TellTail: Fast Scoring and Detection of Dense Subgraphs

Bryan Hooi,[1] Kijung Shin,[2] Hemank Lamba,[3] Christos Faloutsos[3]
[1]National University of Singapore
[2]KAIST
[3]Carnegie Mellon University
bhooi@comp.nus.edu.sg, kijungs@kaist.ac.kr, hlamba@cs.cmu.edu, christos@cs.cmu.edu

Abstract

Suppose you visit an e-commerce site, and see that 50 users each reviewed almost all of the same 500 products several times each: would you get suspicious? Similarly, given a Twitter follow graph, how can we design principled measures for identifying surprisingly dense subgraphs? Dense subgraphs often indicate interesting structure, such as network attacks in network traffic graphs. However, most existing dense subgraph measures either do not model normal variation, or model it using an Erdős-Renyi assumption - but this assumption has been discredited decades ago. What is the right assumption then? We propose a novel application of extreme value theory to the dense subgraph problem, which allows us to propose measures and algorithms which evaluate the surprisingness of a subgraph probabilistically, without requiring restrictive assumptions (e.g. Erdős-Renyi). We then improve the practicality of our approach by incorporating empirical observations about dense subgraph patterns in real graphs, and by proposing a fast pruning-based search algorithm. Our approach (a) provides theoretical guarantees of consistency, (b) scales quasi-linearly, and (c) outperforms baselines in synthetic and ground truth settings.

Introduction

Given an undirected, possibly weighted graph, how can we measure how surprising or anomalous a subgraph is? How can we do this in a way that exonerates subgraphs that are within the range of normal variation, but catches only subgraphs which are truly surprising? Dense subgraph detection is useful for detecting social network communities, protein families (Saha et al. 2010), follower-boosting on Twitter, and rating manipulation (Hooi et al. 2016). In these situations, it is useful to measure how surprising a dense subgraph is, to focus the user's attention on surprising or anomalous subgraphs.

Many measures exist for identifying dense subgraphs, such as average degree, modularity, etc. However, knowing that a subgraph has, for example, an average degree of 10, does not tell us how surprising it is, since we do not know if this is plausible under normal variation. To quantify how surprising a subgraph is, we need a *probabilistic* measure, that tells us, e.g., that the probability of a random subgraph of the same size being as dense as this one is 10^{-5}. Such a probabilistic measure is useful for decision making: if a subgraph is very unlikely under normal variation, we can more confidently take action such as investigating these users.

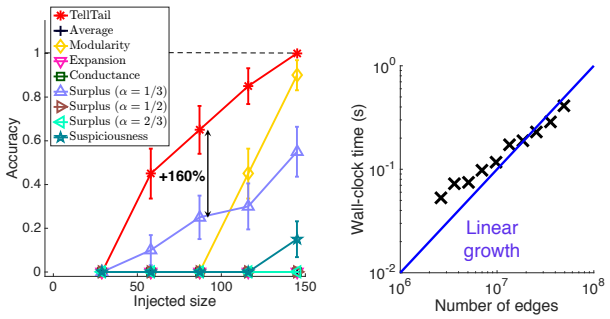
The key challenge in this process is to accurately model subgraph densities under normal behavior. In other words, what is a good null model? The simplest approach would be to assume an Erdős-Renyi model, as in (Jiang et al. 2016). However, the Erdős-Renyi model does not accurately model density patterns of real graphs: it ignores the clustering structure of graphs, as well as dense subgraphs caused by high degree nodes, such as 'hyperbolic communities' (Araujo et al. 2014) observed in real graphs.

Instead, our approach uses a novel application of extreme value theory to the dense subgraph problem, with the goal of probabilistically measuring how 'extreme' a dense subgraph is. Extreme value theory is an elegant statistical approach for modelling the distribution of extreme (or rare) events. It was initially proposed to model the distribution of flood heights, with the goal of finding the minimal dike height which would be tall enough to withstand floods with high probability. Since then, it has been applied to many types of extreme events, such as earthquakes, financial crashes, and network attacks. Extreme value theory allows us to accurately estimate the extreme **tail** of a distribution without making strong assumptions about the distribution itself. This allows us to theoretically characterize dense subgraph patterns without restrictive assumptions.

To improve its practicality, our method further makes use of two novel empirical findings about the distribution of subgraph densities in real graphs. We use these empirical findings in our measure which assesses how surprising a subgraph is, then propose a fast pruning-based algorithm for detecting dense subgraphs using this measure, in quasi-linear time. Figure 1a shows that our measure outperforms baselines in its accuracy of identifying injected subgraphs. Figure 1b shows that our search algorithm is fast and scales quasi-linearly: it took 0.41s per subgraph to find, on a graph with 49 million edges, on a laptop computer.

Our contributions are:

- **Theoretical underpinnings:** we propose a probabilistic framework (Definition 5) for finding dense subgraphs based on extreme value theory. Theorem 2 provides a guarantee of consistency.



(a) Accuracy (`UCForum` dataset) (b) Quasi-linear runtime

Figure 1: (a) Our measure outperforms baselines in accuracy of detecting injected blocks. (b) Our search algorithm scales quasi-linearly.

- **Discoveries:** We make 2 novel empirical observations on dense subgraph patterns in real graphs, which we use to speed up our measure.
- **Effectiveness:** Our approach outperforms baselines in finding injected (Figure 1a) and ground truth dense subgraphs (Table 3 and 4).
- **Algorithm:** Our TELLTAIL+ search algorithm scales quasi-linearly, and we further speed it up using a safe pruning step (based on Theorem 1).

Reproducibility: Our code and datasets are available at https://bhooi.github.io/projects/telltail.

Related Work

Measures based on internal connectivity: average degree (Goldberg 1984) is a common measure for dense subgraph detection. Variants allow for size restrictions (Andersen and Chellapilla 2009) or local subgraphs (Andersen 2010). Other measures include edge surplus (Tsourakakis et al. 2013), triangle and k-clique density (Tsourakakis 2015), discounted average degree (Yanagisawa and Hara 2018), and minimum internal degree, which defines k-cores (Shin, Eliassi-Rad, and Faloutsos 2016). Related measures underlie k-plexes (Seidman and Foster 1978) and k-trusses (Cohen 2008). (Almeida et al. 2012) evaluates density by subtracting the expected density of similar clusters.

Measures based on internal and external connectivity: external connectivity refers to the edges between the subgraph and the rest of the graph. These measures find subgraphs which are dense internally but sparsely connected externally. These include modularity (Newman 2006), Maximum, Average, and Flake-ODF (Flake, Lawrence, and Giles 2000), local density (Qin et al. 2015), and cut-based measures, such as expansion (Radicchi et al. 2004) and conductance (Shi and Malik 2000). (Miller et al. 2015) considers a spectral norm-based approach to detecting anomalous subgraphs.

Model-based measures: (Stumpf, Wiuf, and May 2005) studies the sampling properties of a network's degree distribution, particularly for scale-free networks. (Koyutürk, Szpankowski, and Grama 2007) defines two-level Erdős-Renyi (ER)-based models to assess significance of clusters in protein interaction networks. Closely related to our approach is van Leeuwen (2016) (van Leeuwen et al. 2016), which proposes a subgraph interestingness measure relative to a user's prior beliefs, where interestingness is based on the ratio of a pattern's *information content* over its *description length*. Also closely related is Jiang (2016) (Jiang et al. 2016): which uses an ER null model. In contrast, our approach uses a probabilistic framework based on extreme value theory, which allows more accurate modelling and theoretical results without relying on the Erdős-Renyi assumption (Theorem 2).

Background: Extreme Value Theory

In this section we introduce the Generalized Pareto (GP) Distribution, a family of probability distributions commonly used within extreme value theory.

The Generalized Pareto (GP) distribution has 3 parameters: its **location** μ, its **scale** σ, and its **shape** ξ (Coles et al. 2001). Its CDF is:

$$GPD_{\mu,\sigma,\xi}(x) = \begin{cases} 1 - (1 + \frac{\xi(x-\mu)}{\sigma})^{-1/\xi} & \text{if } \xi \neq 0 \\ 1 - \exp(-\frac{x-\mu}{\sigma}) & \text{if } \xi = 0 \end{cases} \quad (1)$$

for $\sigma > 0$. The GP distribution generalizes several well-known distributions:

- For $\xi > 0$ it is a Pareto distribution with $\alpha = 1/\xi$;
- For $\xi = 0$ and $\mu = 0$ it is an exponential distribution with mean σ.

Pareto distributions exhibit heavy-tailed decay (power law tails) while exponential distributions exhibit light-tailed decay (exponential tails), while the GP distribution interpolates smoothly between the two regimes. The GP distribution also has a 'universality' property (Balkema and De Haan 1974): intuitively, GP distributions can approximate the tails of almost any distribution, with error approaching zero. This makes GP distributions uniquely suitable for modeling the upper tail of subgraph mass distributions, as we will do.

Given any random variable X, let $t \in \mathbb{R}$ and define a new random variable X_t, which intuitively represents the tail of X past threshold t: thus, define F_t as the distribution of $X - t$ conditioned on $X > t$.

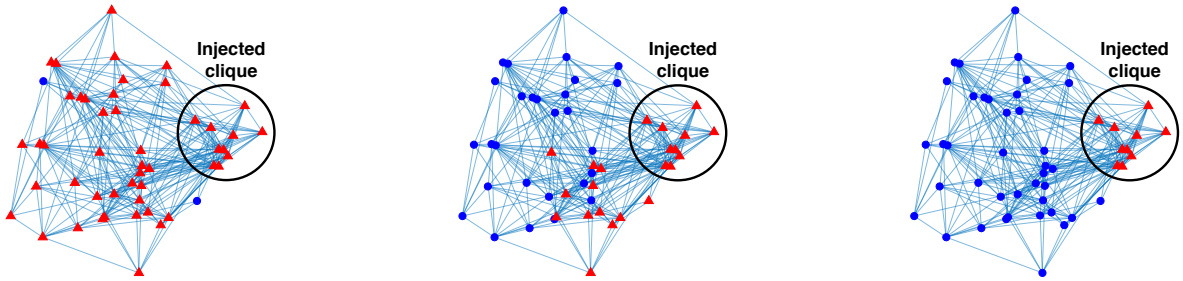
Definition 1. *The **conditional excess at threshold** t has CDF $F_t(y) = P(X - t \leq y \mid X > t)$.*

The universality property, or Pickands-Balkema-de Haan theorem (Balkema and De Haan 1974), then states that the GP distribution can approximate the tails of an arbitrary distribution:

Property 1 (Universality)**.** *Let F be any distribution function from a broad class of distributions (Embrechts, Klupelberg, and Mikosch 1999). There exists $\xi, \sigma(t)$ such that:*

$$\lim_{t \to t_{max}} \sup_x |F_t(x) - GPD_{0,\sigma(t),\xi}(x)| = 0, \quad (2)$$

where t_{max} is the right endpoint of F (t_{max} can be ∞), and GPD represents the CDF of a GP distribution.



(a) *Average Degree measure* (b) *Modularity measure* (c) TAILF *measure*

Figure 2: **TAILF avoids size bias:** it detects the injected clique. Red triangles show the subgraph detected using each measure.

Intuitively, the error of approximating the tails of a random variable by a GP distribution approaches zero as the threshold defining the tail gets larger. This justifies using GP distributions for modeling the upper tail of subgraph mass distributions, and will also allow us to prove theoretical results about our dense subgraph measure.

Problem Definition

Table 1 summarizes the symbols used in this paper.

Symbol	Interpretation
$G(V, E)$	Graph, with its vertex and edge set
n, m	Number of nodes (rep. edges)
S	Subset of nodes
k	Size of subset (i.e. $\|S\|$)
A	Adjacency matrix of G
d	$n \times 1$ vector of node degrees
B	Modularity matrix: $B = A - d \cdot d^T/(2m)$
$e(S)$	Number of edges in induced subgraph of S (also called its **mass**)
$\tilde{e}(S)$	Adjusted mass of induced subgraph of S
μ, σ, ξ	Location, scale, shape parameters of GP distribution

Table 1: Symbols and Definitions

Given a graph $G = (V, E)$, our first goal is to estimate how surprising a subgraph induced by $S \subseteq V$ is by defining a score f, which quantifies the probability of observing a subgraph at least as dense as this one:

Problem 1 (Scoring Function)**.**
Given: *graph $G = (V, E)$, subgraph induced by $S \subseteq V$*
Output: *$f(S)$, an estimate of how surprising the mass (i.e. edge count) of S is compared to the distribution of subgraphs of the same size.*

Our second goal is to develop a fast and effective algorithm for *optimizing* such a measure, so as to provide a practical approach for detecting surprisingly dense subgraphs.

Proposed Approach

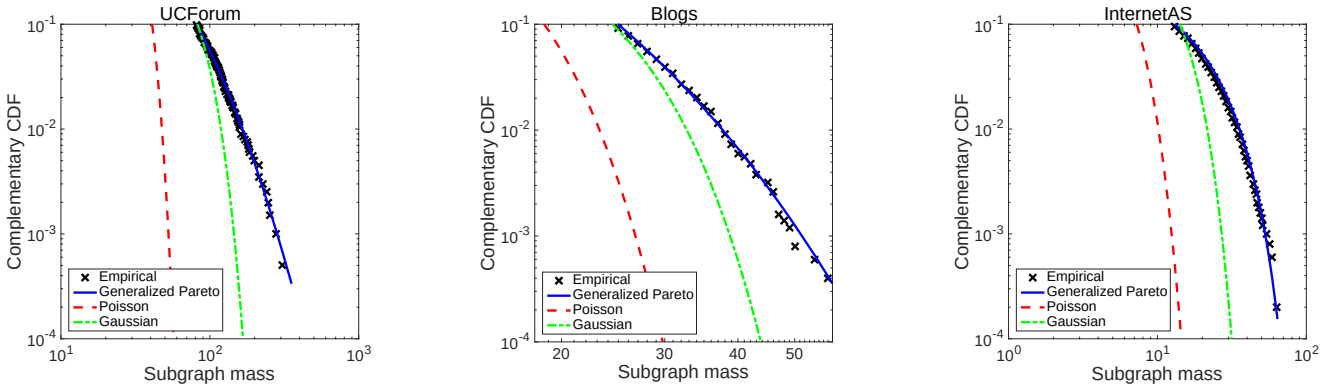
Introductory Example

Consider an Erdős-Renyi graph G of size 50 with edge probability 0.2. We select 10 random nodes and add all the edges between them to G, creating an injected clique, shown circled in Figure 2. To detect the clique, we optimize each of three measures (average degree, modularity, and TAILF) using a standard greedy local search approach (Jiang et al. 2016). Figure 2a shows that the highest average degree subgraph (red triangles) contains almost all the nodes; Figure 2b shows that the highest modularity subgraph contains around half the nodes; and Figure 2c shows that under TAILF, the densest subgraph is the injected clique.

Why does this happen? Consider average degree, $2e(S)/|S|$. The average degree of the entire graph (i.e. setting $S = V$) is $2|E|/|V|$, while the average degree of any subgraph of k nodes cannot exceed $k - 1$, even for a clique. Hence, average degree is biased in favor of larger subsets. Similarly, modularity tends to select subgraphs of size around $n/2$, as also observed by (Leskovec, Lang, and Mahoney 2010). This problem is not specific to these measures: the key issue is many measures are **size-biased**, in that their values are not comparable across sizes in a principled way. Hence a highly surprising subgraph may have a lower score than a less surprising subgraph, due to size differences. In contrast, TAILF handles this by **controlling for size**: it evaluates a subgraph by comparing it to the distribution of subgraphs of the same size. Specifically, it estimates the probability that a random subgraph of the same size would have higher density than the given subgraph. Since the measure values are in 'units' of probability, they can be compared across sizes in a principled way.

Empirical Observations

Figure 3 shows the distribution of subgraph masses in 3 real graphs. The crosses show the empirical CCDF (i.e. 1− CDF) of the mass of 5000 random subgraphs of size $k = \lfloor \sqrt{n} \rfloor$. The colored lines are maximum likelihood fits of 3 distributions to these masses. The Poisson distribution underestimates how many dense subgraphs we should observe. This makes sense, as (Jiang et al. 2016) shows that



(a) `UCForum` dataset ($n = 899$) (b) `Blogs` dataset ($n = 1.2$K) (c) `InternetAS` dataset ($n = 34$K)

Figure 3: **The GP distribution closely fits mass distributions of real graphs**: Black crosses indicate the empirical distribution of subgraph masses for subgraphs of size $k = \lfloor \sqrt{n} \rfloor$, in the form of its complementary CDF.

approximately Poisson mass distributions occur under an Erdős-Renyi model. However, the Erdős-Renyi model ignores the clustering effects present in real graphs. Hence, in real graphs, much denser subgraphs exist. Figure 3 shows that Gaussian distributions also decay too quickly, while the GP distribution fits the empirical distribution closely, due to its 'universality' property. For space reasons, plots for our other datasets are in our supplement[1]. To summarize:

Observation 1. *Upper tails of subgraph mass distributions of real graphs closely follow a GP distribution.*

Proposed Measures

Hence, we now propose measures which approximate a subgraph mass distribution using a GP distribution, and use it to estimate the surprisingness of each subgraph.

TAIL Measure

TAIL estimates this GP distribution via sampling. Given a subgraph S of size $|S| = k$ and mass $e(S)$, we sample N uniformly random subsets of V of size k. For $\epsilon = 0.1$, we fit a GP distribution using maximum likelihood (Grimshaw 1993) to the largest $\lfloor \epsilon N \rfloor$ masses. The surprisingness of S is the CDF of this GP distribution, evaluated at $e(S)$:

Definition 2 (TAIL Measure)**.**

$$f(S) = GPD_{\hat{\mu},\hat{\sigma},\hat{\xi}}(e(S))$$

where $\hat{\mu}, \hat{\sigma}, \hat{\xi}$ are the maximum likelihood GP fit.

Adjusting for Degree: the TAILD Measure

How do we identify subgraphs that are not just dense, but also sparsely connected to the rest of the graph? For example, in a user-product review graph, such subgraphs could suggest rating manipulation.

Instead of mass $e(S)$ of a subset S, we compute its **adjusted mass** $\tilde{e}(S)$, its mass minus its expectation if edges are rewired randomly. As in the modularity measure (Newman 2006), the expectation of $e(S)$ is $d(S)^2/(4m)$, where $d(S)$ is the sum of degrees of nodes in S.

Definition 3 (Adjusted mass)**.** *The adjusted mass of subgraph S is:*

$$\tilde{e}(S) = e(S) - d(S)^2/(4m)$$

TAILD differs from TAIL only in that it uses adjusted mass instead of mass:

Definition 4 (TAILD Measure)**.**

$$f(S) = GPD_{\hat{\mu},\hat{\sigma},\hat{\xi}}(\tilde{e}(S))$$

where $\hat{\mu}, \hat{\sigma}, \hat{\xi}$ are maximum likelihood GP parameters for the distribution of adjusted masses.

Dense Subgraph Power Laws

TAIL and TAILD require a time-consuming sampling step. How do we speed them up? We first observe near-power law empirical patterns, which we use to greatly speed up TAILD. Let $\mu(k)$ be the GP location parameter as a function of subgraph size k (and similarly define $\sigma(k)$). For GP distributions fitted to adjusted mass $\tilde{e}(S)$, we observe near-power law patterns for $\mu(k)$ and $\sigma(k)$:

$$\mu(k) = \mu_0 k^{\alpha} \tag{3}$$

$$\sigma(k) = \sigma_0 k^{\beta}. \tag{4}$$

where μ_0, σ_0, α and β are constants to be determined.

Plotting $\mu(k)$ and $\sigma(k)$ against k on a log-log plot, this takes the form of a straight line. Figure 4 shows such plots for the `InternetAS` graph. $\mu(k)$ and $\sigma(k)$ closely follow the near-power law patterns in Eq. (3) and (4), with R^2 of 1.00 and 0.99 (where $R^2 = 1$ indicates a perfect linear fit). Table 2 shows that the same pattern holds across many datasets, with R^2 very near 1. α and β are close to 0.9 and 0.8, while ξ is close to -0.1.

Observation 2 (Dense Subgraph Power Laws)**.** *GP parameters of adjusted mass distributions in real graphs closely follow $\mu(k) = \mu_0 k^{0.9}$ and $\sigma(k) = \sigma_0 k^{0.8}$.*

[1] https://bhooi.github.io/projects/telltail/supplement.pdf

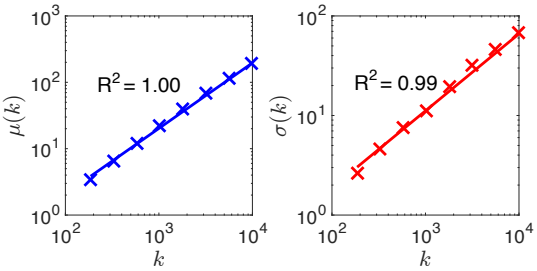


Figure 4: **Power law patterns in GP parameters**: in the `InternetAS` graph.

Dataset	μ slope	R^2	σ slope	R^2	ξ
UCForum	0.87	0.99	0.63	0.98	-0.02
Email	0.82	0.99	0.74	0.98	-0.12
Blogs	0.87	0.99	0.76	0.99	-0.05
Petster	0.85	0.99	0.78	0.99	-0.11
PGP	0.86	0.99	0.81	0.99	-0.11
AstroPh	0.88	0.99	0.85	0.99	-0.12
DBLP	0.87	0.99	0.81	0.99	-0.13
InternetAS	0.98	1.00	0.76	0.99	-0.07

Table 2: **Power law patterns across multiple datasets:** as a function of k, $\mu(k)$ and $\sigma(k)$ closely follow the power-law patterns, (3) and (4). Dataset information is given in our supplementary information.

TAILF: Fast Scoring using Power Laws

Using Observation 2, we now propose our main TAILF measure, which avoids sampling, providing large speedup. We ignore strictly increasing transformations of the measure (e.g. multiplication by a constant), as they do not change the relative order of different subsets.

As in TAILD, the surprisingness of S is the CDF of a GP distribution, evaluated at the adjusted mass of S. As observed previously, ξ remains fairly constant empirically, so we approximate it as a constant in k. Then $\text{GPD}_{\xi,\mu,\sigma}(\tilde{e}(S))$ is a function of $\frac{\tilde{e}(S)-\mu}{\sigma}$; moreover, it is strictly increasing function of $\frac{\tilde{e}(S)-\mu}{\sigma}$ within its support $(0, -\frac{1}{\xi})$. Thus, we can use $\frac{\tilde{e}(S)-\mu}{\sigma}$ in place of $\text{GPD}_{\xi,\mu,\sigma}(\tilde{e}(S))$. Substituting the power law patterns (3) and (4) and ignoring the constant σ_0 gives our TAILF expression:

Definition 5 (Proposed TAILF Measure)**.**

$$\boxed{f(S) = \frac{\tilde{e}(S) - \mu_0 k^\alpha}{k^\beta}}$$

Following Observation 2, for simplicity we use $\alpha = 0.9$ and $\beta = 0.8$ as fixed default values, though in practice these values could also be set by estimating them as in Table 2 for any given graph. For space reasons, we explain our approach for estimating μ_0 in our supplementary material.

TELLTAIL Search Algorithm

TELLTAIL uses randomized local search on the TAILF measure, starting from random seeds. Let S be the current set of nodes. To efficiently find good candidates for insertion, we use a max-heap H_I storing all neighbors of nodes in S. The key (or priority) of node i is $e_S(i) - d(i)/n$, where $e_S(i)$ is the number of neighbors of i in S, and $d(i)$ is the degree of i. Intuitively, we prefer to insert nodes with many neighbors in S, and secondarily prefer nodes with lower degree (the latter is typically less important, so the division by n means that it is only used to break ties). For deletions, we want the reverse, so we use a min-heap H_D storing all nodes in S instead, with the same keys.

TELLTAIL is given in Algorithm 1, omitting the steps in green. We first initialize the heaps (Line 9 to 10). In each iteration we extract the candidate insertions and deletions from the heaps (Line 13), then greedily modify S to maximize TAILF (Line 15). We then update the keys of nodes in the heaps (Line 16), inserting nodes at the same time if they are not already in the heap, and have not been popped so far. In Line 16, when a node is added or deleted from S, the degree in S of its neighbors change, so we recompute their priority (i.e. key). We only need to update the neighbors of the inserted or deleted node, due to our definition of keys. We use Fibonacci heaps, taking $O(\log n)$ per key update.

Improved TELLTAIL+ Algorithm

TELLTAIL+ uses pruning to improve efficiency. We will show that the form of TAILF allows us to safely prune (i.e. remove) nodes, while guaranteeing that we never prune nodes that are in the subset that optimizes TAILF.

Definition 6 (Deviation of a node)**.** *The* **deviation** *of node i is the sum of the modularity matrix $B = A - d \cdot d^T/(2m)$ between it and its neighbors:*

$$Dev_i = \sum_{i,j \in E} B_{ij} \tag{5}$$

Intuitively, deviation measures how much a node can 'contribute' to the adjusted mass of a subset. Let $S^* = \arg\max_{S \subseteq V} \text{TAILF}(S)$ and $s = \lfloor n/2 \rfloor$. Define:

$$\Delta(S) = (s^\beta - (s-1)^\beta)\text{TAILF}(S) + \mu_0(s^\alpha - (s-1)^\alpha) \tag{6}$$

Our pruning is based on the following theorem. For space reasons, all proofs are in our supplementary material.

Theorem 1 (Safe Pruning)**.** *For any node i, if $i \in S^*$, then:*

$$Dev_i \geq \Delta(S^*) \geq \Delta(S) \; \forall \; S \subseteq V \tag{7}$$

So, we can prune nodes with deviation $< \Delta(S)$, for **any** *S.*

TELLTAIL+ first uses the neighborhoods of t randomly chosen nodes to bound for pruning (Lines 2 to 5). When a new best S is found, we prune the nodes using the improved threshold (Line 18). We recommend $t = 500$ as it has minimal impact on running time in practice, while providing a large enough sample size.

Algorithm 1 TELLTAIL and TELLTAIL+

1: **Input**: Graph G, no. of neighborhoods t, repetitions r
2: **for** $i = 1$ **to** t **do**
3: Let N_i be the neighborhood of a random node
4: Prune nodes with deviation less than $\Delta(N_i)$
5: **end for**
6: **for** $i = 1$ **to** r **do**
7: Sample S_i to contain a single random node $\{s\}$
8: // Heaps with candidates for insertion / deletion
9: Initialize max-heap H_I containing all neighbors of s
10: Initialize min-heap H_D containing s
11: **while** H_I and H_D are nonempty **do**
12: // Get insertion and deletion candidates
13: Pop n_I from H_I and n_D from H_D
14: // Greedy local step
15: $S_i \leftarrow \arg\max_{S' \in \{S_i \cup n_I, S_i \setminus n_D, S_i\}}$ TAILF(S')
16: Update keys in H_I and H_D
17: **end while**
18: Prune nodes with deviation less than $\Delta(S_i)$
19: **end for**
20: **Return** $\arg\max_{S' \in \{S_1, \cdots, S_r\}}$ TAILF(S')

Scalability

We store only the graph edges and heaps, using linear ($O(m)$) memory ($O(n)$ if we only load the adjacency lists of n_I and n_D at a time, and store the rest of the graph on disk). For running time, the bottleneck is the heap key updates (Line 16); each update in a Fibonacci heap takes $O(\log n)$, each edge only has a constant number of updates across it (since popped nodes are not re-inserted), so at most $O(m)$ updates occur, taking $O(m \log n)$ time.

Theoretical Results

Consistency

Our estimate for the surprisingness of subgraphs is consistent: as n increases, the error of our estimate approaches zero. Formalizing this uses a graphon $\mathcal{G}$, a flexible generative model for graphs (see (Orbanz and Roy 2015)). TAIL outputs $\hat{F}$, the CDF of the GP distribution. Our consistency theorem states that $\hat{F}$ converges to F, the true surprisingness under $\mathcal{G}$, in relative error, as $n \to \infty$.

Theorem 2 (Consistency). *Let* $y_n = -\sigma(\hat{\mu}_n)(1-z)/\xi$. *Then as* $n \to \infty$, *there exists a sequence of* $N_n \to \infty, \epsilon_n \to 0$ *such that:*

$$\frac{1 - \hat{F}(\hat{\mu}_n + y_n)}{1 - F(\hat{\mu}_n + y_n)} \xrightarrow{P} 1 \tag{8}$$

where $\xrightarrow{P}$ *denotes convergence in probability.*

Experiments

Our experiments answer the following questions:

- **Q1. Scalability:** how does TELLTAIL+ scale?
- **Q2. Accuracy of Measure:** does TAILF accurately identify injected dense subgraphs?
- **Q3. Real-World Effectiveness:** do TAILF and TELLTAIL+ accurately find ground-truth communities?

Dataset details are in our supplementary information.

Q1. Scalability

To get graphs that follow real-world patterns, rather than using a synthetic generator, we use the real `WikiP` graph and extract subsets of its nodes. Hence, we run TELLTAIL+ on random subsets of the `WikiP` graph containing $\lfloor (0.85)^k \rfloor$ fraction of nodes for $k = 0, \ldots, 9$, averaging each over 10 trials. Figure 1b shows that TELLTAIL+ scales near-linearly. TELLTAIL+ is fast, taking 0.41 seconds (using a laptop computer) on the full graph of 49.0 million edges.

Pruning Effectiveness How effective are the pruning steps (i.e. the green lines in Algorithm 1) for reducing the graph size? Figure 5 shows that TELLTAIL+ reduces the node count by 45, 18 and 52 times on the largest `InternetAS`, `WikiP` and `Twitter` datasets.

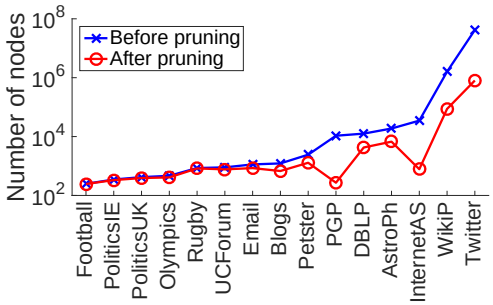


Figure 5: **TELLTAIL+ reduces graph size.**

Q2. Accuracy of Measure

In this section, we evaluate the accuracy of our TAILF measure in distinguishing injected dense subgraphs from normal subgraphs. Note that our goal here is to evaluate its accuracy as a measure, against baseline measures, rather than against detection algorithms. Hence, our baseline measures are chosen to provide a mix of standard internal density measures (average degree, modularity), measures which consider internal and external density (expansion, conductance) and recent dense subgraph measures (edge surplus (Tsourakakis et al. 2013), suspiciousness (Jiang et al. 2016)).

While edge density $e(S)/\binom{|S|}{2}$ is intuitive, it does not directly work as a single edge attains the maximum possible density of 1. Average Degree, Modularity and Edge Surplus are different approaches for adjusting edge density to alleviate this problem. For the Edge Surplus baseline we use $\alpha = 1/2$; for space reasons, additional results varying α can be found in our supplement.

For each graph, in each trial we inject a dense subgraph, whose nodes are uniformly sampled at random from the graph's existing nodes, and whose size is one of $\lfloor\sqrt{n}\rfloor, 2\lfloor\sqrt{n}\rfloor, \cdots, 5\lfloor\sqrt{n}\rfloor$ (we obtain separate results for each of these sizes). We choose these values as it tends to be the range where performance varies meaningfully. We inject edges uniformly at random to this subgraph, adding additional density equal to twice the average density of the whole graph (i.e. density $2m/\binom{n}{2}$). A good measure should distinguish 'unnatural' (injected) subgraphs from 'natural'

(non-injected) subgraphs. Thus, in each trial we generate 500 random null subgraphs: their sizes follow an evenly distributed grid of sizes between 1 and n, while their nodes are chosen uniformly at random. A measure is successful on a trial if it gives a higher value to the injected subgraph than to all the null subgraphs. We repeat each trial 20 times to generate error bars (representing 1 standard deviation).

Figure 1a shows accuracy (i.e. the fraction of trials where the injected subgraph received the highest score) against injected size on the UCForum dataset ($n = 899$). Further results are in our supplement, and similarly show large improvements for TAILF. Why does this happen? In our introductory example we introduced **size-bias**: many measures do not compare across sizes in a balanced manner: e.g. average degree chooses large subsets of almost the whole graph, while modularity chooses subsets of around half the graph size. TAILF evaluates subgraphs of each size using an accurate GP model, allowing fair comparison across sizes.

Q3. Real-World Effectiveness

We next evaluate TAILF on topically curated Twitter graphs with ground truth dense communities corresponding to sports teams or political parties manually labelled by (Greene and Cunningham 2013). On each dataset, we randomly generate a pool of null subgraphs: for each size (twice), we generate 200 random subgraphs of that size and add the community with highest mass into the pool. This ensures that the null subgraphs are reasonably dense. We then compute each measure on the ground truth and null subgraphs. For each measure, its accuracy is its precision at k, i.e. the fraction of ground truth communities in the top k subsets according to the measure, where k is the true number of ground truth communities. Table 3 shows that TAILF clearly outperforms the baselines, and has more consistent performance across datasets.

	TAILF	Average	Modularity	Expansion	Conduct.	Surplus	Susp.
Football	**1.00**	0.00	0.96	0.71	0.55	**1.00**	0.68
PolIE	**1.00**	0.00	0.94	0.87	0.86	0.75	0.38
PolUK	**0.96**	0.46	0.87	0.54	0.87	0.71	0.29
Olympics	**1.00**	0.00	0.85	0.74	0.67	**1.00**	0.63
Rugby	**0.97**	0.00	0.85	0.90	0.80	0.94	0.55

Table 3: **TAILF outperforms baselines** in identifying ground truth communities.

Effectiveness of our Algorithms We now evaluate TELLTAIL and TELLTAIL+ on detecting ground truth communities in the same topical Twitter graphs. We use the same baselines, each optimized using standard local search (Jiang et al. 2016). We run each algorithm 10 times and choose the subset which it gave the highest score to. We evaluate each method based on the largest Jaccard similarity between its output and any ground truth community.

The results are shown in Table 4. Averaging over the 5 graphs, TELLTAIL and TELLTAIL+ outperforms the best-performing baseline by 0.34 and 0.31 respectively. This occurs likely due to their use of TAILF, which also performs better in identifying ground truth communities in Table 3.

	TELLTAIL	TELLTAIL+	Average	Modularity	Expansion	Conduct.	Surplus	Susp.
Football	**0.78**	0.67	0.11	0.19	0.06	0.11	0.15	0.13
PolIE	**0.90**	0.82	0.42	0.39	0.02	0.41	0.41	0.39
PolUK	0.82	**0.84**	0.81	0.75	0.01	0.36	0.53	0.76
Olympics	0.68	**0.72**	0.20	0.21	0.11	0.15	0.29	0.22
Rugby	0.47	**0.49**	0.43	0.24	0.36	0.17	0.25	0.28

Table 4: **TELLTAIL and TELLTAIL+ outperform baselines** in detecting ground truth communities.

Case Study on Twitter Data

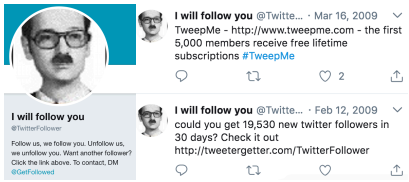


Figure 6: **TELLTAIL+ detects a follower-boosting scheme:** we detect a large group of accounts, 85% of which are directly connected to the pictured account.

We apply TELLTAIL+ on a Twitter following graph of 41.7M users and 1.47B follows (Kwak et al. 2010). TELLTAIL+ detects a group of 4334 users with edge density 75%, which is highly suspicious in itself. To further analyze the users in this block, we randomly sample of 400 of these users. We then exclude users whose accounts were deleted, suspended, or not searchable, resulting in 280 remaining users. Of these, we use a script to search for which of these users have made tweets linking 'tweepme.com' or 'tweetergetter.com,' which are two known follower-buying services in which users who purchase the service follow one another. We find that 125 (45%) of these users have made such tweets. Moreover, 85% of the users in this block are directly adjacent to the user in Figure 6, who advertises over 20 follower-boosting services, such as 'TweepMe' and 'TweeterGetter'.

Conclusion

In this paper, we introduced principled measures and algorithms for dense subgraphs. A persistent question in graph analysis is: how can we fairly compare subgraphs of different sizes and densities? We answer this by using GP distributions to model the tail of subgraph mass distributions. Our contributions are as follows:

- **Theoretical underpinnings:** We propose a probabilistic framework (Definition 5) for finding dense subgraphs based on extreme value theory. Theorem 2 provides a guarantee of consistency.

- **Discoveries:** We make 2 novel empirical observations on dense subgraph patterns in real graphs, which we use to speed up our measure.
- **Effectiveness:** Our approach outperforms baselines in finding injected (Figure 1a) and ground truth dense subgraphs (Table 3 and 4).
- **Algorithm:** Our TELLTAIL+ search algorithm scales quasi-linearly, and also adds a safe pruning step (based on Theorem 1).

References

Almeida, H.; Neto, D. G.; Meira Jr, W.; and Zaki, M. J. 2012. Towards a better quality metric for graph cluster evaluation. *Journal of Information and Data Management* 3(3):378.

Andersen, R., and Chellapilla, K. 2009. Finding dense subgraphs with size bounds. In *WAW*.

Andersen, R. 2010. A local algorithm for finding dense subgraphs. *TALG* 6(4):60.

Araujo, M.; Günnemann, S.; Mateos, G.; and Faloutsos, C. 2014. Beyond blocks: Hyperbolic community detection. In *ECML-PKDD*, 50–65. Springer.

Balkema, A. A., and De Haan, L. 1974. Residual life time at great age. *The Annals of probability* 792–804.

Cohen, J. 2008. Trusses: Cohesive subgraphs for social network analysis. *National Security Agency Technical Report* 16.

Coles, S.; Bawa, J.; Trenner, L.; and Dorazio, P. 2001. *An introduction to statistical modeling of extreme values*, volume 208. Springer.

Embrechts, P.; Kluppelberg, C.; and Mikosch, T. 1999. Modelling extremal events. *British Actuarial Journal* 5(2):465–465.

Flake, G. W.; Lawrence, S.; and Giles, C. L. 2000. Efficient identification of web communities. In *KDD*, 150–160. ACM.

Goldberg, A. V. 1984. *Finding a maximum density subgraph*. Technical Report.

Greene, D., and Cunningham, P. 2013. Producing a unified graph representation from multiple social network views. In *ACMWeb*, 118–121. ACM.

Grimshaw, S. D. 1993. Computing maximum likelihood estimates for the generalized pareto distribution. *Technometrics* 35(2):185–191.

Hooi, B.; Song, H. A.; Beutel, A.; Shah, N.; Shin, K.; and Faloutsos, C. 2016. Fraudar: bounding graph fraud in the face of camouflage. In *KDD*, 895–904. ACM.

Jiang, M.; Beutel, A.; Cui, P.; Hooi, B.; Yang, S.; and Faloutsos, C. 2016. Spotting suspicious behaviors in multimodal data: A general metric and algorithms. *TKDE* 28(8):2187–2200.

Koyutürk, M.; Szpankowski, W.; and Grama, A. 2007. Assessing significance of connectivity and conservation in protein interaction networks. *Journal of Computational Biology* 14(6):747–764.

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*, 591–600. AcM.

Leskovec, J.; Lang, K. J.; and Mahoney, M. 2010. Empirical comparison of algorithms for network community detection. In *WWW*, 631–640. ACM.

Miller, B. A.; Beard, M. S.; Wolfe, P. J.; and Bliss, N. T. 2015. A spectral framework for anomalous subgraph detection. *IEEE Transactions on Signal Processing* 63(16):4191–4206.

Newman, M. E. 2006. Modularity and community structure in networks. *PNAS* 103(23):8577–8582.

Orbanz, P., and Roy, D. M. 2015. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE TPAMI* 37(2):437–461.

Qin, L.; Li, R.-H.; Chang, L.; and Zhang, C. 2015. Locally densest subgraph discovery. In *KDD*. ACM.

Radicchi, F.; Castellano, C.; Cecconi, F.; Loreto, V.; and Parisi, D. 2004. Defining and identifying communities in networks. *PNAS* 101(9):2658–2663.

Saha, B.; Hoch, A.; Khuller, S.; Raschid, L.; and Zhang, X.-N. 2010. Dense subgraphs with restrictions and applications to gene annotation graphs. In *RECOMB*.

Seidman, S. B., and Foster, B. L. 1978. A graph-theoretic generalization of the clique concept. *Journal of Mathematical sociology* 6(1):139–154.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *PAMI* 22(8):888–905.

Shin, K.; Eliassi-Rad, T.; and Faloutsos, C. 2016. Corescope: Graph mining using k-core analysis - patterns, anomalies and algorithms. In *ICDM*.

Stumpf, M. P.; Wiuf, C.; and May, R. M. 2005. Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proceedings of the National Academy of Sciences* 102(12):4221–4224.

Tsourakakis, C.; Bonchi, F.; Gionis, A.; Gullo, F.; and Tsiarli, M. 2013. Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In *KDD*, 104–112. ACM.

Tsourakakis, C. 2015. The k-clique densest subgraph problem. In *Proceedings of the 24th international conference on world wide web*, 1122–1132. International World Wide Web Conferences Steering Committee.

van Leeuwen, M.; De Bie, T.; Spyropoulou, E.; and Mesnage, C. 2016. Subjective interestingness of subgraph patterns. *Machine Learning* 105(1):41–75.

Yanagisawa, H., and Hara, S. 2018. Discounted average degree density metric and new algorithms for the densest subgraph problem. *Networks* 71(1):3–15.