log n/\varepsilon^2$ edges of G according to Theorem 2. Then with probability at least $1 - 1/n$, H satisfies the spectral property (1.1) and the computational complexity of estimating the number of common neighbors is at most*

$$\sum_{(i,j)\in E_G : \theta_{ij} \leq \varepsilon/2} \min\{d_i, d_j\} + 100\varepsilon^{-1}|E_G| \log n. \tag{6.1}$$

The complexity of estimating the number of common neighbors in Theorem 9 is nearly linear in the number of edges $|E_G|$ (up to the $\log n$ factor) and depends on the local structure of the network via the first term of (6.1). If the local connectivity of G is sufficiently strong so that $\theta_{ij} \geq \varepsilon/2$ for all edges, then the first term disappears. However, for networks with very weak local connectivity, such as Erdős-Rényi random networks, the first term of (6.1) may be as large as $d \cdot |E_G|$, where d is the average node degree. In that case, it is not clear if the computational complexity of the (sequential) estimation algorithm can be substantially improved; we leave this problem for future study.

Section 7.2 shows the performance of the proposed sampling method using both exact and estimated numbers of common neighbors.

7. Numerical Study

In this section, we empirically analyze the behavior of parameter α and the accuracy of the proposed sampling methods.

7.1 Local Connectivity

According to Theorem 1, α directly controls the accuracy of our sampling method. This section shows that α is relatively small for many simulated and real-world networks.

Network size	100	500	1000	2000	4000
Parameter α	1.84	2.34	2.49	2.59	2.65
Clustering coefficient	0.59	0.53	0.52	0.52	0.52
Average degree	27.64	36.42	38.89	40.89	42.57

Table 1: Statistics of networks generated from the GIRG with $\delta = \gamma = 2$, $r = 3$ and $\beta = 2.5$, averaged over 20 replications.

7.1.1 Simulated Networks

We consider geometric inhomogeneous random networks (GIRG) generated from a latent space model that exhibits several properties of real-world networks, such as the strong transitivity and the power-law distribution of node degrees Bringmann et al. (2017). To model the power law, each node i is assigned a weight $w_i = \delta \cdot (n/i)^{1/(\beta-1)}$, where $\delta > 0$ and $2 \leq \beta \leq 3$ are parameters. The latent positions x_i are drawn uniformly at random from an r-dimensional torus $\mathbb{T}^r = \mathbb{R}^r/\mathbb{Z}^r$ equipped with the distance

$$d(u, v) = \max_{1 \leq k \leq r} \min\{|u_k - v_k|, 1 - |u_k - v_k|\}.$$

For a parameter $\gamma > 1$ and $w = \sum_{i=1}^{n} w_i$, an edge is independently drawn between each pair of nodes i, j with probability

$$p_{ij} = \min\left\{ \frac{1}{\|x_i - x_j\|^{\gamma r}} \Big(\frac{w_i w_j}{w} \Big)^{\gamma}, 1 \right\}.$$

We report in Table 1 the value of α, the clustering coefficient and the average node degree (averaged over 20 replications) of networks generated from GIRG with parameter $\delta = \gamma = 2$, $r = 3$, $\beta = 2.5$ and $n = 100, 500, 1000, 2000, 4000$. Table 1 shows that while the network size and average node degree increase, the value of α increases mildly from 1.84 to 2.65 and the clustering coefficient decreases from 0.59 to 0.52.

7.1.2 Real-world Networks

We further report in Table 2 the value of α, the clustering coefficient and the average degree of several well-known real-world networks: karate club network Zachary (1977), dolphins network Lusseau et al. (2003), political blogs network Adamic and Glance (2005), Facebook ego network McAuley and Leskovec (2012), Astrophysics collaboration network Leskovec et al. (2007), Enron email network Klimt and Yang (2004), Twitter Social circles Yang and Leskovec (2012), Google+ social circles Yang and Leskovec (2012), DBLP collaboration network Yang and Leskovec (2012) and LiveJournal social network Yang and Leskovec (2012). Again, we observe that while the network size and average node degree vary, α and the clustering coefficient are very stable, with the value of α between 1.40 and 4.38 and the value of the clustering coefficient between 0.26 and 0.63, respectively.

Data	n	Average degree	c	α
Karate club	34	4.59	0.57	1.46
Dolphins	62	5.13	0.26	1.65
Political blogs	1490	22.44	0.32	3.04
Facebook ego	4039	43.69	0.61	1.96
Astrophysics collaboration	18771	21.10	0.63	1.96
Enron email	36692	10.02	0.50	1.59
Twitter	81306	33.02	0.57	2.33
Google+	107614	227.45	0.49	4.38
DBLP collaboration	317080	6.62	0.63	1.40
LiveJournal	3997962	17.35	0.28	3.65

Table 2: Statistics of some real-world networks.

7.2 Accuracy of Network Sampling

In this section, we compare the performance of our sampling method that uses the number of common neighbors (CN), the uniform sampling (UN), and a version of CN that uses $\hat{t}_{ij}$ defined in (6.2) to approximate the number of common neighbors (CNA). We use these methods to sparsify networks, both simulated and real-world, and then measure the accuracy of the resulting sparsified networks by comparing their Laplacians with those of the original networks. Motivated naturally by the strong spectral property (1.1), for a connected network G and its sparsification H, we report the following relative error

$$\text{Relative error} = \max_{x: L_G x \neq 0} \frac{x^\mathsf{T}(L_H - L_G)x}{x^\mathsf{T} L_G x} = \|L_G^{-1/2}(L_H - L_G)L_G^{-1/2}\|, \tag{7.1}$$

where $L_G^{-1/2}$ is the square root of the Moore–Penrose pseudo-inverse L_G^{-1} of L_G. This error reflects the accuracy of H in preserving the structure of G. Since calculating the relative error involves inverting the Laplacian, we consider in this section only networks of relatively small sizes.

7.2.1 Simulated Networks

We first analyze the performance of CN, CNA, and UN on random networks generated from the latent space stochastic block model Ng et al. (2018), which has been shown to capture important characteristics of real-world networks. Specifically, we assume that nodes are partitioned into three disjoint groups or communities, and conditioning on the community labels, subnetworks corresponding to the communities follow the GIRG model defined in Section 7.1 with $\gamma = \delta = 1$, $\beta = 3$ and $r = 10$. Edges between nodes in different communities are independently drawn with the same probability adjusted so that the ratio of the expected numbers of edges between communities and within communities equals $\rho \in \{0.01, 0.1\}$, which measures the strength of the community structure.

For each $\rho \in \{0.01, 0.1\}$, we consider three settings corresponding to different community size ratios $(1/20, 9/20, 1/2)$, $(1/10, 2/5, 1/2)$ and $(1/3, 1/3, 1/3)$. In the first two settings, network communities are of very different sizes, while in the last setting, all communities are of the same size $n/3$. The networks generated in these settings are relatively dense for the sampling purpose, with expected degrees ranging from 300 to 500. We vary the sample size m by setting $m = \tau n$, where τ is the sample size factor taking values from 10 to 210. To approximate the number of common neighbors for CNA, we sample $k = 50$ neighbors using Algorithm 1.

Figure 2 shows the relative error averaged over ten repetitions of CN, CNA, and UN in three settings and different values of ρ. We observe that as the sample size m increases, all methods perform better, with CN slightly better than CNA, and both methods are more accurate than UN when the communities are of different sizes and especially when $\rho = 0.01$. This is because when ρ is small, and one community is of much smaller size than the others, UN focuses on sampling edges within large communities, mostly ignoring edges between communities and within the smallest community. In contrast, CN and CNA sample more edges within the smallest community and between communities because they have fewer common neighbors, resulting in better estimates of the Laplacian. However, when all communities are of the same size, UN tends to perform better than CN and CNA. This is perhaps because no part of any balanced and dense network needs to be sampled much more frequently than others, and UN often performs well in this case Sadhanala et al. (2016). However, real networks are usually far from balanced, and UN may be much less

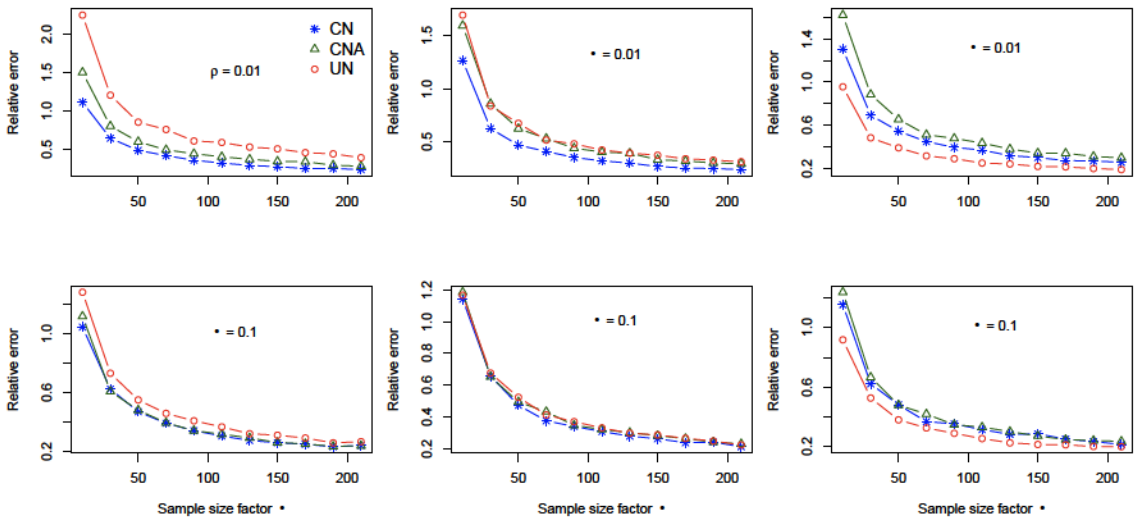

Figure 2: Accuracy of different sampling methods on random networks.

7.2.2 Real-world Networks

We further compare the performance of CN, CNA, and UN on the political blogs network Adamic and Glance (2005) and the Facebook ego network McAuley and Leskovec (2012), the two largest networks in Table 2 for which inverting the Laplacian can be done reasonably

fast. (Note that the matrix inversion is only needed for calculating the relative error in (7.1) while our proposed methods can easily handle all networks in Table 2.) Similar to the analysis in the previous section, we vary the sample size m by setting $m = \tau n$ with $\tau \in [10, 50]$. To approximate the number of common neighbors for CNA, we sample $k = 20$ neighbors according to Algorithm 1. Figure 3 shows that as τ increases, all three methods perform better, with CN slightly better than CNA, both having much smaller errors than UN.

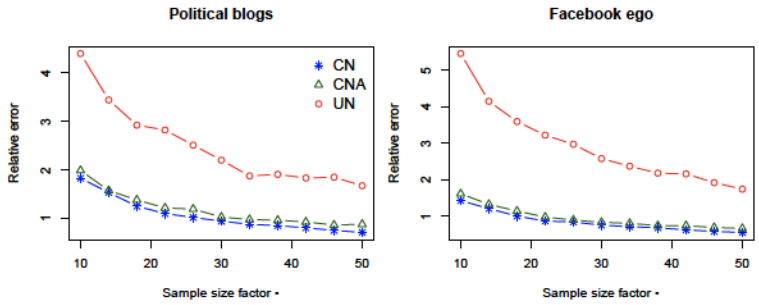

Figure 3: Accuracy of different sampling methods on real networks.

Note that the performance gap between UN and the proposed methods is much more visible on the real networks than on simulated networks shown in the previous section. This is probably due to the significant difference between the distributions of the number of common neighbors of simulated and real networks. As shown in Figure 4, most of the edges of the simulated networks have very large numbers of common neighbors, and for such nodes, the uniform sampling performs well (this is partially explained in the discussion following Theorem 2). In contrast, edges of the real networks considered here have relatively smaller numbers of common neighbors, resulting in the much worse performance of the uniform sampling. The numerical results on real networks show that sampling using the number of common neighbors may be very useful in practice.

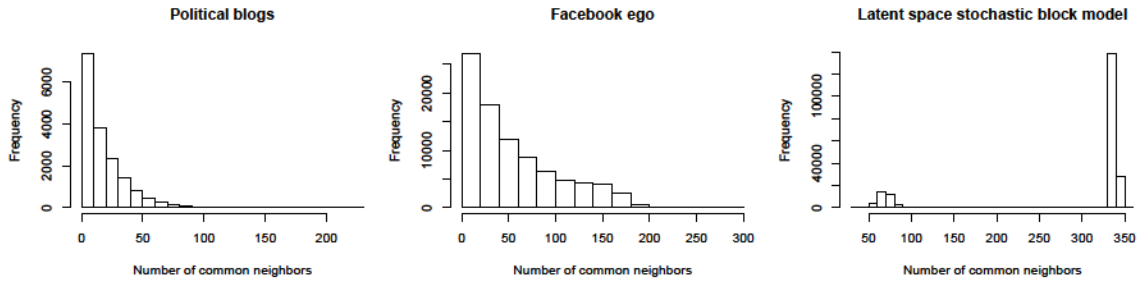

Figure 4: Distributions of the number of common neighbors for the political blogs network, the Facebook ego network, and a network generated from a latent space stochastic block model described in Section 7.2.1 with community size ratios $(1/3, 1/3, 1/3)$ and $\rho = 0.1$.

8. Discussion

In this paper, we study an edge sampling algorithm that uses only the number of common neighbors. This simple statistic provides an easy way to measure the strength of network local connectivity through parameter α, which directly controls the accuracy of the sampling method. However, in practice, we often have access to the numbers of common neighbors and neighborhood networks around edges. In that case, we should use the information from these local networks, provided that it is available or easily computed because it contains more structural information of the network than just the numbers of common neighbors. Measuring the strength of local connectivity through local networks is more challenging, and we leave it for future work.

Acknowledgments

We would like to thank the anonymous reviewers for their careful reading and valuable comments, which helped us improve the quality of the paper.

Appendix A. Proofs of Results in Section 2

Theorem 1 directly follows from Theorem 2 with $\hat{t}_{ij} = t_{ij}$ for all edges $(i, j) \in E_G$ and $C = 1$. To prove Theorem 2, we use the following result about the concentration of the sum of random matrices Vershynin (2009).

Theorem 10 (Concentration of sum of matrices) *Let $Y_1, ..., Y_m$ be independent $n \times n$ random positive semidefinite matrices such that $\|Y_k\| \leq M$ for all $1 \leq k \leq m$. Let $S_m = \sum_{k=1}^m Y_k$ and $E = \sum_{k=1}^m \|\mathbb{E}\, Y_k\|$. Then for every $\varepsilon \in (0, 1)$ we have*

$$\mathbb{P}\left\{\|S_m - \mathbb{E}\, S_m\| > \varepsilon E\right\} \leq n \cdot \exp\left(\frac{-\varepsilon^2 E}{4M}\right).$$

Proof [Proof of Theorem 2] Let X be a random matrix such that

$$X = \frac{1}{p_{ij}}(e_i - e_j)(e_i - e_j)^\mathsf{T} \quad \text{with probability } \hat{p}_{ij},$$

where $(i, j) \in E_G$, $\{e_i, 1 \leq i \leq n\}$ are standard basis vectors (the ith entry of e_i is one and all other entries are zero), and

$$\hat{p}_{ij} = \frac{\frac{2}{\hat{t}_{ij}+2}}{\sum_{(i,j)\in E_G} \frac{2}{\hat{t}_{ij}+2}}. \tag{A.1}$$

Then

$$\mathbb{E}\, X = \sum_{(i,j)\in E_G} \hat{p}_{ij} \times \frac{1}{\hat{p}_{ij}}(e_i - e_j)(e_i - e_j)^\mathsf{T} = L_G. \tag{A.2}$$

Let X_k be m independent copies of X. By the sampling scheme we have

$$L_H = \frac{1}{m}\sum_{k=1}^m X_k, \quad \mathbb{E}\, L_H = L_G.$$

Denote by L_G^{-1} the Moore-Penrose pseudoinverse of L_G and by $L_G^{-1/2}$ the squared root of L_G^{-1}. Note that the kernel of the map L_G is a one-dimensional vector space spanned by the all-one vector $\mathbf{1}$ and it is contained in the kernel of L_H. Therefore the strong spectral property (1.1) is equivalent to

$$(1-\varepsilon)I_{\mathbf{1}} \preceq \frac{1}{m}\sum_{k=1}^{m} L_G^{-1/2} X_k L_G^{-1/2} \preceq (1+\varepsilon)I_{\mathbf{1}}, \tag{A.3}$$

where $I_{\mathbf{1}} = I - (1/n)\mathbf{1}\mathbf{1}^{\mathsf{T}}$ is the identity map on the $(n-1)$-dimensional subspace orthogonal to the all-one vector $\mathbf{1}$. Here, we write $U \preceq V$ if $V - U$ is positive semidefinite.

To prove (A.3), we apply Theorem 10 to $Y_k := L_G^{-1/2} X_k L_G^{-1/2}$. Since $X_k \succeq 0$ and $\mathbb{E}\, X_k = L_G$ by (A.2), it follows that $Y_k \succeq 0$ and $\|\mathbb{E}\, Y_k\| = \|I_{\mathbf{1}}\| = 1$. To bound $\|Y_k\|$, note that Y_k takes one of the following matrix values

$$\frac{1}{\hat{p}_{ij}} \left(L_G^{-1/2}(e_i - e_j)\right)\left(L_G^{-1/2}(e_i - e_j)\right)^{\mathsf{T}}, \quad (i,j) \in E_G.$$

By (A.1) and (2.3) we have $1/\hat{p}_{ij} = n\hat{\alpha}(\hat{t}_{ij}+2)/2$. Therefore

$$\|Y_k\| \le \max_{(i,j)\in E_G} \frac{n\hat{\alpha}(\hat{t}_{ij}+2)}{2} \cdot (e_i - e_j)^{\mathsf{T}} L_G^{-1}(e_i - e_j). \tag{A.4}$$

Note that $(e_i - e_j)^{\mathsf{T}} L_G^{-1}(e_i - e_j)$ is the effective resistance of the edge between i and j Ghosh et al. (2008). We claim that it is upper bounded by $2/(t_{ij}+2)$. To show that, let N_{ij} be the set of common neighbors of i and j. Denote by $G_{ij} = (V_{ij}, E_{ij})$ the subgraph of G such that

$$V_{ij} = \{i,j\} \cup N_{ij}, \qquad E_{ij} = \{(i,j), (i,k), (j,k) : k \in N_{ij}\}.$$

Thus, G_{ij} consists of an edge and t_{ij} paths of length two between i and j. It is easy to see that the effective resistance $(e_i - e_j)^{\mathsf{T}} L_{G_{ij}}^{-1}(e_i - e_j)$ of the edge between i and j in G_{ij} is $2/(t_{ij}+2)$. Indeed, let $x = L_{G_{ij}}^{-1}(e_i - e_j)$. Then $L_{G_{ij}} x = e_i - e_j$ and by comparing the i-th and j-th components of $L_{G_{ij}} x$ and $e_i - e_j$, we have

$$(t_{ij}+1)x_i - x_j - \sum_{k\in N_{ij}} x_k = 1, \qquad x_i - (t_{ij}+1)x_j + \sum_{k\in N_{ij}} x_k = 1.$$

Adding these equalities, we obtain that the effective resistance of the edge between i and j in G_{ij} is $(e_i - e_j)^{\mathsf{T}} x = x_i - x_j = 2/(t_{ij}+2)$. Since G_{ij} is a subgraph of G and adding edges does not increase the effective resistance (see e.g. Corollary 9.13 in Levin and Peres (2017)), it follows that

$$(e_i - e_j)^{\mathsf{T}} L_G^{-1}(e_i - e_j) \le (e_i - e_j)^{\mathsf{T}} L_{G_{ij}}^{-1}(e_i - e_j) = \frac{2}{t_{ij}+2}.$$

Together with (A.4) this implies $\|Y_k\| \le n\hat{\alpha}(\hat{t}_{ij}+2)/(t_{ij}+2) \le n\hat{\alpha}C$. Therefore by Theorem 10 we have

$$\mathbb{P}\left\{\left\|\frac{1}{m}\sum_{k=1}^{m} Y_k - I_{\mathbf{1}}\right\| > \varepsilon\right\} \le n \cdot \exp\left(\frac{-\varepsilon^2 m}{4C\hat{\alpha}n}\right).$$

Inequality (A.3) then follows by choosing $m = 8C\hat{\alpha}n\log n/\varepsilon^2$. ∎

Appendix B. Proofs of Results in Section 3

Proof [Proof of Theorem 4] Let (i, j) be an edge of G such that $\kappa(i, j) \geq \kappa_0$ or equivalently $W_1(m_i, m_j) \leq 1 - \kappa_0$. Recall that

$$W_1(m_i, m_j) = \inf_{\xi \in \Pi(m_i, m_j)} \sum_{(k,k') \in V \times V} d(k, k')\xi(k, k'),$$

where $\Pi(m_i, m_j)$ is the set of all probability measures on $V \times V$ with marginals m_i and m_j. For every $\xi \in \Pi(m_i, m_j)$,

$$\xi(k, k) \leq \min \left\{ \sum_{k' \in V} \xi(k', k), \sum_{k' \in V} \xi(k, k') \right\} = \min\{m_j(k), m_i(k)\}.$$

Since m_i and m_j are uniform measures with supports being the sets of neighbors of i and j, respectively, if k is not a common neighbor of i and j then $\min\{m_j(k), m_i(k)\} = 0$; when k is one of the t_{ij} common neighbors of i and j then $\min\{m_j(k), m_i(k)\} = \min\{1/d_i, 1/d_j\}$. Therefore

$$\sum_{k \in V} \xi(k, k) \leq t_{ij} \cdot \min\{1/d_i, 1/d_j\},$$

which implies

$$\begin{aligned} \sum_{(k,k') \in V \times V} d(k, k')\xi(k, k') &= \sum_{k \neq k'} d(k, k')\xi(k, k') \\ &\geq \sum_{k \neq k'} \xi(k, k') \\ &= 1 - \sum_{k \in V} \xi(k, k) \\ &\geq 1 - t_{ij} \cdot \min\{1/d_i, 1/d_j\}. \end{aligned}$$

Taking the infimum over all $\xi \in \Pi(m_i, m_j)$, we have

$$1 - \kappa_0 \geq W_1(m_i, m_j) \geq 1 - t_{ij} \cdot \min\{1/d_i, 1/d_j\},$$

or $t_{ij} \geq \kappa_0 / \min\{1/d_i, 1/d_j\}$. Denote by $\mathcal{E}$ the set of all edges of G such that $\kappa(i, j) \geq \kappa_0$ and by $\mathcal{E}^c$ its complement. Since $|\mathcal{E}^c| \leq Cn$ by assumption,

$$\begin{aligned} \alpha &= \frac{1}{n} \sum_{(i,j) \in \mathcal{E}} \frac{2}{2 + t_{ij}} + \frac{1}{n} \sum_{(i,j) \in \mathcal{E}^c} \frac{2}{2 + t_{ij}} \\ &\leq \frac{2}{\kappa_0 n} \sum_{(i,j) \in \mathcal{E}} \min\{1/d_i, 1/d_j\} + C \\ &\leq \frac{1}{\kappa_0 n} \sum_{i \in V} \sum_{j \in N_i} \min\{1/d_i, 1/d_j\} + C \\ &\leq \frac{1}{\kappa_0} + C. \end{aligned}$$

For the last inequality, we use the fact that $\sum_{j\in N_i} \min\{1/d_i, 1/d_j\} \le 1$. ■

For proving Theorem 3, we need the following lemma.

Lemma 11 *For positive numbers $x_1, x_2, ..., x_k$ the following inequality holds*

$$(x_1 + x_2 + \cdots + x_k)\left(\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_k}\right) \ge k^2.$$

The two sides are equal if and only if $x_1 = x_2 = \cdots = x_n$.

Proof [Proof of Lemma 11] Using the inequality of arithmetic and geometric means, we have

$$x_1 + x_2 + \cdots + x_k \ge k(x_1 x_2 \cdots x_k)^{1/k}, \quad \frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_k} \ge k(x_1 x_2 \cdots x_k)^{-1/k}.$$

Lemma 11 follows directly from these inequalities. ■

Proof [Proof of Theorem 3] For each node i, denote by N_i and t_i the set of neighbors of i and the number of triangles that contain i, respectively. Using Lemma 11, we have

$$\sum_{j\in N_i} \frac{2}{t_{ij}+2} \ge \frac{2|N_i|^2}{\sum_{j\in N_i}(t_{ij}+2)} = \frac{d_i^2}{t_i + d_i} \ge \frac{1}{2c_i + \frac{1}{d_i}}.$$

Summing over all nodes i and applying Lemma 11 again, we obtain

$$\sum_{(i,j)\in E} \frac{4}{t_{ij}+2} \ge \sum_{i\in V} \frac{1}{2c_i + \frac{1}{d_i}} \ge \frac{|V|^2}{\sum_{i\in V}\left(2c_i + \frac{1}{d_i}\right)} = \frac{n}{2c + \frac{1}{n}\sum_{i\in V}\frac{1}{d_i}}.$$

The proof is complete by dividing both sides of this inequality by $2n$. ■

Appendix C. Proofs of Results in Section 4

Proof [Proof of Theorem 5] We rewrite $S := n\alpha$ as follows:

$$S = \sum_{(i,j)\in E_G} \frac{2}{t_{ij}+2} = \sum_{i<j} \frac{2A_{ij}}{t_{ij}+2} =: \sum_{i<j} Y_{ij}.$$

The proof consists of two parts: showing that $S \le 2\,\mathbb{E}\,S$ with high probability and upper bounding $\mathbb{E}\,S$. For the first part, note that S is a sum of $n(n-1)/2$ weakly depdendent random variables $Y_{ij} = 2A_{ij}/(t_{ij}+2)$, where Y_{ij} and $Y_{i'j'}$ are independent if $i' \ne i$ and $j' \ne j$. To deal with the dependence among Y_{ij}, we will use the moment method (see for example Warnke (2017)).

For notational simplicity, we denote $\gamma = \{i, j\}$ as a set of two elements i, j and write $S = \sum_\gamma Y_\gamma$. With the new notation, Y_γ and $Y_{\gamma'}$ are independent if $\gamma \cap \gamma' = \emptyset$. For a positive integer k, let

$$M_k = \sum_{(\gamma_1,...,\gamma_k)} \prod_{i=1}^{k} Y_{\gamma_i},$$

where the sum is over all k-tuples $(\gamma_1, ..., \gamma_k)$ such that $\gamma_i \cap \gamma_j = \emptyset$ if $i \neq j$. Since $Y_{\gamma_1}, Y_{\gamma_2}, ..., Y_{\gamma_k}$ are independent by construction,

$$\mathbb{E}\, M_k = \sum_{(\gamma_1,...,\gamma_k)} \prod_{i=1}^{k} \mathbb{E}\, Y_{\gamma_i} \leq \left(\sum_\gamma \mathbb{E}\, Y_\gamma \right)^k = (\mathbb{E}\, S)^k . \tag{C.1}$$

Denote by $\mathcal{E}$ the event that $S > 2\,\mathbb{E}\, S$. When $\mathcal{E}$ occurs,

$$\begin{aligned} M_{k+1} &= \sum_{(\gamma_1,...,\gamma_k)} \prod_{i=1}^{k} Y_{\gamma_i} \left(\sum_\gamma Y_\gamma - \sum_{\gamma \cap \gamma_i \neq \emptyset \text{ for some } 1 \leq i \leq k} Y_\gamma \right) \\ &\geq \sum_{(\gamma_1,...,\gamma_k)} \prod_{i=1}^{k} Y_{\gamma_i} \left(2\,\mathbb{E}\, S - \sum_{\gamma \cap \gamma_i \neq \emptyset \text{ for some } 1 \leq i \leq k} Y_\gamma \right). \end{aligned}$$

For each $(\gamma_1, ..., \gamma_k)$ we have

$$\sum_{\gamma \cap \gamma_i \neq \emptyset \text{ for some } 1 \leq i \leq k} Y_\gamma \leq \sum_{i=1}^{k} \sum_{\gamma \cap \gamma_i \neq \emptyset} Y_\gamma \leq k \cdot \max_{\gamma'} \sum_{\gamma \cap \gamma' \neq \emptyset} Y_\gamma.$$

Let $Z = \max_i \sum_{j=1}^n A_{ij}$ be the maximal node degree. Since $Y_\gamma \leq A_\gamma$,

$$\max_{\gamma'} \sum_{\gamma \cap \gamma' \neq \emptyset} Y_\gamma \leq \max_{\gamma'} \sum_{\gamma \cap \gamma' \neq \emptyset} A_\gamma \leq 2Z.$$

Therefore if $\mathcal{E}$ occurs then

$$M_{k+1} \geq \sum_{(\gamma_1,...,\gamma_k)} \prod_{i=1}^{k} Y_{\gamma_i} \left(2\,\mathbb{E}\, S - 2kZ \right) = M_k \left(2\,\mathbb{E}\, S - 2kZ \right). \tag{C.2}$$

We now show that $2\,\mathbb{E}\, S - 2kZ \geq (3/2) \cdot \mathbb{E}\, S > 0$ with high probability for $k = O(\log n)$, so the above inequality can be applied repeatedly to obtain a desired lower bound for M_k. Since upper diagonal elements of A are independent, by Jensen's inequality we have

$$\mathbb{E}\, S = \sum_{i<j} P_{ij} \cdot \mathbb{E} \left[\frac{2}{t_{ij} + 2} \right] \geq \sum_{i<j} \frac{2P_{ij}}{\mathbb{E}\, t_{ij} + 2} \geq \frac{2}{\max_{i,j} \mathbb{E}\, t_{ij} + 2} \cdot \sum_{i<j} P_{ij}.$$

The second inequality of (4.1) then implies

$$\mathbb{E}\, S \geq C \Delta \log n.$$

Let $\mathcal{E}_1$ be the event that $Z \le 2\Delta$. Since $\Delta > C \log n$, it follows from the Chernoff and union bounds that

$$\mathbb{P}(\mathcal{E}_1) = \mathbb{P}(Z \le 2\Delta) \ge 1 - \frac{1}{2n}. \tag{C.3}$$

When $\mathcal{E}_1$ occurs,

$$2\,\mathbb{E}\,S - 2kZ = \frac{3\,\mathbb{E}\,S}{2} + \frac{\mathbb{E}\,S}{2} - 2kZ \ge \frac{3\,\mathbb{E}\,S}{2} + \frac{C\Delta \log n}{2} - 4k\Delta \ge \frac{3\,\mathbb{E}\,S}{2}$$

for all $k \le m$, where $m = \lfloor (C/8) \log n \rfloor$ is the largest integer not greater than $(C/8) \log n$. Therefore if $\mathcal{E} \cap \mathcal{E}_1$ occurs then $M_1 = S > 2\,\mathbb{E}\,S$ and by applying (C.2) repeatedly,

$$M_m \ge M_{m-1} \cdot \frac{3\,\mathbb{E}\,S}{2} \ge \cdots \ge \left(\frac{3\,\mathbb{E}\,S}{2}\right)^m.$$

Using Markov's inequality, (C.1) and (C.3), we have

$$\begin{aligned}
\mathbb{P}(S > 2\,\mathbb{E}\,S) &\le \mathbb{P}(S > 2\,\mathbb{E}\,S, Z \le 2\Delta) + \mathbb{P}(Z > 2\Delta) \\
&\le \mathbb{P}\left(M_m \ge \left\lceil \frac{3\,\mathbb{E}\,S}{2} \right\rceil^m\right) + \frac{1}{2n} \\
&\le \left(\frac{2}{3}\right)^m + \frac{1}{2n} \\
&\le \frac{1}{n},
\end{aligned}$$

where the last inequality holds for sufficiently large C. Thus, $S \le 2\,\mathbb{E}\,S$ with probability at least $1 - 1/n$.

It remains to bound

$$\mathbb{E}\,S = \sum_{i<j} 2P_{ij} \cdot \mathbb{E}\left[\frac{1}{t_{ij}+2}\right].$$

For every pair of nodes (i, j) we have

$$\left|\mathbb{E}\,\frac{1}{t_{ij}+2} - \frac{1}{\mathbb{E}\,t_{ij}+2}\right| = \frac{1}{\mathbb{E}\,t_{ij}+2} \cdot \left|\mathbb{E}\,\frac{t_{ij} - \mathbb{E}\,t_{ij}}{t_{ij}+2}\right| \le \frac{1}{\mathbb{E}\,t_{ij}+2} \cdot \mathbb{E}\,\frac{|t_{ij} - \mathbb{E}\,t_{ij}|}{t_{ij}+2}. \tag{C.4}$$

Since t_{ij} is the sum of n independent Bernoulli random variables, by Chernoff bound,

$$\mathbb{P}\left(t_{ij} \le \mathbb{E}\,t_{ij}/2\right) \le \exp\left(-\frac{\mathbb{E}\,t_{ij}}{8}\right).$$

Consider the function $g(x) = |x - \mathbb{E}\,t_{ij}|/(x+2)$ with $x \ge 0$. It is easy to show that $g(x) \le \mathbb{E}\,t_{ij}$ if $x \le \mathbb{E}\,t_{ij}/2$ and $g(x) \le 1$ if $x > \mathbb{E}\,t_{ij}/2$. Therefore

$$\begin{aligned}
\mathbb{E}\,\frac{|t_{ij} - \mathbb{E}\,t_{ij}|}{t_{ij}+2} &= \mathbb{E}\,g(t_{ij}) \cdot I(t_{ij} \le \mathbb{E}\,t_{ij}/2) + \mathbb{E}\,g(t_{ij}) \cdot I(t_{ij} > \mathbb{E}\,t_{ij}/2) \\
&\le \mathbb{E}\,t_{ij} \cdot \mathbb{P}(t_{ij} \le \mathbb{E}\,t_{ij}/2) + \mathbb{P}(t_{ij} > \mathbb{E}\,t_{ij}/2) \\
&\le \mathbb{E}\,t_{ij} \cdot \exp\left(-\frac{\mathbb{E}\,t_{ij}}{8}\right) + 1 \\
&\le 4.
\end{aligned}$$

The last inequality follows from the fact that $x \cdot \exp(-x/8) \leq 3$ for all $x \geq 0$. By (C.4) and the last inequality, we get

$$\mathbb{E}\,\frac{1}{t_{ij}+2} \leq \frac{5}{\mathbb{E}\,t_{ij}+2}.$$

Finally,

$$\mathbb{E}\,S = \sum_{i<j} 2P_{ij} \cdot \mathbb{E}\left[\frac{1}{t_{ij}+2}\right] \leq \sum_{i<j} \frac{10P_{ij}}{\mathbb{E}\,t_{ij}+2}$$

and the proof is complete. ■

Proof [Proof of Corollary 6] We first verify the conditions in Theorem 5. By (4.3),

$$\Delta = \max_i \sum_{j=1}^n P_{ij} = \max_i n_i\delta \geq C^{-1}nr_n^d\delta \geq C\log n.$$

Therefore the first condition of (4.1) is satisfied if C is sufficiently large. Also, since

$$\Delta \cdot \max_{i,j}\left[1 + \left(P^2\right)_{ij}\right] = \max_i n_i\delta \cdot \max_{ij}(1 + n_{ij}\delta^2) \leq 2Cn^2r_n^{2d}\delta^3$$

and

$$\frac{1}{\log n}\sum_{i<j} P_{ij} = \frac{1}{2\log n}\sum_{i=1}^n n_{ij}\delta \geq \frac{n^2r_n^d\delta}{2C\log n},$$

the second condition of (4.1) holds because $r_n^d \leq (4C^3\delta^2\log n)^{-1}$. Therefore by Theorem 5, with probability at least $1 - 1/n$, the upper bound of α is

$$\begin{aligned}
\frac{1}{n}\sum_{i<j}\frac{10P_{ij}}{\mathbb{E}\,t_{ij}+2} &= \frac{1}{n}\sum_{i=1}^n \sum_{j:\|x_i-x_j\|\leq r_n} \frac{5\delta}{n_{ij}\delta^2+2} \\
&\leq \frac{1}{n}\sum_{i=1}^n \frac{5n_i\delta}{n_{ij}\delta^2+2} \\
&\leq 5C^2\delta^{-1},
\end{aligned}$$

and the proof is complete. ■

Appendix D. Proof of Results in Section 5

Proof [Proof of Lemma 7] By the definition of E and $\mathcal{E}$, we have

$$\sum_{(i,j)\in E} \tilde{t}_{ij}^{-1} \leq \sum_{e\in\mathcal{E}} \sum_{\{i,j\}\subseteq e} \tilde{t}_{ij}^{-1}.$$

Since $\tilde{t}_{ij} \geq |e|$ for each $e \in E_{\mathcal{G}}$ that contains $\{i,j\}$ and there are $|e|(|e|-1)/2$ pairs $\{i,j\} \in e$, it follows from above inequality that

$$\sum_{(i,j)\in E} \tilde{t}_{ij}^{-1} \leq \sum_{e\in\mathcal{E}} \frac{|e|-1}{2} \leq \frac{1}{2}\sum_{e\in\mathcal{E}} |e| \leq \frac{dn}{2}.$$

For the last inequality we use the assumption that each node belongs to at most d hyperedges. ■

Proof [Proof of Theorem 8] The proof of this theorem is similar to the proof of Theorem 1 with one exception that we replace $\tilde{\alpha}$ with the upper bound $d/2$ shown in Lemma 7. ■

Appendix E. Proof of Results in Section 6

Proof [Proof of Theorem 9] Let S be the sum of k independent Bernoulli random variables with success probability θ. Then by Bernstein's inequality, for any $\delta \in [0,1]$,

$$\mathbb{P}\left(|S - k\theta| > \delta k\theta\right) \leq 2\exp\left(\frac{-(\delta k\theta)^2/2}{k\theta + (\delta k\theta)/3}\right) \leq 2\exp\left(\frac{-\delta^2 k\theta}{4}\right). \tag{E.1}$$

Let $\mathcal{E}$ be the set of all edges with $\theta_{ij} < \varepsilon/2$. For each $(i,j) \in \mathcal{E}$, $k\hat{\theta}_{ij}$ is stochastically bounded by S with $\theta = \varepsilon/2$. Therefore by (E.1) with $\delta = \varepsilon$,

$$\mathbb{P}(\hat{\theta}_{ij} \geq \varepsilon) \leq \mathbb{P}(S \geq k\varepsilon) \leq \mathbb{P}(|S - k\varepsilon/2| \geq k\varepsilon/2) \leq 2\exp\left(\frac{-k\varepsilon}{8}\right).$$

Since $k = 100\log n/\varepsilon$, by the union bound,

$$\mathbb{P}\left(\hat{\theta}_{ij} < \varepsilon \text{ for all } (i,j) \in \mathcal{E}\right) \geq 1 - 2n^2\exp\left(\frac{-k\varepsilon}{8}\right) \geq 1 - \frac{1}{2n}. \tag{E.2}$$

Consider now $\mathcal{E}^c$, the set of nodes with $\theta_{ij} \geq \varepsilon/2$. Then using (E.1) with $\theta = \theta_{ij}$ and $\delta = 1/2$, we get

$$\mathbb{P}(|\hat{\theta}_{ij} - \theta_{ij}| \geq \theta_{ij}/2) \leq \mathbb{P}(|S - k\theta_{ij}| \geq k\theta_{ij}/2) \leq 2\exp\left(\frac{-k\theta_{ij}}{16}\right) \leq 2\exp\left(\frac{-k\varepsilon}{32}\right).$$

Therefore with $k = 100\log n/\varepsilon^3$, we have

$$\begin{aligned}
\mathbb{P}\left(\left|\frac{\hat{t}_{ij}}{t_{ij}} - 1\right| \leq \frac{1}{2} \text{ for all } (i,j) \in \mathcal{E}^c\right) &= \mathbb{P}\left(\left|\frac{\hat{\theta}_{ij}}{\theta_{ij}} - 1\right| \leq \frac{1}{2} \text{ for all } (i,j) \in \mathcal{E}^c\right) \\
&\leq 2n^2\exp\left(\frac{-k\varepsilon}{32}\right) \\
&\leq 1 - \frac{1}{2n}.
\end{aligned} \tag{E.3}$$

Recall that for edges (i,j) with $\min\{d_i,d_j\} > k$, we calculate t_{ij} directly if $\hat{\theta}_{ij} < \varepsilon$ and estimate t_{ij} by $\hat{t_{ij}} = \hat{\theta}_{ij} \cdot \min\{d_i,d_j\}$ otherwise. From (E.2) and (E.3), we obtain that $|\hat{t}_{ij} - t_{ij}| \leq 1/2$ for all (i,j) with probability at least $1 - 1/n$. The spectral property (1.1) then follows from Theorem 2.

The computational complexity of estimating all t_{ij} is bounded by

$$\sum_{(i,j)\in\mathcal{E}} \min\{d_i,d_j\} + \sum_{(i,j)\in\mathcal{E}^c} k \leq \sum_{(i,j):\theta_{ij}\leq\varepsilon/2} \min\{d_i,d_j\} + 100|E_G|\log n/\varepsilon.$$

The proof is complete. ■

References

L. A. Adamic and N. Glance. The political blogosphere and the 2004 US election. In *Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem*, 2005.

Sameer Agarwal, Kristin Branson, and Serge Belongie. Higher order learning with graphs. *ICML '06*, pages 17–24, 2006.

Noga Alon, Itai Benjamini, and Alan Stacey. Percolation on finite graphs and isoperimetric inequalities. *Ann. Probab.*, 32(3):1727–1745, 2004.

Shaikh Arifuzzaman, Maleq Khan, and Madhav Marathe. Fast parallel algorithms for counting and listing triangles in big graphs. *ACM Transactions on Knowledge Discovery from Data*, 14(1):5:1–5:34, 2019.

Frank Bauer, J Urgen, and Shiping Liu. Ollivier-Ricci curvature and the spectrum of the normalized graph Laplace operator. *Math. Res. Lett.*, 19(6):1185–1205, 2012.

Luca Becchetti, Paolo Boldi, Carlos Castillo, and Aristides Gionis. Efficient semi-streaming algorithms for local triangle counting in massive graphs. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–24, 2008.

M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. *COLT*, 3120:624–638, 2004.

András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\tilde{O}$(n2) time. *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing (STOC '96)*, pages 47–55, 1996.

Bhaswar B. Bhattacharya and Sumit Mukherjee. Exact and asymptotic results on coarse Ricci curvature of graphs. *Discrete Mathematics*, 338(6):23–42, 2015.

B. Bollobas, S. Janson, and O. Riordan. The phase transition in inhomogeneous random graphs. *Random Structures and Algorithms*, 31:3–122, 2007.

Béla Bollobás, Christian Borgs, Jennifer Chayes, and Oliver Riordan. Percolation on dense graph sequences. *Ann. Probab.*, 38(1):150–183, 2010.

Karl Bringmann, Ralph Keusch, and Johannes Lengler. Sampling Geometric Inhomogeneous Random Graphs in Linear Time. In *25th Annual European Symposium on Algorithms (ESA 2017)*, volume 87 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.

Andrei Z. Broder. On the resemblance and containment of documents. *Proceeding of Compression and Complexity of Sequences*, pages 21–29, 1997.

Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29:1157–1166, 1997.

S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

Arpita Ghosh, Stephen Boyd, and Amin Saberi. Minimizing effective resistance of a graph. *SIAM Review*, 1(50):37–66, 2008.

Ashish Goel, Michael Kapralov, and Sanjeev Khanna. Graph sparsification via refinement sampling. *arXiv:1004.4915*, 2010.

A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2:129–233, 2010.

Rishi Gupta, Tim Roughgarden, and C. Seshadhri. Decompositions of triangle-dense graphs. *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 471–482, 2014.

Michael Hamann, Gerd Lindner, Henning Meyerhenke, Christian L. Staudt, and Dorothea Wagner. Structure-preserving sparsification methods for social networks. *Social Network Analysis and Mining*, 6(22):1–22, 2016.

Jian Huang, Shuangge Ma, Hongzhe Li, and Cun Hui Zhang. The sparse laplacian shrinkage estimator for high-dimensional regression. *Annals of Statistics*, 39(4):2021–2046, 2011.

Jürgen Jost and Shiping Liu. Ollivier's Ricci curvature, local clustering and curvature-dimension inequalities on graphs. *Discrete and Computational Geometry*, 51(2):300–322, 2014.

Michael Kapralov, Yin Tat Lee, Cameron Musco, Christopher Musco, and Aaron Sidford. Single pass spectral sparsification in dynamic streams. *SIAM J. Comput.*, 46(1):456–477, 2014.

Jonathan Kelner and Alex Levin. Spectral sparsification in the semi-streaming setting. *Theory of Computing Systems*, 53(2):243–262, 2013.

Alisa Kirichenko and Harry van Zanten. Estimating a smooth function on a large graph by bayesian laplacian regularisation. *Electron. J. Statist.*, 11(1):891–915, 2017.

Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *Boulicaut JF., Esposito F., Giannotti F., Pedreschi D. (eds) Machine Learning: ECML 2004*, volume 3201, pages 217–226, 2004.

Can M. Le and Tianxi Li. Linear regression and its inference on noisy network-linked data. *arXiv:2007.00803*, 2020.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (ACM TKDD)*, 1(1):2007, 2007.

David A. Levin and Yuval Peres. *Markov Chains and Mixing Times*. American Mathematical Society, 2017.

Tianxi Li, Elizaveta Levina, and Ji Zhu. Network cross-validation by edge sampling. *Biometrika*, 107(2):257–276, 2020.

Yong Lin, Linyuan Lu, and S. T. Yau. Ricci-flat graphs with girth at least five. *Communications in Analysis and Geometry*, 22(4):671–687, 2014.

D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of doubtful sound features a large propor- tion of long-lasting associations. can geographic isola- tion explain this unique trait? *Behavioral Ecology and Sociobiology*, 54:396–405, 2003.

Julian McAuley and Jure Leskovec. Learning to discover social circles in ego networks. *Neural Information Processing Systems (NIPS)*, 2012.

Asaf Nachmias. Percolation on dense graph sequences. *Geometric and Functional Analysis*, 19(4):1171–1194, 2010.

M. E. J. Newman. *Networks: An introduction*. Oxford University Press, 2010.

Tin Lok James Ng, Thomas Brendan Murphy, Ted Westling, Tyler H. McCormick, and Bailey K. Fosdick. Modeling the social media relationships of Irish politicians using a generalized latent space stochastic blockmodel. *arXiv:1807.06063*, 2018.

Arlind Nocaj, Mark Ortmann, and Ulrik Brandes. Untangling hairballs – From 3 to 14 degrees of separation. *Duncan CA, Symvonis A (eds) Graph Drawing–22nd international symposium, GD 2014, Würzburg, Germany, September 24-26, 2014*, pages 101–112, 2014.

Roberto Oliveira. Concentration of the adjacency matrix and of the laplacian in random graphs with independent edges. *arXiv:0911.0600*, 2010.

Yann Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, 2009.

Fragkiskos Papadopoulos, Rodrigo Aldecoa, and Dmitri Krioukov. Network geometry inference using common neighbors. *Physical Review E*, 92(2):022807, 2015.

J. A. Rodríguez. On the Laplacian eigenvalues and metric parameters of hypergraphs. *Linear and Multilinear Algebra*, 50(1):1–14, 2002.

Karl Rohe and Tai Qin. The blessing of transitivity in sparse and stochastic networks. *arXiv:1307.2302*, 2013.

Veeranjaneyulu Sadhanala, YuXiang Wang, and Ryan J. Tibshirani. Graph sparsification approaches for laplacian smoothing. *AISTATS*, pages 1250–1259, 2016.

Venu Satuluri, Srinivasan Parthasarathy, and Yiye Ruan. Local graph sparsification for scalable clustering. *Proceedings of the 2011 international conference on Management of data (SIGMOD'11)*, pages 721–732, 2011.

Anshumali Shrivastava and Ping Li. Densifying one permutation hashing via rotation for fast near neighbor search. *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 557–565, 2014.

Anshumali Shrivastava and Ping Li. Asymmetric minwise hashing for indexing binary inner products and set containment. *Proceedings of the 24th International Conference on World Wide Web (WWW'15)*, pages 981–991, 2015.

A.J. Smola and R. Kondor. Kernels and regularization on graphs. *Learning Theory and Kernel Machines*, 2777:144–158, 2003.

D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. *Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing (STOC-04)*, pages 81–90, 2004.

Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011.

Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *arXiv:1111.4503*, 2011.

R. Vershynin. A note on sums of independent random matrices after Ahlswede-Winter. *Available at* `http://www-personal.umich.edu/~romanv/teaching/reading-group/ahlswede-winter.pdf`, 2009.

Lutz Warnke. Upper tails for arithmetic progressions in random subsets. *Israel Journal of Mathematics*, 1(221):317–365, 2017.

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393: 440, 1998.

J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *2012 IEEE 12th International Conference on Data Mining*, pages 745–754, 2012.

Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.