Supplementary Document for TellTail

Bryan Hooi, Kijung Shin, Hemank Lamba, Christos Falousos

1 Proof of Safe Pruning

Let $S^* = \arg\max_{S \subseteq V} \textsc{TellTail}(S)$ and $s = \lfloor n/2 \rfloor$. Define:

$$\Delta(S) = (s^\beta - (s-1)^\beta)\textsc{TellTail}(S) + \mu_0(s^\alpha - (s-1)^\alpha) \tag{1.1}$$

Theorem 1.1. (Safe Pruning) *For any node i, if $i \in S^*$, we have:*

$$Dev_i \geq \Delta(S^*) \geq \Delta(S) \; \forall \; S \subseteq V \tag{1.2}$$

Thus, we can prune nodes with deviation less than $\Delta(S)$, for **any** *S.*

Proof. Define $k_* = |S^*|$, and $\gamma = k_*^\alpha - (k_* - 1)^\alpha$, and $\delta = k_*^\beta - (k_* - 1)^\beta$.

Since Dev_i is the sum of positive modularity terms along the ith row of B, Dev_i is an upper bound for how much adjusted mass the ith row can contribute to $\tilde{e}(S^*)$. Hence:

$$\textsc{TellTail}(S^*) \geq \textsc{TellTail}(S^* \setminus \{i\}) \tag{1.3}$$

$$\implies \frac{\tilde{e}(S^*) - \mu_0 \cdot k_*^\alpha}{k_*^\beta} \geq \frac{\tilde{e}(S^*) - \text{Dev}_i - \mu_0 \cdot (k_* - 1)^\alpha}{(k_* - 1)^\beta} \tag{1.4}$$

$$\implies \frac{\tilde{e}(S^*) - \mu_0 \cdot k_*^\alpha}{k_*^\beta} \geq \frac{\tilde{e}(S^*) - \text{Dev}_i - \mu_0 \cdot k_*^\alpha + \mu_0 \cdot \gamma}{k_*^\beta - \delta} \tag{1.5}$$

$$\implies -\delta \cdot \tilde{e}(S^*) + \mu_0 \cdot \delta \cdot k_*^\alpha \geq -k_*^\beta \cdot \text{Dev}_i + k_*^\beta \cdot \mu_0 \cdot \gamma \tag{1.6}$$

$$\implies \textsc{TellTail}(S^*) \leq \frac{\text{Dev}_i - \mu_0 \gamma}{\delta} \tag{1.7}$$

$$\implies \text{Dev}_i \geq \delta \cdot \textsc{TellTail}(S^*) + \mu_0 \cdot \gamma \tag{1.8}$$

$$\implies \text{Dev}_i \geq \Delta(S^*) \tag{1.9}$$

The second inequality $\Delta(S^*) \geq \Delta(S) \; \forall \; S \subseteq V$ follows from $\textsc{TellTail}(S^*) \geq \textsc{TellTail}(S)$, since this implies that $\Delta(S^*) \geq \Delta(S) \; \forall \; S \subseteq V$.

2 Proof of Consistency

Theorem 2.1. (Consistency) *Let G be a graph drawn from $\mathcal{G}$, and fix k. Define any fixed $z > 0$, and let $y_n = -\sigma(\hat{\mu}_n)(1-z)/\xi$. Then as $n \to \infty$, there exists a sequence[1] of $N_n \to \infty, \epsilon_n \to 0$ such that:*

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

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

Fix N and let $n \to \infty$. Consider an n-node graph $G \sim \mathcal{G}$ and random subsets $S_1, \ldots, S_N$ of size k. For any i, j, the probability that S_i and S_j are completely disjoint is $(\frac{n-k}{n})^k \to 1$ as $n \to \infty$ (recall that k is fixed). Extending this to all of the pairs of i, j by union bound, the probability that all of $S_1, \ldots, S_N$ are disjoint goes to 1 as $n \to \infty$. This implies that with high probability, $S_1, \ldots, S_N$ are disjoint and hence are i.i.d. samples of size k from $\mathcal{G}$, and their masses (denoted by $m_1, \ldots, m_N$) are i.i.d. samples from F. Note that the event in which $S_1, \ldots, S_N$ are non-disjoint has probability converging to zero, and since the statement we want to prove is about convergence in probability, this event can be ignored.

At this point, $m_1, \ldots, m_N$ is an i.i.d. sample from a distribution F, and our algorithm $\textsc{Tail}$ estimates a GP distribution using maximum likelihood from this sample. [3] shows that under these conditions, the maximum likelihood procedure produces consistent estimators of the tail probabilities of the distribution F. Formally:

$$\frac{1 - \hat{F}(\hat{\mu_n} + y_n)}{1 - F(\hat{\mu_n} + y_n)} \xrightarrow{P} 1$$

i.e. $\hat{F}$ converges to F, measured in terms of relative error with respect to the CCDF $1 - F$.

3 TellTail: Estimator for μ_0

Let S be a uniformly random subgraph of size k.

[1] N_n, ϵ_n, $\hat{\mu}_n$ denote the original variables $(N, \epsilon, \hat{\mu})$ indexed over runs corresponding to different values of n. ξ and $\sigma(\cdot)$ are from (**??**).

LEMMA 3.1. *The mean and variance of $\tilde{e}(S)$ are:*

$$\mathbb{E}(\tilde{e}(S)) = p_2 S_1$$

$$\text{Var}(\tilde{e}(S)) = p_2 S_2 + p_3(S_3 - 2S_2) + p_4(S_1^2 + S_2 - S_3) - (p_2 S_1)^2 \tag{3.10}$$

Proof. Define random variables $Z_{ij} = B_{ij} 1\{i \in S, j \in S\}$. Note that $\tilde{e}(S) = \sum_{i<j} Z_{ij}$. Substituting this into $\mathbb{E}(\tilde{e}(S))$ and $\text{Var}(\tilde{e}(S))$ and expanding with further computation gives the result.

We now derive our estimator (3.11) for μ_0:

$$\mu_0 = \left(\frac{n}{2}\right)^{-\alpha} (p_2 S_1 + z_{1-\epsilon}(p_2 S_2 + p_3(S_3 - 2S_2) + p_4(S_1^2 + S_2 - S_3) - (p_2 S_1)^2)^{1/2}) \tag{3.11}$$

Let $k = n/2$. For subgraphs of this large size, $\tilde{e}(S)$ is the sum of a large number of small values: $\tilde{e}(S) = \sum_{i<j} Z_{ij}$, which recalling the Central Limit Theorem, suggests approximating $\tilde{e}(S)$ with a normal distribution. From our initial definition of the GP distribution and parameters, $\mu(k)$ is the minimum value in the GP distribution's support. Since we threshold the data at its $(1-\epsilon)$-quantile, $\mu(k)$ is the $(1-\epsilon)$-quantile of the subgraph mass distribution. For $k = n/2$, this distribution is approximately Gaussian, so the corresponding $(1-\epsilon)$-quantile is:

$$\mu(n/2) \approx \mathbb{E}(\tilde{e}(S)) + z_{1-\epsilon}\sqrt{\text{Var}(\tilde{e}(S))} \tag{3.12}$$

The power-law plot for μ passes through the point $(n/2, \mu(n/2))$, so substituting into the Dense Subgraph Power Law, its intercept is $\mu_0 = \mu(n/2)/(n/2)^\alpha$. Combining this with (3.10) and (3.12) gives the final result.

4 Computing S_1, S_2, S_3

Recall that $S_1 = \sum_{i<j} B_{ij}$, $S_2 = \sum_{i<j} B_{ij}^2$, $S_3 = \sum_{i=1}^n (\sum_{j=1}^n B_{ij})^2$.

Computing S_1 to S_3 naively is $O(n^2)$, by this can be sped up to linear time using matrix operations.

LEMMA 4.1. *S_1 to S_3 can be computed in $O(m)$ time.*

Proof. However, $B = A - d \cdot d^T/(2m)$, a sum of a sparse and a low rank matrix. This substitution allows us to rewrite the expressions for S_1 to S_3:

$$S_1 = \frac{\sum_{i=1}^n d_i^2}{4m} \tag{4.13}$$

$$S_2 = \sum_{i<j} A_{ij}^2 - \frac{d^T A d}{2m} + \frac{(\sum_{i=1}^n d_i^2)^2 - \sum_{i=1}^n d_i^4}{8m} \tag{4.14}$$

$$S_3 = \frac{\sum_{i=1}^n d_i^4}{4m^2} \tag{4.15}$$

The only terms that require matrix operations are $\sum_{i<j} A_{ij}^2$ and $d^T A d$. Both can be computed in $O(m)$ time using standard sparse matrix operations.

5 Proof of NP Completeness

In this section, we show that maximizing TELLTAIL is NP-complete. Let $\tilde{e}_G(S)$ denote the adjusted mass of subset S with respect to the graph G, i.e. $\tilde{e}_G(S) = e(S) - d(S)^2/(4m)$, all with respect to G. Define the following two problems:

PROBLEM 1. (MODULARITY) *Given a graph G, does there exist a subset S of its nodes that such $\tilde{e}_G(S) > 0$?*

The NP-hardness of modularity maximization was first shown by [1], though this differs from the formulation here. The NP-hardness of this exact formulation was shown by [2]. They do this by reducing the known NP-hard PARTITION problem, of partitioning n integers $x_1, \cdots, x_n$ into two subsets of equal sum, to the MODULARITY problem. To do this, they show that given $x_1, \cdots, x_n$, we can construct a graph such that a subset S of positive modularity exists iff there exists a partition of $x_1, \cdots, x_n$ into two halves of equal sum, which completes the reduction and establishes the NP-hardness of MODULARITY.

The problem we are interested in is:

PROBLEM 2. (TELLTAILPROB) *Given a graph G', does there exist a subset S of its nodes such that $\text{TELLTAIL}_{G'}(S) > 0$?*

We now show our main NP-completeness result:

THEOREM 5.1. TELLTAILPROB *is NP-complete.*

Proof. Given a subset S of the nodes of G, we can compute TELLTAIL(S) in polynomial time. Thus, TELLTAILPROB can be *verified* in polynomial time, and is therefore in NP. It remains to establish that TELLTAILPROB is NP-hard.

We do this by reducing MODULARITY to TELLTAILPROB: given an algorithm A' that solves TELLTAILPROB, we show that it can be used as a subroutine in a polynomial-time algorithm A to solve MODULARITY. Then, since we know that MODULARITY is NP-hard, this would imply that TELLTAILPROB is NP-hard as well.

Consider an instance of MODULARITY with graph G. Construct a new graph G' by adding r extra nodes to G, in which the extra nodes have no edges attached to them. Note then that for any subset S of the nodes of G, the adjusted mass of S is the same when computed with respect to either G or G': this is because in the formula $\tilde{e}(S) = e(S) - d(S)^2/(4m)$, none of the terms ($e(S)$, $d(S)$ or m) differ between G and G'.

Consider Eq. (3.11) for μ_0. Each of the p_i is at most 1, and the S_i are all constant as we increase r, which we can verify from Eq. (6) to (8) of the original paper. Then, since G' has $n+r$ nodes, we have from Eq. (3.11) that $\mu_0 \leq (\frac{n+r}{2})^{-\alpha} B$, for some $B > 0$ that does not depend on r.

Set $r > 2(4mn^\alpha B)^{(1/\alpha)}$. Then in G', we have:

$$\begin{aligned}\mu_0 &\leq \left(\frac{n+r}{2}\right)^{-\alpha} B \\ &< \left(\frac{r}{2}\right)^{-\alpha} B \\ &= (4mn^\alpha B)^{(1/\alpha)\cdot(-\alpha)} B \\ &= \frac{1}{4mn^\alpha}\end{aligned}$$

Define the algorithm $A(G)$ for Modularity that given G, constructs G', runs our subroutine for solving TellTailProb on G', and outputs $A'(G')$. We claim that $A(G)$ correctly solves Modularity. To show this, we consider two cases:

- **case 1:** there exists S such that $\tilde{e}_G(S) > 0$. Then since $\tilde{e}_G(S)$ is a fraction with an integer in the numerator and a denominator of $4m$, we have $\tilde{e}_G(S) \geq 1/(4m)$. Then, recalling that $\tilde{e}_G(S) = \tilde{e}_{G'}(S)$,

$$\begin{aligned}\textsc{TellTail}_{G'}(S) &= \frac{\tilde{e}_G(S) - \mu_0|S|^\alpha}{|S|^\beta} \\ &\geq \frac{\frac{1}{4m} - \mu_0|S|^\alpha}{|S|^\beta} \\ &\geq \frac{\frac{1}{4m} - \frac{1}{4mn^\alpha}|S|^\alpha}{|S|^\beta} \\ &> 0\end{aligned}$$

  Thus $A(G)$ outputs the correct result in this case: $A'(G')$ will return 'true' since $\textsc{TellTail}_{G'}(S) > 0$, so $A(G)$ will return 'true,' which is correct since $\tilde{e}_G(S) > 0$.

- **case 2:** there does not exist such an S; i.e. $\tilde{e}_G(S) \leq 0$ for all S. Then for all S,

$$\textsc{TellTail}_{G'}(S) = \frac{\tilde{e}_G(S) - \mu_0|S|^\alpha}{|S|^\beta} \leq \frac{\tilde{e}_G(S)}{|S|^\beta} \leq 0$$

  Thus $A(G)$ returns 'false', which is the correct result in this case as well.

In conclusion, $A(G)$ is a correct, polynomial time algorithm for Modularity, assuming we have a subroutine A' that solves TellTailProb. Since Modularity is NP-hard by [2], this implies that TellTailProb is NP-hard as well. Since we also showed that TellTailProb is in NP, this implies that it is NP-complete.

6 Subgraph Mass Distribution Plots in Real Data

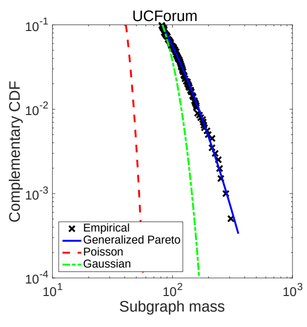
In this Section, we plot the empirical distribution of subgraph masses for all 8 of our original datasets as plotted in Table III, and all 5 of the Twitter subset graphs.

References

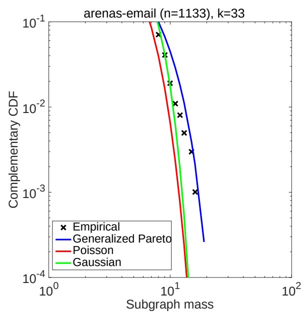
[1] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On finding graph clusterings with maximum modularity. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 121–132. Springer, 2007.

[2] T. N. Dinh, X. Li, and M. T. Thai. Network clustering via maximizing modularity: Approximation algorithms and theoretical limits. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 101–110. IEEE, 2015.

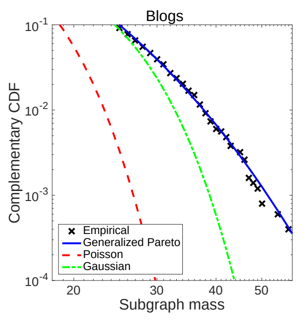
[3] R. L. Smith. Estimating tails of probability distributions. *The annals of Statistics*, pages 1174–1207, 1987.



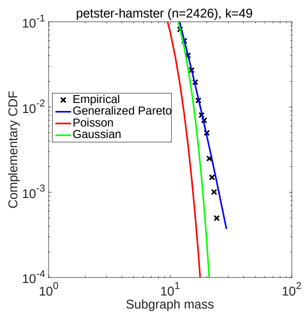
(a) `UCForum` ($n = 899$)



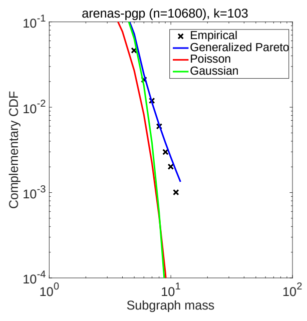
(b) `Email` ($n = 1.1$K)



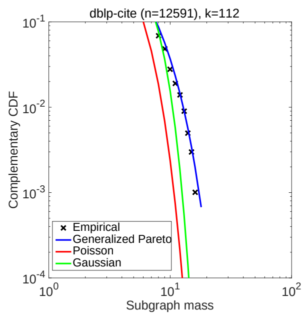
(c) `Blogs` ($n = 1.2$K)



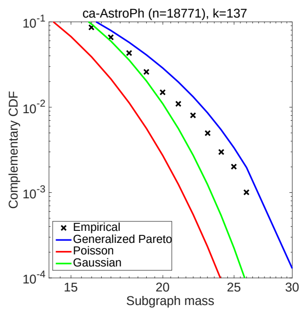
(d) `Petster` ($n = 2.4$K)



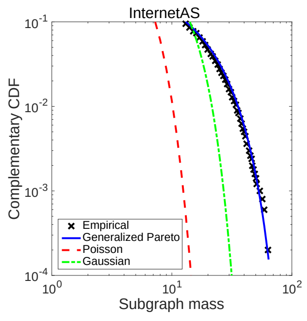
(e) `PGP` ($n = 11$K)



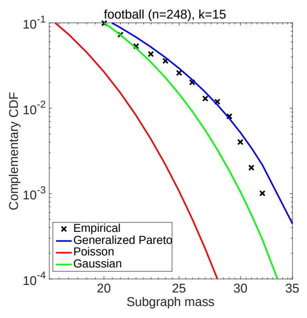
(f) `DBLP` ($n = 13$K)



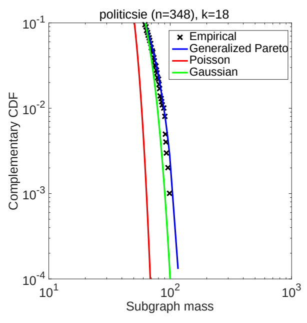
(g) `AstroPh` ($n = 18$K)



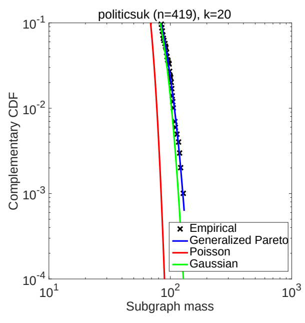
(h) `InternetAS` ($n = 34$K)



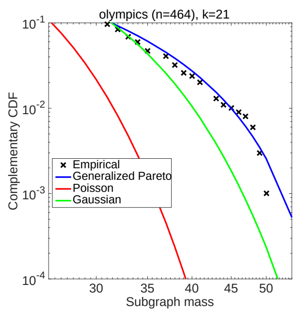
(i) `Football` ($n = 248$)



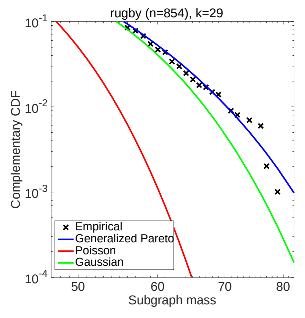
(j) `PoliticsIE` ($n = 348$)



(k) `PoliticsUK` ($n = 419$)



(l) `Olympics` ($n = 464$)



(m) `Rugby` ($n = 854$)

Figure 1: *The GP distribution fits the empirical distribution of subgraph masses much more closely than other distributions.* 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. The colored curves are the best fit GP, Poisson, and Gaussian to the empirical distribution. The Poisson curve is far to the left of the empirical distributions because Poisson distributions greatly underestimate the number of dense subgraphs that we should observe.