Measuring Edge Sparsity on Large Social Networks

J. David Smith
University of Florida
emallson@ufl.edu

My T. Thai
University of Florida
mythai@cise.ufl.edu

Abstract

How strong are the connections between individuals? This is a fundamental question in the study of social networks. In this work, we take a topological view rooted in the idea of local sparsity to answer this question on large social networks to which we have only incomplete access. Prior approaches to measuring network structure are not applicable to this setting due to the strict limits on data availability. Therefore, we propose a new metric, the *Edgecut Weight*, for this task. This metric can be calculated efficiently in an online fashion, and we empirically show that it captures important elements of communities. Further, we demonstrate that the distribution of these weights characterizes connectivity on a network. Subsequently, we estimate the distribution of weights on Twitter and show both a lack of strong connections and a corresponding lack of community structure.

Introduction

Historically, the social networks available to the research community had been small, manually collected datasets comprised of individuals interacting—usually in physical spaces. The rise of online social networks over the past two decades has changed this, not only increasing the scale but also the availability of data. In these halcyon days, network data was (relatively) freely available to researchers—leading to works that were able to study social networks in their entirety (for example: Twitter; Kwak et al. 2010).

Studies on the scale of Kwak et al. are, unfortunately, no longer possible. The social network giants have locked down their systems to prevent malicious data collection, and in doing so have prevented collection of complete network data by researchers. The inability to collect more data does not eliminate our research questions, however. We are thus motivated to seek alternate methods to study the topology of social networks.

In this work, we propose a novel, purely-local metric to calculate *edge sparsity*—a local quantity that measures how well-connected the endpoints of an edge are. In contrast to existing metrics, this *Edgecut Weight* requires relatively little data and in particular **can be calculated directly via social network APIs.** After exploring the empirical and theoretical connections this quantity has to existing work, we exploit this unique property to calculate hundreds of weights directly on Twitter.

We compare these weights to a number of other networks. While we do see a core-periphery structure on many networks (which agrees with Leskovec et al. 2009), we *do not* observe this on either the complete Twitter topology from 2010 or our new data collected in December 2019–January 2020. In fact, we find that this follower network is overwhelmingly sparse, leading us to question whether communities (as commonly defined in terms of network topology) occur on this network at all.

Contributions. Our contributions can be summarized as:

1. We develop a novel random-walk-based quantity, the *Edgecut Weight*, that measures edge sparsity. We then show how it can be efficiently calculated—even when network data is incomplete.
2. We study the empirical and theoretical connections between Edgecut Weights and prior work. We find that it assigns high weights to edges that cross between communities. Further, we compare it to Edge Betweenness, which has previously been used for similar purposes, and find that despite theoretical differences the two metrics have substantial correlation.
3. We apply our metric to study a range of networks. In particular, our visualizations clearly show widespread core-periphery structure, in agreement with prior work. We then construct a visualization using weights calculated directly via the Twitter API, and find that this network exhibits a striking lack of dense connections.

Related Works

One of the earliest metrics to be used in the study of network structure was the *conductance*, which measures the ratio of edges leaving a group to edges incident to the group. This metric sees wide application in the study of real-world social networks (Leskovec et al. 2009) and has deep ties to the theory of random walks (Sinclair 1988). However, it is also infeasible to calculate in a global sense—the conductance of a graph is the minimum conductance of any subset of the

graph's vertices—and requires the user to construct a graph cut *a priori* to be used in a local context.

Nonetheless it has seen some use in characterizing overall network structure. The *Network Community Profile* (Leskovec et al. 2009) has previously been used to conduct analysis of networks that are widely-used in the literature. The NCP, which shows the conductance of the (approximately) best community for each size k, was used to show the prevalence of the "core-periphery" network structure. Such networks have a dense core along with a large number of peripheral, highly sparse "whiskers". However, the NCP is constructed by running like Metis+MQI to construct the best community at each scale. While this is tractable when one has complete data due to the efficiency of such algorithms, it is inapplicable to settings with incomplete data. As we are interested in measuring connectivity on modern social networks, we cannot apply this method.

Another classic metric is the *transitivity* or clustering coefficient of a network (Watts and Strogatz 1998). Suppose the edges $(a, b), (b, c)$ exist. The nodes a, b, c form a triangle, which is called *closed* the third edge (a, c) exists. Transitivity is the fraction of triangles which are closed, i.e. it measures the property of *triadic closure* (Rapoport 1953). However, calculation in a global context requires counting triangles—a challenging task, which leads to the use of heuristics in practice (Berry et al. 2014). The local clustering coefficient, which only counts the triangles incident to an individual node, somewhat addresses this—though with scaling issues of its own. In particular: to calculate the LCC of a node v, one must look up the neighborhoods of each neighbor of v—a task that would frequently require tens of thousands of API calls per node to compute on social media like Twitter.

The *modularity* metric seeks to quantify the degree to which a network may be broken down into well-defined groups (Newman and Girvan 2004). This metric is quite popular despite its issues (Chakraborty et al. 2017), in part because it provides a tractable optimization objective for scalable community detection (Blondel et al. 2008). However, modularity is inherently a global metric. In settings with incomplete access to network data—like our target—modularity cannot be applied.

Edge weighting methods like ours have also seen use. Perhaps the most well-known is *edge betweenness centrality* (Girvan and Newman 2002), which measures the proportion of shortest paths that cross an edge. This variation on node betweenness (Freeman 1977) was used to identify sparse edges for hierarchical community detection. However, exact calculation is expensive (at least $O(nm)$ (Brandes 2001)) and not all sparse connections are highly-weighted on the first pass—requiring an iterative approach that exacerbates this cost. As a result, the research community rapidly moved on. The study of approximate betweenness has seen a revival in recent years (Riondato and Kornaropoulos 2016; Yoshida 2014). However, despite much superior complexity (Riondato and Kornaropoulos is $O(r(n + m))$, $r \ll m$) these methods remain impractical for very large networks.

One variant that is particularly relevant to our study is *random walk betweenness* (Newman 2005). While on the surface this would appear to be closely related to our work, there are substantial differences. Random walk betweenness is defined in terms of the (global) transition matrix of the network, and as such exact computation again has complexity at least $O(nm)$. Recent adaptations with improved complexity exist (Kourtellis et al. 2013), though they do not compete with the work of Riondato and Kornaropoulos in practice.

Moreover, this approach simply uses random walks as randomly selected paths in place of the shortest paths used in traditional betweenness. In contrast, we only use random walks to check for connectivity—a substantial change that makes calculation much, much more efficient in practice.

Random walks have seen application on problems ranging from network embedding (Grover and Leskovec 2016) to community detection (Rosvall and Bergstrom 2008) (see Masuda, Porter, and Lambiotte 2017 for a survey on the subject). We would like to note that not all local methods are equal. Spielman and Teng presents the *Nibble* algorithm for local community detection, but this method cannot be run without knowing the total number of edges m on the network (which determines the stopping condition). It is important to distinguish between methods such as this that require *global meta-knowledge* (like edge count) on top of local topology from those that only require local topology.

We do note that prior work on community detection via random walks—including both (Spielman and Teng 2013) and (Rosvall and Bergstrom 2008)—makes use of the same property we exploit: random walks tend to remain within dense regions rather than cross sparse cuts between regions. However, much of this work depends on global meta-knowledge in a similar manner to Nibble.

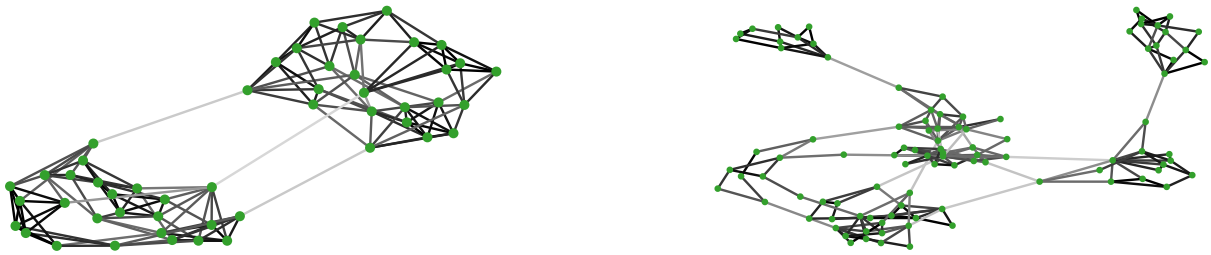
Edgecut Weights

Before proceeding with the construction of our method, we will first review the criteria that inform it. Our overarching goal is to study the structure of networks to which we have only limited access. To accomplish this, we will take *local* measurements—which are feasible via social media APIs—in order to build up an understanding of global structures.

Our approach is based on the idea of *edge sparsity*. Consider an edge $e = (u, v)$. This pair of nodes is sparsely connected if there are few distinct paths to reach u from v (and visa-versa). We can get a local image of this property by using random walks to check for connections in the surrounding area. Put simply: our method simultaneous walks from each endpoint of e until one of two things occurs: (a) a walk reaches a node previously reached by the other (i.e. they *intersect* at a node), or (b) both walks stop. We use random stopping, where each walk flips a weighted coin with parameter $0 < \rho < 1$ and stops if it lands on tails, to control the length of the walks. The edge (u, v) itself is both a trivial path and by far the most likely to be found by such walks, so we remove it to measure the surrounding network. Formally:

Definition 1 (Edgecut Walk)**.** *Let $G = (V, E)$ be a graph and $e \in E$ an edge on it. A random walk is an Edgecut Walk and said to be e-adapted if it walks on $G' = (V, E \setminus \{e\})$.*

Definition 2 (Edgecut Weight)**.** *Let $e = (u, v)$ be an edge on G. The edgecut weight of e, denoted γ_e, is the probability*



(a) Multiply-Connected Pair of Small-World Graphs

(b) LFR Benchmark Graph

Figure 1: Small sample networks, with edges colored according to Edgecut Weights. Dark color indicates a *low* weight. Graphs are (a) a pair of small-world networks with $n = 25$ (Watts and Strogatz 1998), (b) an LFR community benchmark network (Lancichinetti, Fortunato, and Radicchi 2008) with $n = 100$ and a power-law exponent of $\tau_1 = 3$.

that a pair of e-adapted walks rooted at u and v that stop after each step with probability $1 > \rho > 0$ do not intersect (i.e. there is no node reached by both walks).

This definition gives a weight where γ_e should intuitively be high when e is sparse, but low when it is not. The awkwardness of the definition ("do not intersect") is in service of the algorithm we use for calculation, which is presented in the next subsection.

To illustrate our idea, let us consider *communities*. Communities are defined both by the density of connections within and the sparsity of connections between them. Prior work has exploited the property of random walks to remain within dense regions to perform community detection (Spielman and Teng 2013; Rosvall and Bergstrom 2008; Viswanath et al. 2010). Fig. 1 shows sample Edgecut Weights on a pair of small, sythetic networks. While there is certainly variability among weights within communities, there is also a visible difference between intra- and inter-community edges.

Efficient Approximation of Edgecut Weights

Analytic calculation of even a single edgecut weight γ_e is infeasible outside of the simplest cases. However, these random walks are trivial to construct and with the application of existing results on Monte-Carlo sampling, we can construct an approximation extremely efficiently.

Observe that if we sample a pair of e-adapted walks, there are two possible events: they intersect ($Z = 0$) or they do not ($Z = 1$). The probability $\Pr[Z = 1]$ is equal to γ_e by definition. This is a Bernoulli random variable, and thus $\Pr[Z = 1] = \mathbb{E}[Z] = \gamma_e$. We can therefore use the *sample mean* $\tilde{\gamma_e} = \sum_{i=0}^{n} Z_i$ of a sequence of samples Z_i to estimate the true mean γ_e. Our samples in this case are $\{0, 1\}$ values representing the intersection (or lack thereof) of walks constructed according to Def. 2.

The EBGStop algorithm (Alg. 1; Mnih, Szepesvári, and Audibert 2008) can be used to estimate γ_e with a near-optimal number of samples.[1] In the Bernoulli case, their results are:

Theorem 1 (Mnih, Szepesvári, and Audibert 2008)**.** *Let Z be a random variable distributed in $[0, 1]$ with mean $\mu = \mathbb{E}[Z] > 0$ and variance σ^2. Let $\tilde{\mu}$ be the estimate produced by EBGStop, let T be the number of samples used to construct it, and let C be a constant. Then if $\epsilon, \delta \in (0, 1)$ are user-defined error-bound parameters, we have:*

1. $\Pr[\mu(1-\epsilon) \leq \tilde{\mu} \leq \mu(1+\epsilon)] > 1 - \delta$,
2. $\Pr\left[T \geq C \max\left\{\frac{\sigma^2}{\epsilon^2\mu^2}, \frac{1}{\epsilon\mu}\right\}\left(\log\log\frac{1}{\epsilon\mu} - \log\delta\right)\right] < \delta$

In other words, the EBGStop algorithm will produce an estimate $\tilde{\gamma_e}$ that is within a factor of $1 \pm \epsilon$ of the true value γ_e with probability at least $1-\delta$, where both ϵ and δ are user-specified parameters. Additionally, it guarantees the number of samples used is at most the complexity given in point (2) with probability at least $1 - \delta$.

Note that when variance is low, the complexity is linear in ϵ^{-1} and μ^{-1} as well as logarithmic in δ. That is: it depends only indirectly on the properties of the local network, and grows very slowly with changes in μ and only a bit faster with σ. Also note that an expected δ fraction of edges will have weights exceeding their error bounds, but due to the logarithmic scaling it is trivial to set δ to a very small value. We use $\delta = 0.01$ in our experiments. Additionally, we remark that this theorem makes use of the *true* mean and standard deviation, while the algorithm makes use of the *sample* mean $\bar{X}_t$ and variance $\overline{\sigma_t}$ at each step t.

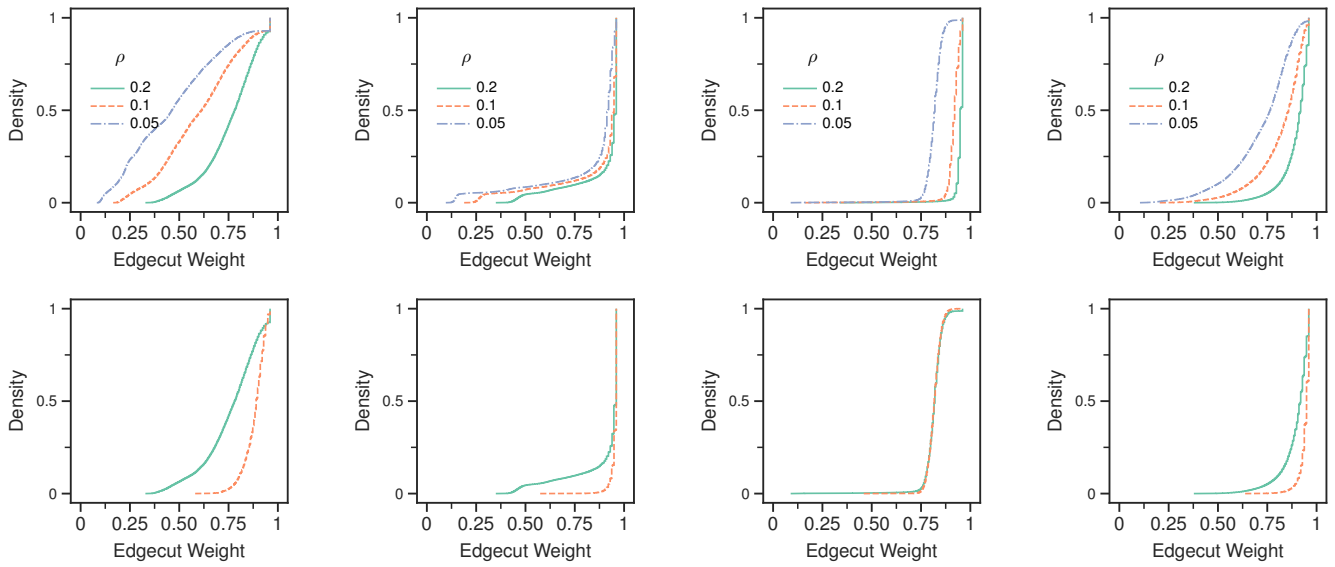
To make use of this algorithm, we need to show that our random variable has a non-zero expected value.

Lemma 1. *Let Z be 1 if a pair of e-adapted walks with stopping probability ρ rooted at the endpoints u, v of e **do not** intersect and 0 otherwise. Then $\mathbb{E}[Z] > 0$ unless $u = v$.*

Proof. Due to the constraint that $\rho > 0$, there is a non-zero probability that each walk stops immediately. Thus, there will always be some portion of walks that do not intersect. As a result, the expected value is non-zero. □

Thus, the EBGStop algorithm is applicable to estimation of Edgecut Weights.

[1] Other algorithms for this problem exist (Dagum et al. 2000) including specializations for Bernoulli variables (Huber 2017); EBGStop performed best in our tests.



(a) GrQc ($Q \approx 0.86$) (b) Slashdot ($Q \approx 0.41$) (c) #GamerGate ($Q \approx 0.27$) (d) Facebook Pages ($Q \approx 0.80$)

Figure 2: Top: Distribution of Edgecut Weights as a function of ρ. Bottom: Distribution among edges that are cross communities (orange, dashed) identified by the Louvain algorithm, along with overall distribution (solid). Each edge appears in the Louvain distribution once for each time cut across 100 runs. The average modularity Q across all runs is shown.

Algorithm 1 EBGStop for $Z \in \{0, 1\}$ (Mnih et. al. 2008)

$LB \leftarrow 0, UB \leftarrow \infty, t \leftarrow 1, k \leftarrow 0, \beta \leftarrow 1.1, p \leftarrow 1.1$
Sample Z_1
while $(1+\epsilon)LB < (1-\epsilon)UB$ **do**
  $t \leftarrow t + 1$
  Sample Z_t
  if $t >$ floor(β^k) **then**
    $k \leftarrow k + 1$
    $\alpha \leftarrow$ floor(β^k)/floor(β^{k-1})
    $d_k \leftarrow \delta(1 - 1/p)/(\log_\beta k)^p$
    $x \leftarrow -\alpha \log d_k/3$
  $c_t \leftarrow \overline{\sigma_t}\sqrt{2x/t} + 3x/t$
  $LB \leftarrow \max(LB, |\bar{X}_t| - c_t)$
  $UB \leftarrow \min(UB, |\bar{X}_t| + c_t)$
return $1/2 \cdot [(1+\epsilon)LB + (1-\epsilon)UB]$

The Behavior of Edgecut Weights

We next empirically examine the behavior of Edgecut Weights along two axes: efficiency and correctness. Efficiency is evaluated relative to (approximate) calculation of Edge Betweenness Centrality. Correctness is—unfortunately—harder to directly evaluate. We take a multifaceted approach to doing so.

First, we compare to communities produced by the Louvain method (Blondel et al. 2008). If the Edgecut Weights are behaving as intended, the edges that cross between the detected communities should have *high* weights. We qualitatively find this to be the case on many networks.

Second, we compare to *Edge Betweenness Centrality* (Girvan and Newman 2002), which is one of the few edge-weighting methods to have seen prior use for partitioning problems. While betweenness is an imperfect match, it has the similar property that cut edges *should* have high weight (this idea underpins the classical Girvan-Newman community detection method). Thus, we should see some level of correlation between the two quantities.

Table 1: Datasets used in our comparison. All data is taken from SNAP (Leskovec and Krevl 2014) unless otherwise noted.[2] All networks are treated as undirected.

Name	Kind	n	m
GrQc	Collab.	5,242	14,496
HepPh	Collab.	12,008	118,521
NetHept	Collab.	15,229	31,376
DBLP	Collab.	317,080	1,049,866
#GamerGate	Interaction	13,188	182,176
Enron Email	Interaction	36,692	183,831
EU Email	Interaction	34,845	99,074
Wiki Talk	Interaction	2,394,385	4,659,565
Facebook Pages	Social	22,470	171,002
Slashdot (2009)	Social	82,168	582,533
Orkut	Social	3,072,434	29,424,825
Twitter (2010)	Social	41.7M	1.47B

[2]Twitter is from (Kwak et al. 2010). #GamerGate is the sparse Twitter interaction network of #GamerGate users from (Smith and Thai 2019). NetHept is from (Chen, Wang, and Yang 2009). EU Email includes only nodes observed as both sender and recipient.

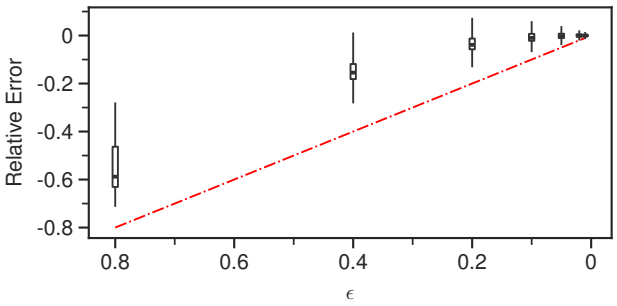


Figure 3: Sensitivity to choice of ϵ. Relative error is estimated on GrQc against a randomly selected run with $\epsilon = 0.01$. Data for 10 runs is shown, with whiskers extending to the 99th percentile. The red dashed line shows the error margin guaranteed with probability $1 - \delta = 0.99$.

The Weight of Cut Edges

We will begin with a comparison to the Louvain method. This algorithm is a heuristic community detection method that operates by maximizing modularity. Despite being a heuristic, it is widely used and well-regarded.

Our goal is to determine whether the edges that cross between communities (or *cut* edges) by this method have high Edgecut Weight. Intuitively, this should be the case if the quantity performs as desired. This is somewhat complicated by the fact that Louvain is order-dependent: if the same topology is input in a different order, the solution changes (despite being otherwise deterministic).

We deal with this by *shuffling* the input networks and comparing both (1) which edges are cut, and (2) how frequently different edges are cut. Unless specified otherwise, we estimate the Edgecut Weight with $\rho = 0.2$, $\epsilon = 0.2$ and $\delta = 0.01$. Fig. 2 compares the overall distribution of weights to the distribution of weights among edges that are cut among 100 shuffled solutions.

From Fig. 2a, 2b and 2d we can clearly see that this hypothesis holds. However, Fig. 2c (which uses $\rho = 0.05$ for visibility) shows an apparently contradictory case. Upon inspection the Louvain method cuts well over half of all edges on this network each time, resulting in a very low modularity of 0.27. Combined with repeated runs and inconsistent cuts, the outcome is that nearly all edges (save those in the left tail) are cut by the Louvain method on this network. This network is built from a 1% sample of Twitter interactions, and is dominated by a singular community (Smith and Thai 2019). Thus, the inability of the Louvain method to further subdivide it is not altogether surprising.

Interestingly, decreasing the stopping probability ρ appears to have minimal impact on Slashdot compared to the other networks (c.f. Fig. 2b). While the other networks see fairly dramatic shifts towards lower weights (i.e. more intersections), we continue to see a heavily skewed distribution on Slashdot. This does not appear to be a symptom of our choice of error margin ϵ. Fig. 3 shows the actual relative error on GrQc. Note that it decreases much more rapidly than ϵ. This is not altogether surprising: approximation guarantees are often conservative.

Correlation with Edge Betweenness

Next, we examine the relationship between Edgecut Weights and Edge Betweenness Centrality. Edge Betweenness weights are notable for their use in community detection (Girvan and Newman 2002), though despite numerous extensions a scalable method of calculating them has yet to be found. Within this section, we additionally compare to the state of the art approximation method for Edge Betweenness (Riondato and Kornaropoulos 2016).

We use our own implementation of each method, in no small part to take advantage of parallelism. The methods of (Brandes 2001) and (Riondato and Kornaropoulos 2016) are parallelized by running the main loop on t threads, and accumulating result vectors. We validated the weights constructed by our parallel implementation against those of the `igraph` package (Csardi and Nepusz 2006). Our method is trivially parallelized across t threads on a per-edge basis. The source code for each method is available online.[3] Unless specified otherwise, we fix $\delta = 0.01, \rho = 0.2, \epsilon = 0.2$. For the method of Riondato and Kornaropoulos, we use the relative error bound given in their work with $\epsilon = 0.2$ that reduces to an additive bound on weights below $q = 0.01$.

We calculate correlation with Kendall's rank-order coefficient τ (Zwillinger and Kososka 2000) as implemented in the SciPy package (Jones et al. 2001). τ ranges from -1 to $+1$, with 0 indicating no correlation, $+1$ indicating that elements are ranked in identical orders by both metrics, and -1 indicating exactly inverted orderings. τ also has the property that the fraction of correctly ordered pairs is equal to $\tau + (1 - \tau)/2$. As a result, when $\tau = 0.5$ exactly 75% of all pairs agree in both orderings.

Table 2 illustrates two key results. First and foremost: on many networks, Edgecut Weights display a correlation that is competitive with the state-of-the-art approximation—even outperforming it on two networks (Slashdot, DBLP). However, this is coupled with notably poor correlation on the #GamerGate and Facebook Pages datasets. Given the degree of centralization among weights on the #GamerGate network—especially with $\rho = 0.2$, as used in this table—the low correlation is not particularly surprising. Betweenness is a ranking metric, and so will *always* produce a spread distribution. When Edgecut Weights do not, we expect the correlation to be low. The performance gap between the methods is also quite clear, especially on a large network like DBLP.

Not shown in the above are the results on the Twitter data. We ran our method and attempted to run Riondato and Kornaropoulos on this data on a large server[4] with 70 threads. While our method completed in 5h 9m, R&K did not complete in a reasonable timeframe and was stopped at 24 hours.

Interestingly, we further find that the correlation between Edgecut Weights and Edge Betweenness is strongest among *low-weighted edges* (c.f. Table 3). This contrasts with prior methods for approximating betweenness, which focused on

[3] `https://gitlab.com/emallson/edgecuts`

[4] This network is larger than main memory on the machine used to run the other tests.

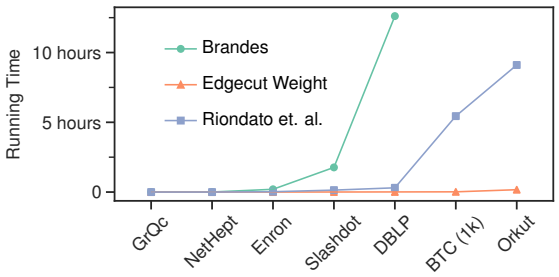


Figure 4: Running time of Edgecut Weights and (Riondato and Kornaropoulos 2016) on each network in Table 1. Brandes' method is not run on BTC or Orkut.

Table 2: Correlation (τ) with & Speedup over Edge Betweenness Centrality. Values are listed as mean $\pm$ standard deviation over 10 runs. Speedup is measured by comparing the wall-clock times when run with 8 threads.

	Dataset	τ		Speedup	
EW	GrQc	0.47	$\pm$0.00	164.00	$\pm$9.92
	Slashdot	0.38	$\pm$0.00	501.50	$\pm$37.72
	#GG	0.02	$\pm$0.00	94.17	$\pm$18.67
	FB Pgs.	0.17	$\pm$0.00	119.56	$\pm$25.14
	DBLP	0.48	$\pm$0.00	1,650.29	$\pm$161.48
R&K	GrQc	*Not Run*			
	Slashdot	0.36	$\pm$0.01	13.61	$\pm$1.29
	#GG	0.42	0.00	3.69	0.05
	FB Pgs.	0.57	0.00	4.42	0.08
	DBLP	0.38	$\pm$0.01	41.02	$\pm$2.88

the highly-weighted edges used by the Girvan–Newman method. On the whole, these results indicate that while both metrics measure similar things, they are not interchangeable.

Conductance, Centrality, and Edgecut Weights

As indicated above, there is a fundamental connection shared by both Edgecut Weights and Edge Betweenness: they identify sparse cuts. In this section we explore the theoretical relation between the two by way of *conductance*. In particular, we find that (1) betweenness has close relationship with conductance *on highly-weighted edges*, while (2) Edgecut Weights are related solely to local properties. To

Table 3: Overlap of the top and bottom 10% of each ordering with the Edge Betweenness.

Dataset		Jaccard Index: This Work	Jaccard Index: Riondato et al.
Slashdot	Top	0.06	0.25
	Bottom	0.49	0.05
DBLP	Top	0.16	0.33
	Bottom	0.46	0.05

begin, let us review the definition of Edge Betweenness.

Definition 3 (Edge Betweenness Centrality)**.** *Define σ_{st} as the number of shortest paths from vertex s to vertex t on a graph $G = (V, E)$. Likewise define $\sigma_{st}(e)$ to be the number of those paths containing edge $e \in E$. Then the edge betweenness centrality of e is*

$$c(e) = \sum_{s \neq t \in V} \frac{\sigma_{st}(e)}{\sigma_{st}}$$

We can directly show that the edges in a sparse cut will typically have higher betweenness centrality than edges in more densely-connected regions. Let us begin with a simple (but useful) lemma:

Lemma 2. *Define the* total betweenness *of a set $S \subseteq E$ as*

$$T(S) = \sum_{s \neq t \in V} \sigma_{st}^{-1} \sum_{p \in P_{st}} |p \cap S|$$

where P_{st} is the set of shortest paths from s to t. Then

$$T(S) = \sum_{e \in S} c(e)$$

Proof.

$$\begin{aligned} T(S) &= \sum_{s \neq t \in V} \sigma_{st}^{-1} \sum_{p \in P_{st}} |p \cap S| \\ &= \sum_{s \neq t \in V} \sigma_{st}^{-1} \sum_{p \in P_{st}} \sum_{e \in p \cap S} 1 \\ &= \sum_{s \neq t \in V} \sigma_{st}^{-1} \sum_{e \in S} \sigma_{st}(e) = \sum_{e \in S} c(e) \end{aligned}$$

where the third equivalence arises from the definition of $\sigma_{st}(e)$ as the number of shortest paths from s to t on which e lies. Enumerating the shortest paths p and counting the number of edges from S that lie thereon is equivalent to enumerating the edges in S and counting the number of shortest paths on which they lie. □

With this, we can prove a stronger result: that high-centrality cuts put a bound on the conductance of the cut, and that balanced cuts have the tightest bound. Before that, we need one more simple lemma.[5]

Lemma 3. *Let $x, y, a, b > 0$. Then*

$$\frac{x+y}{a+b} < \frac{y}{b} \iff \frac{x}{a} < \frac{y}{b}$$

With this, we can prove the main result of this section:

Theorem 2. *Suppose a connected, undirected, unweighted graph $G = (V, E)$ is divided into disjoint connected components $A, B \subset V$ by a cut $C \subset E$, where the partition defined by A has fewer edges than the one defined by B. Define $\bar{c}(S) = T(S)/|S|$ as the mean betweenness of the edges $e \in S \subset E$. Let $E(A)$ be the set of edges with both endpoints in A and $\Phi(A)$ be the conductance of A. Then when*

$$\bar{c}(E(A)) < \bar{c}(C)$$

we have

$$\Phi(A) = \Phi(B) < \frac{T(C)}{2\left[T(E(A)) + T(C)\right]}$$

[5]The proofs of Lemma 3 and Theorem 3 are in the appendix.

Proof. By Lemma 3, we know that

$$\bar{c}(E(A)) < \bar{c}(C) \implies \bar{c}(E(A) \cup C) < \bar{c}(C)$$

In other words, we have:

$$\frac{T(E(A)) + T(C)}{|E(A)| + |C|} < \frac{T(C)}{|C|}$$

where the numerator follows from the fact that $T(X \cup Y) = T(X) + T(Y)$. Multiplying by $\frac{1}{2}$ and rearranging gives

$$\frac{|C|}{2(|E(A)| + |C|)} < \frac{T(C)}{2\,[T(E(A)) + T(C)]} \tag{1}$$

Recall the definition of conductance on undirected, unweighted networks:

$$\Phi(S) = \frac{|\delta S|}{2\min\{|E(S)|, |E(V \setminus S)|\} + 2|\delta S|}$$

where δS is the set of edges with one endpoint in S and one endpoint in S^c.

As a result, we have:

$$\Phi(A) = \Phi(B) = \frac{|C|}{2\min\{|E(A)|, |E(B)|\} + 2|C|}$$

By the statement of the theorem, we know that $\min\{|E(A)|, |E(B)|\} = |E(A)|$. Thus, we can simplify Eqn. (1) to

$$\Phi(A) = \Phi(B) < \frac{T(C)}{2\,[T(E(A)) + T(C)]}$$

□

In effect, this places a condition on the cut C in relation to the smaller of the two resulting partitions (A). When the average betweenness of the cut is higher than that of the partition, we can bound the conductance of the cut in terms of the total betweenness $T(\cdot)$. In particular: the larger the size of the smaller partition, the smaller the conductance bound becomes. Balanced cuts, thus, have the smallest bound.

A converse relation holds by a very similar proof (given in the appendix):

Theorem 3. *Suppose a connected, undirected, unweighted graph $G = (V, E)$ with $n = |V|$ nodes is divided into disjoint connected components $A, B \subset V$ by a cut $C \subset E$ as above. Let $q_A = |A|/n$. Define $\bar{d}$ as the mean distance between node pairs on G and $\bar{c}(S)$ the mean betweenness of edges $e \in S \subset E$. Then when*

$$\frac{q_A - q_A^2}{\bar{d}} \geq \Phi(A)$$

we have that

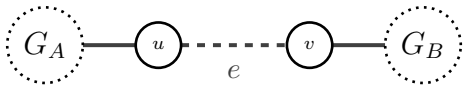
$$\bar{c}(E(A)) < \bar{c}(C)$$

When $|A| \approx |B|$, the condition $1/(4\bar{d}) \geq \Phi(A)$ implies the same result. This is (asymptotically) the lowest bound on $\Phi(A)$ provided by Theorem 3. Note that on many networks the *maximum* distance is quite small and that the average includes exactly m 1s from the m edges. For example, the maximum distance observed on Twitter as of 2010 was 18, but 97.6% of users were within a distance of 6 (Kwak et al. 2010). Thus, it appears that this condition will hold in many cases, linking the two quantities at relatively high levels of conductance.[6]

On the other hand, it is easy enough to show that Edgecut Weights are dominated by local properties.

Theorem 4. *Suppose $C \subset E$ partitions a graph G into two connected components. In general, for an $e \in C$ γ_e is not related to $\Phi(C)$*

Proof. Consider the following construction:



Here, the overall graph G may be partitioned into two parts G_A, G_B[7] by cutting e. Hence, the conductance can be given in closed form as

$$\Phi(A) = \Phi(B) = \frac{1}{2\min\{|E(A)|, |E(B)|\} + 2}$$

That is: the conductance depends solely on the size of the smaller partition. Thus, by altering the number of edges in G_A, G_B we can assign (nearly) arbitrary values to the conductance of the cut on the range $(0, 1/2)$.

However, observe that $\gamma_e = 1$. There is no way for the walks to ever intersect. Thus, there is no relationship between the two values in general. □

While this proof addresses the most degenerate case, it is easy to see the lack of relationship given this result when u, v are remain connected after the cut: one may place the connection between them arbitrarily far away such that $\gamma_e \approx 1$ regardless of the true size of G_A, G_B. Alternately, one may form a clique containing u, v. In this case, γ_e depends principally on the clique, and alterations to G_A, G_B will have negligible impact.

These results help explain the behavior seen in the previous section: high-betweenness edges will tend to be a part of balanced, sparse cuts, while Edgecut Weights instead depend on local structures. Interestingly, Table 3 shows that there is substantial overlap in the bottom end of both metrics, which may indicate that low-betweenness edges have a similar dependence on local structures.

Visualizing Connectivity with Edgecut Weights

In this section, we apply Edgecut Weights to characterize connectivity on a variety of networks, and explore what these reveal about the structure of the data. Figures 5, 7 & 9 show the weight distributions of three kinds of networks:

[6] Note that these are only statements about the *average* centrality. Given prior results (Girvan and Newman 2002; i.e. "there is no guarantee that all of [the edges on a cut] will have high betweenness—we only know that one of them will"), a similar result for individual centrality seems unlikely.

[7] The graphic slightly abuses this notation—u, v should also be members of G_A, G_B.

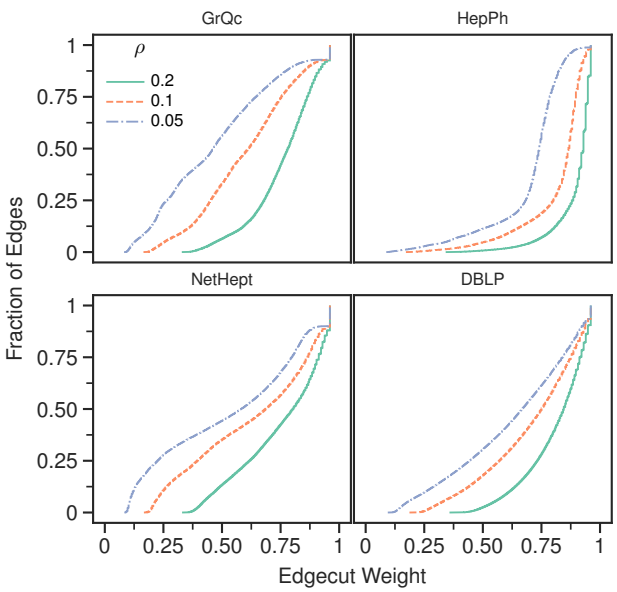


Figure 5: Empirical CDFs of collaboration networks. Edges represent coauthorship.

research collaboration, friendship (or what are typically termed "social networks"), and social interaction. These are drawn as *cumulative distribution plots*, as they are a compact and rich means of representing an empirical distribution.

We will begin by exploring the features of these visualizations that highlight topological structures. Then, we will briefly discuss our method to construct a distribution estimate on the Twitter network ca. December 2019. Finally, we will delve into the similarities and differences between different kinds of networks as shown in these figures.

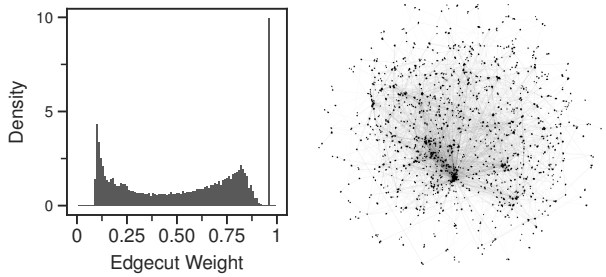
Properties of the Weight Distributions

Skewness. The first and most basic property to consider is how skewed the distribution is. Contrast GrQc and DBLP (Fig. 5), which ascend smoothly from low weights to high weights, to Orkut and Twitter (Fig. 7). The former are examples of spread distributions with minor rightward skew; the latter exhibit extreme skew, with only a minimal tail.

Intuitively, such a skewed distribution indicates a lack of dense regions in the network, while a spread distribution shows the presence of regions at multiple density levels. This is supported by the transitivity[8] of these networks: where GrQc (0.36) and DBLP (0.12) have relatively high transitivity, Orkut (0.02) and Twitter (2010; 8.84×10^{-4}) do not.

Shape. The potency of distributional plots is perhaps most clearly evidenced by the overall shape. We have already commented on the extremities—nearly-level spread and extreme skew—but greater details are contained. Consider NetHept (Fig. 5): this network has two steep ascensions

[8]The fraction of triangles (u, v), (v, w) for which the third edge (u, w) exists. Such triangles are said to be "closed." We use `igraph` (Csardi and Nepusz 2006) to calculate this.



(a) Edgecut Weight Dist. (b) Topology

Figure 6: The distribution of edgecut weights on NetHept along with the network topology. Fig. (b) drawn with Gephi (Bastian, Heymann, and Jacomy 2009); layout computed with OpenOrd (Martin et al. 2011).

shown with $\rho = 0.05$. The first occurs around a weight of $\gamma = 0.1$, and the latter around $\gamma = 0.8$. This bimodal distribution indicates that it is closer to an idealized network with communities: densely connected regions (in the low-weight hump) coupled with sparse connections between.

For clarity, we plot a histogram of NetHept (Fig. 6a). The spike at low weights is easily visible, as are spikes at around 0.8 and 1. The topology of the network (Fig. 6b) gives some insight into why this occurs: we can see a dense region around the center of the drawing that likely corresponds to the low-weight edges. Similarly, we see a relatively smooth transition from the central region out towards the highly sparse outer edges of the network. This is reflected in the relatively smooth transition of weights shown in Fig. 6a.

Meanwhile, other networks exhibit a steep "elbow" shape. Take Enron Email as an example (Fig. 9). In contrast to the extremely skewed networks discussed previously, this network has a heavy tail of low-weight edges. In this case, it appears that much of the network is fairly sparse with one or more smaller dense regions. Slashdot (Fig. 7) and HepPh (Fig 5) both display similar structures, with varying amounts of the distribution in the low-weight tail.

We remark that the similar structure between the Slashdot and HepPh networks is *not* indicated by other metrics: the two differ in transitivity by over an order of magnitude (8.17×10^{-3} and 0.39, respectively). The same holds for the average local clustering coefficient (0.06 and 0.61). Despite this, the two have remarkable similarities. Both networks (see Fig. 8) have clearly visible cores—shown as the low-weight part of the elbows—with varying levels of density as they progressively transition to "peripheral" regions. We can additionally see the relative steepness of this transition represented in both plots: HepPh has a smoother transition from the core, while Slashdot has a stark gap between regions along with multiple distinct cores. This kind of core-periphery structure is widespread in large networks (Leskovec et al. 2009), so the commonness of this elbow shape is unsurprising.

ρ-Sensitivity. Recall that the fundamental idea underlying this measurement process is that we are *cutting* an edge

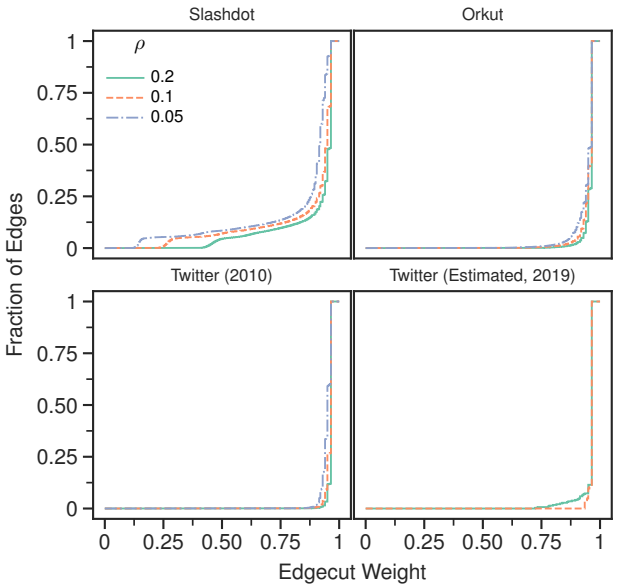


Figure 7: Empirical CDFs of social (or: friend/follower) networks. Edges represent status as a "friend" or "follower".

(a) HepPh (b) Slashdot

Figure 8: Two networks that display "elbow-shaped" weight distributions (see Fig. 5 & 7).

(u, v). The parameter ρ then controls the range within which we search for alternate routes from one endpoint to the other. By varying ρ, we can gain insight into the differences between short- and long-range connections on a network.

For example: the Orkut and Slashdot networks see minimal change in distribution as ρ decreases (with the average walk length increasing from 5 to 20; 20 is more than double the average distance between node pairs on every network we study). This indicates that the intersections observed with large ρ are the bulk of intersections observed with smaller ρ—in other words: when alternate routes are found, they are found nearby. That Orkut and Twitter see little change as ρ increases *and* are so heavily skewed indicates that the networks are only sparsely connected both locally and globally.

In contrast, we see a substantial increase in the number of intersections on networks like GrQc, NetHept, and #GamerGate. This shows that these networks have redundant routes

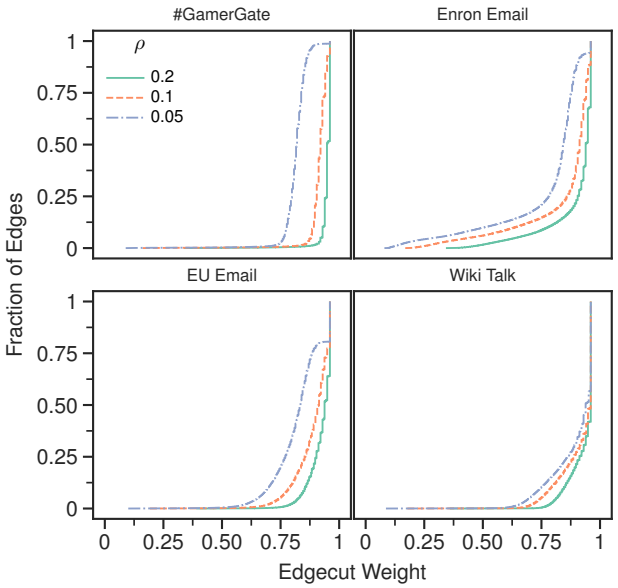


Figure 9: Empirical CDFs of interaction networks. Edges represents a (multiplicity of) interactions on social media.

from u to v, but that many of these routes are longer—meaning walks that are longer on average are more likely to find them. An interesting aspect of this is the known presence of "hub" or "central" users on the #GamerGate network (Smith and Thai 2019). A critical mass of walks reaching these hub users may be responsible for the abrupt decrease in weights as ρ decreases.

Weight Distribution Estimation via the Twitter API

As highlighted previously, one of the strengths of our approach is the dependence on *purely local* information. In particular: the only requirement to calculate an Edgecut Weight is that you can obtain a list of neighbors for a node. The Twitter API gives access to this information, allowing us to **calculate Edgecut Weights directly on Twitter.** This, in turn, allows us to produce the plot shown in Fig. 7.

In order to estimate a distribution, one needs independent samples. We accomplish this by taking uniform samples of edges as follows: we first uniformly sample a user on the network, then take an adjacent edge at random. The weight of this edge is then calculated and recorded. As Twitter moved all identifiers to the non-contiguous Snowflake ID system in 2010 (Kergl, Roedler, and Seeber 2014), simple rejection sampling is impractical to construct our uniform user sample. Instead, we used a MCMC-style approach (detailed in the appendix). In total, we calculate the weights of 288 edges with $\rho = 0.2$ and an additional 124 edges with $\rho = 0.1$.

Our results clearly show that the connectivity of Twitter has not changed much in the intervening decade, as the skewed distribution remains with both values of ρ. We note that the $\rho = 0.2$ distribution is *less* skewed than its $\rho = 0.1$ counterpart in this case, though this is almost certainly a product of variance in sampling. However, even in this case

we do not observe any weights below 0.70 and again nearly 90% of weights are indistinguishable from 1.

Connectivity as an Image of Social Processes

In the course of this study, we observe a number of similarities in the distributions within network types. Perhaps the clearest example of this is the contrast between collaboration networks, which tend to be spread with many low-weight edges, and friendship (or: social) networks, which invariably have a steep elbow shape and tend to be insensitive to ρ. Though each has exceptions (HepPh displays an elbow shape, while Facebook Pages (Fig. 2) lacks the skewness of other friendship networks), the similarities remain striking.

Social networks are, fundamentally, an image of some underlying social process. This is by definition, and in the authors' view a fundamental aspect of why they are interesting. These figures indicate that the *connectivity* in particular can us a great deal of insight into the structure of the social processes which produces this network data.

Take, for example, the skewness of the friendship networks. This skewness is a direct product of an absence of triangles—a lack which contradicts the widespread idea that triadic closure (Rapoport 1953) applies to "friend" links on modern social media (e.g. Boshmaf et al. 2011; Golder and Yardi 2010). Triangles are further closely connected to (topological) communities (Prat-Pérez, Dominguez-Sal, and Larriba-Pey 2014). To be almost entirely absent indicates a distinct *lack* of communities in these social networks—including Twitter. Though other tools allow one to reach such conclusions on complete data, our approach is the first to extend them to the current Twitter network.

In contrast, the interaction networks seen in Fig. 9 display much greater density. Interestingly, only the Enron Email network displays a clear elbow shape, while Wiki Talk appears to follow a similar pattern with a relatively sparse core. Both #GamerGate and EU Email display similarly abrupt decreases in weight distribution as ρ decreases. We believe this is due to "hub" users (whose presence is known *a priori* for #GamerGate) introducing a large number of intersections. On the whole, the substantially higher density shown on these networks is in agreement with prior work that has found evidence of communities on interaction networks (DeMasi, Mason, and Ma 2016; Smith and Thai 2019).

Discussion

In this work, we have presented a novel means to measure and visualize connectivity on large (social) networks, and subsequently applied this to study the structure of multiple networks—including new results on the relationship between Twitter in 2010 and 2019. Our theoretical results further illuminate the connection between Edge Betweenness and globally balanced sparse cuts—and how the local focus of Edgecut Weights leads to differing behavior.

This differing behavior is a side-effect of our focus on measuring local quantities. While this focus is clearly beneficial given the constraints of studying modern social networks, we are curious the extent to which the locality of measurements can be relaxed without sacrificing the purely-local computation. The distributions we use to qualitatively analyze these networks provide insight into global structure based on purely-local measurements, but we believe that future work may be able to make much stronger statements through more complex structured approaches to sampling.

Further, we are interested in further exploration of the differences between interaction and friendship networks. Our results here indicate that they have substantial structural differences, with a striking lack of dense regions that would indicate communities on the large friendship networks—and thus that researchers likely should *not* seek information on communities from the topology of these networks. We would particularly be interested in comparing the prevalence of triadic closure between interaction and friendship networks, as this property has long had ties to theories of friendship formation and community growth (Rapoport 1953) and continues to see application in computational-social work.

On the whole, our work gives a novel and efficient way of measuring connectivity on large networks that has applications both in qualitative analysis (as we have done here) and may have use in the development of efficient algorithms for network optimization (as Edge Betweenness has been used in prior work). Additionally, our results advance the state of knowledge of online social systems, and have potential to do so again in further studies.

Acknowledgements

This work was supported in part by the National Science Foundation under award number CNS-1814614.

References

Bastian, M.; Heymann, S.; and Jacomy, M. 2009. Gephi: an open source software for exploring and manipulating networks. In *Third international AAAI conference on weblogs and social media*.

Berry, J. W.; Fostvedt, L. K.; Nordman, D. J.; Phillips, C. A.; Seshadhri, C.; and Wilson, A. G. 2014. Why Do Simple Algorithms for Triangle Enumeration Work in the Real World? In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS '14, 225–234. New York, NY, USA: ACM.

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008(10):P10008.

Boshmaf, Y.; Muslukhov, I.; Beznosov, K.; and Ripeanu, M. 2011. The Socialbot Network: When Bots Socialize for Fame and Money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, 93–102. ACM.

Brandes, U. 2001. A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology* 25(2):163–177.

Brooks, S. P., and Gelman, A. 1998. General methods for monitoring convergence of iterative simulations. 7(4):434–455.

Chakraborty, T.; Dalmia, A.; Mukherjee, A.; and Ganguly, N. 2017. Metrics for Community Analysis: A Survey. 50(4):54:1–54:37.

Chen, W.; Wang, Y.; and Yang, S. 2009. Efficient Influence Maximization in Social Networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, 199–208. New York, NY, USA: ACM.

Csardi, G., and Nepusz, T. 2006. The igraph software package for complex network research. *InterJournal* Complex Systems:1695.

Dagum, P.; Karp, R.; Luby, M.; and Ross, S. 2000. An Optimal Algorithm for Monte Carlo Estimation. *SIAM Journal on Computing* 29(5):1484–1496.

DeMasi, O.; Mason, D.; and Ma, J. 2016. Understanding communities via hashtag engagement: A clustering based approach. In *International AAAI Conference on Web and Social Media*, ICWSM.

Freeman, L. C. 1977. A set of measures of centrality based on betweenness. *Sociometry* 35–41.

Girvan, M., and Newman, M. E. J. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99(12):7821–7826.

Gjoka, M.; Kurant, M.; Butts, C. T.; and Markopoulou, A. 2010. Walking in Facebook: A Case Study of Unbiased Sampling of OSNs. In *2010 Proceedings IEEE INFOCOM*, 1–9.

Golder, S. A., and Yardi, S. 2010. Structural Predictors of Tie Formation in Twitter: Transitivity and Mutuality. In *2010 IEEE Second International Conference on Social Computing (SocialCom)*, 88–95.

Grover, A., and Leskovec, J. 2016. Node2Vec: Scalable Feature Learning for Networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 855–864. New York, NY, USA: ACM.

Huber, M. 2017. A Bernoulli mean estimate with known relative error distribution. *Random Structures & Algorithms* 50(2):173–182.

Jones, E.; Oliphant, T.; Peterson, P.; et al. 2001. SciPy: Open source scientific tools for Python. [Online; accessed August 2019].

Kergl, D.; Roedler, R.; and Seeber, S. 2014. On the endogenesis of Twitter's Spritzer and Gardenhose sample streams. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, 357–364.

Kourtellis, N.; Alahakoon, T.; Simha, R.; Iamnitchi, A.; and Tripathi, R. 2013. Identifying high betweenness centrality nodes in large social networks. *Social Network Analysis and Mining* 3(4):899–914.

Kwak, H.; Lee, C.; Park, H.; and Moon, S. 2010. What is Twitter, a Social Network or a News Media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, 591–600. New York, NY, USA: ACM.

Lancichinetti, A.; Fortunato, S.; and Radicchi, F. 2008. Benchmark graphs for testing community detection algorithms. 78(4):046110.

Leskovec, J., and Krevl, A. 2014. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Leskovec, J.; Lang, K. J.; Dasgupta, A.; and Mahoney, M. W. 2009. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. *Internet Mathematics* 6(1):29–123.

Martin, S.; Brown, W. M.; Klavans, R.; and Boyack, K. W. 2011. Openord: an open-source toolbox for large graph layout. In *Visualization and Data Analysis 2011*, volume 7868, 786806. International Society for Optics and Photonics.

Masuda, N.; Porter, M. A.; and Lambiotte, R. 2017. Random walks and diffusion on networks. 716:1–58.

Mnih, V.; Szepesvári, C.; and Audibert, J.-Y. 2008. Empirical Bernstein Stopping. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, 672–679. New York, NY, USA: ACM.

Newman, M. E. J., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical Review E* 69(2):026113.

Newman, M. E. 2005. A measure of betweenness centrality based on random walks. *Social networks* 27(1):39–54.

Prat-Pérez, A.; Dominguez-Sal, D.; and Larriba-Pey, J.-L. 2014. High Quality, Scalable and Parallel Community Detection for Large Real Graphs. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, 225–236. ACM.

Rapoport, A. 1953. Spread of information through a population with socio-structural bias: I. Assumption of transitivity. 15(4):523–533.

Riondato, M., and Kornaropoulos, E. M. 2016. Fast approximation of betweenness centrality through sampling. *Data Mining and Knowledge Discovery* 30(2):438–475.

Rosvall, M., and Bergstrom, C. T. 2008. Maps of Random Walks on Complex Networks Reveal Community Structure. 105(4):1118–1123.

Sinclair, A. J. 1988. *Randomised algorithms for counting and generating combinatorial structures*. Ph.D. Dissertation, University of Edinburgh.

Smith, J. D., and Thai, M. T. 2019. Supporting a Storm: The Impact of Community on #GamerGate's Lifespan. *IEEE Transactions on Network Science and Engineering*.

Spielman, D. A., and Teng, S.-H. 2013. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. 42(1):1–26.

Viswanath, B.; Post, A.; Gummadi, K. P.; and Mislove, A. 2010. An Analysis of Social Network-based Sybil Defenses. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, 363–374. New York, NY, USA: ACM.

Watts, D. J., and Strogatz, S. H. 1998. Collective dynamics of 'small-world' networks. *Nature* 393(6684):440–442.

Yoshida, Y. 2014. Almost linear-time algorithms for adaptive betweenness centrality using hypergraph sketches. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1416–1425. ACM.

Zwillinger, D., and Kososka, S. 2000. *CRC Standard Probability and Statistics Tables and Formulae.* Chapter & Hall.

Twitter Weight Estimation Methodology

As noted in the main text, we take a sampling approach to estimating the distribution of weights on Twitter. In particular: we construct a uniform sample of edges, then calculate their weights. Our methodology is detailed here.

As Twitter does not explicitly represent edges as objects, we construct our edge sample by first selecting a user uniformly at random, and then selecting one of their friends or followers at random. This pair then forms an edge. Historically, one would use simple rejection sampling on the user ID space (i.e. generate an ID, check if it exists, and if not "reject" it and take another sample) to obtain a uniform sample of users. However, the changeover to 64-bit non-contiguous Snowflake IDs in 2010 (Kergl, Roedler, and Seeber 2014) means the ID space is too sparse for this approach.

Instead, we take an MCMC-style approach. We use a set of 4 parallel Metropolis-Hastings Random Walks, which asymptotically produce a uniform sample of nodes on the network, to construct the user sample. MHRWs are known to be a very efficient choice for this in practice (Gjoka et al. 2010). Our walks are rooted at users selected uniformly at random from the 1% sample of Twitter activity during October 2019, and we only begin sampling from them after they have converged to the uniform distribution. Convergence is declared when $\hat{R} < 1.02$ (see Brooks and Gelman 1998 for details). To limit the impact of autocorrelation between samples, we "thin" the random walks by keeping only every 10th sample and discarding the rest.

An additional challenge is the presence of users with extremely large numbers of followers. For example: `@neiltyson`—whose profile introduced us to this problem—has over 13 million followers, which would take a minimum of 27 days to enumerate with a single API key. We strike a balance: if a node has enough neighbors to require more than an hour of API time (300,000 neighbors), the walk backtracks to the previous node. A similar method is used to deal with "protected" users, whose neighbors cannot be enumerated via the API. The simple stopping walks used in calculating weights instead stop at these nodes.

Miscellaneous Proofs

Proof of Lemma 3

Proof. We will begin with the forward ($\implies$) case. Assume the contrary: that $\frac{x+y}{a+b} < \frac{y}{b}$ but $\frac{x}{a} \geq \frac{y}{b}$. Then $y \leq \frac{xb}{a}$. Observe that there is a $c \geq 0$ so that $y + c = \frac{xb}{a}$. Therefore:

$$\frac{x + xb/a - c}{a+b} < \frac{x}{a} - \frac{c}{b}$$

Multiplying both sides by $a + b$, we get:

$$x + \frac{xb}{a} - c < x + \frac{xb}{a} - \frac{ca}{b} - c \implies 0 < -\frac{ca}{b}$$

but $a, b > 0$ and $c \geq 0$, so $-\frac{ca}{b}$ must either be a negative value or zero. In either case, we have a contradiction.

Now let us consider the reverse ($\impliedby$) case. Again, assume the contrary: $\frac{x}{a} < \frac{y}{b}$ but $\frac{x+y}{a+b} \geq \frac{y}{b}$. As a result, we have $x + y \geq \frac{ya}{b} + y \implies x \geq \frac{ya}{b}$. There is once again a $c \geq 0$ s.t. equality holds. Substituting this, we obtain

$$\frac{ya/b + c}{a} < \frac{y}{b} \implies \frac{y}{b} + \frac{c}{a} < \frac{y}{b} \implies \frac{c}{a} < 0$$

However, $c \geq 0$ and $a > 0$—giving a contradiction. □

Proof of Theorem 3

Proof. Our proof is based on the construction of the condition. First, observe that

$$\bar{d} = \sum_{s \neq t \in V} \frac{d(s,t)}{\frac{1}{2}n(n-1)} = 2\frac{T(E)}{n(n-1)}$$

due to the fact that $|p \cap E| = |p| = d(s,t)$ in the definition of $T(E)$ (c.f. Lemma 2). Thus, we have the condition:

$$2\left(\frac{T(E)}{n(n-1)}\right)\left(\frac{n^2}{n|A| - |A|^2}\right) \leq \Phi(A)^{-1}$$

by substituting this and the definition of q_A into the inverted relation. Simplifying and using the identity $|A|(n - |A|) = |A||B|$, we get:

$$2\left(\frac{T(E)}{|A||B|}\right)\left(\frac{n}{n-1}\right) \leq \Phi(A)^{-1} \tag{2}$$

The definition of conductance used in Thm. 2 allows us to rewrite Eq. (2) as

$$2\left(\frac{T(E)}{|A||B|}\right)\left(\frac{n}{n-1}\right) \leq \frac{2\min\{|E(A)|, |E(B)|\} + 2|C|}{|C|}$$

We then apply the fact that $\frac{n}{n-1} > 1$ to convert this into a strict inequality, then simplify and rearrange to get

$$\frac{T(E)}{\min\{|E(A)|, |E(B)|\} + |C|} < \frac{|A||B|}{|C|} \leq \frac{T(C)}{|C|} = \bar{c}(C)$$

Observe that we can replace $\min\{|E(A)|, |E(B)|\}$ with $|E(A)|$ without increasing the left-hand side, and thus:

$$\frac{T(E)}{|E(A)| + |C|} < \bar{c}(C)$$

We can simplify this further by multiplying by $T(E(A) \cup C)/T(E(A) \cup C)$ and rearranging, giving us

$$\bar{c}(E(A) \cup C)\frac{T(E)}{T(E(A) \cup C)} < \bar{c}(C)$$

We use the fact that the fractional term is trivially greater than one to eliminate it, and then apply Lemma 3 to obtain

$$\bar{c}(E(A)) < \bar{c}(C)$$

A symmetric argument can be made for B. □