Bounding Procedures for k-clique Enumeration in Large Social Networks

Harry Xi [1] Supervised By: Lijun Chang[2]

[1]University of Melbourne [2]University of Sydney

Introduction

Clique (complete subgraph) enumeration is a fundamental problem within network science, and within the realm of social networks, represent *communities* of users. Unfortunately, as k increases, the number of k-cliques suffer from *combinatorial* explosion, which have been overcome by a number of *approximation* algorithms [4] [8] and *exact* algorithms [2].

Here, we present procedures for bounding the k-clique counts for fixed k, with a number of heuristics and even theoretical results.

Keywords: Social Networks, Algorithms, Cliques, Combinatorics, Heuristics

Shadows and the Kruskal-Katona Theorem [5]

Kruskal considered problems of the form: In a graph G, with $1,000,000$ edges (K_2 subgraphs), what is the maximum no. of triangles (K_3 subgraphs) in G? Observe: cliques are essentially combinations of vertices.

Definition: Given a t-combination, $\alpha = \{c_1, \ldots, c_t\}$, its shadow, $\partial\alpha$ is the set of $(t-1)$-subsets. For a set, A, of t-combinations,

$$\partial A = \bigcup\{\partial\alpha \,:\, \alpha \in A\}$$

Theorem (Kruskal-Katona): Given, A, a set of t-combinations with $|A| = N$ and $\partial^{t-i}A$ the set of $(t-i)$-element subsets satisfies,

$$\left|\partial^{t-i}A\right| \geq \binom{n_t}{t-i} + \binom{n_{t-1}}{t-i-1} + \ldots + \binom{n_j}{j}$$

Here, $N = \binom{n_t}{t} + \ldots + \binom{n_j}{j}$ is the *unique* binomial representation of N. In general, this can be found via a greedy algorithm by successively finding the closest lower bounding n_i and differencing.

This strange theorem solves our problem in the following way:

$$1,000,000 = \binom{1414}{2} + \binom{1009}{1} \quad \text{and} \quad 470,700,300 = \binom{1414}{3} + \binom{1009}{2}$$

So, for a family of 470,700,300 distinct 3-combinations (3-cliques) we have at least 1,000,000 edges. By incrementing the no. of triangles, $470,700,301 = \binom{1414}{3} + \binom{1009}{2} + \binom{1}{1}$ implying at least 1,000,001 edges. Thus, given 1,000,000 edges, the maximum no. of triangles is 470,700,300.

Coloured Complexes [1]

Let A be as above in a universe V. We say, A is $r-$coloured if there exists a *partition* of V into colour classes V_i's such that $\forall\alpha \in A$, $|\alpha \cap V_i| \leq 1$.

For *positive integers* $n,\, k,\, r$, with $n \geq k$, $r \geq k$, define the quantity,

$$\binom{n}{k}_r = \sum_{i=0}^{k} \binom{r_1}{i}\binom{r - r_1}{k-i}(a+1)^i a^{k-i}$$

With $a = \lfloor n/r \rfloor$ and $r_1 = n - ra$. This quantity arises from the size of the family,

$$\mathcal{H}(n,\, k,\, r) = \{S \,:\, |S| = k,\, |S \cap X_i| \leq 1,\, 1, \ldots, r\}$$

Where X_i are pairwise disjoint with $|X_i| = a+1$ for $1 \leq i \leq r_1$ and $|X_i| = a$ for $r_1 \leq i \leq r$. We prove via induction (with the above interpretation), that for fixed positive integer $r \leq k$ with $n > r$,

$$\binom{n+1}{k}_r > \binom{n}{k}_r$$

Thus, $\binom{n}{k}_r$ is strictly increasing in n.

A Kruskal-Katona type result has been established with this quantity.

Efficient Representation-Finding Algorithm

One can imagine, for fixed k and r, in finding a representation, the worst case scenario: $m = \binom{n}{k}_r$. By the Vandermonde identity, we can show that,

$$r\left[\frac{m}{\binom{r}{k}}\right]^{1/k} - r \leq n \leq r\left[\frac{m}{\binom{r}{k}}\right]^{1/k} + r \implies m^{1/k} - r < n < r(m^{1/k} + 1)$$

The second bound tells us for large $m \gg r$, the first bound should be a strict subset of the candidate interval $[0,\, m]$, thus giving a good starting interval for a greedy binary search procedure.

Upper Bound Procedure

We proceed via simple application of the Kruskal-Katona theorem and above representation-finding algorithm. Denote χ'_G the colouring attained by an $O(n+m)$ greedy colouring scheme.

Algorithm 1: Upper Bounding Procedure

Data: $G = (V,\, E)$, $k \leq 3$, $\;C(t)$ for $2 \leq t < k$
Compute the Degeneracy Ordering of G, denoted Π;
$\chi'_G \leftarrow$ greedyColour$(G,\, \Pi)$;
Initialise T, a zero-array of size $t+1$;
Compute the coloured complex representation of $C(t)$ such that $T[i] = n_{t-i}$;
return $\sum_{i=0}^{t} \binom{T[i]}{k-i}_{\chi'_G}$

Here, the *degeneracy ordering* refers to an ordering given in [7] (or its reverse), which is computed in linear $O(n+m)$ time, and is widely used in clique counting procedures. For a single instance of the representation-finding algorithm (running at most t times), the size of the search space is bounded above by $\chi'_G(C(t)^{1/k}+1) - C(t)^{1/t} + \chi'_G = (\chi'_G - 1)C(t)^{1/t} + 2\chi'_G$ so the time of finding the bound is,

$$O\left(n + m + t\log\left(\chi'_G C(t)^{1/k}\right)\right)$$

Empirically, this is far superior to a naive, $\binom{n}{k}$ bound, but for large k inferior to the DP solution in [8], depending on t, an initial known clique count.

Lower Bounding Procedure

Given a graph, G, to find a lower bound on the number of k-cliques, an easy heuristic is to find a single large t-clique. Then, an a lower bound is given by $\binom{t}{k}$. However, the word "easy" is disingenuous, as we hope to find the *maximum* clique, which is computationally hard. However, a necessary condition is that a maximum clique must be *maximal*, and we can find maximal cliques greedily.

This process can be improved: by finding multiple independent maximal cliques, and using the *degeneracy ordering*. In our implementation, the dense parts of G are located at the suffix, so the clique finding algorithm begins from the tail.

v0	v1	$\cdots$	"dense" suffix

With parameter, μ, we control the size of the suffix to length $(1-\mu)|V|$. Label the suffix R. With the ordering, we form a DAG such that the outneighbours of a vertex v_i is given by $N(v_i)^+ = N(v) \cap \{v_{i+1}, \ldots, v_n\}$. By considering $G[N(v_i)^+]$, the graph induced by the outneighbourhood with respect to the ordering, we may apply a greedy maximal clique search to find a clique of size t and since it exists within the outneighbourhood, we also include v_i, giving a $t+1$-clique.

Fixing v_i, the no. of independent k-cliques in that outneighbourhood is $\binom{t}{k-1}$. Repeat such a process for all vertices in the suffix and sum to obtain a lower bound on the number of k-cliques.

Sampling Application

In [8], a sampling procedure occurs on S, a set of vertices labelled "dense". The proportion ρ_p is determined empirically via sampling. The true proportion is given by the number of k-cliques within a set of k-colour paths.

However, as a result of the *Chernoff bound*, their approximation algorithm gives a $1-\epsilon$ approximation of the number of k-cliques in "dense" regions of G with probability $1 - 2\sigma$ if $t \geq \frac{3}{\rho_p \epsilon^2}\log\frac{1}{\sigma}$, which depends on the proportion itself. By lower bounding the number of k-cliques in the dense regions, we can find a theoretical bound for t, the number of required samples, removing the need for trial and error, at the cost of computational efficiency.

Experimental Results

As is standard for evaluating network analysis tools, a variety of large networks have been used from the Stanford Large Network Dataset Collection [6]. The number of triangles in a graph, $C(3)$ has been studied extensively, before larger k could be considered, thus, there exists a ground truth to many data sets for the number of triangles. We use this for our upper bound procedure, but note that due to new exact algorithms [2], our upper bounding procedure is able to perform better.

For all tests, $\mu = 0.99$, $k = 10$.

Graph	n	m	α	χ'_G	$L(k;\, \mu)$	$U(k; C(3))$	$C(k)^*$
`web-Stanford`	281,903	1,992,636	71	63	1.38E+11	1.93E+19	$\approx$ 5.82E+12
`com-lj`	4,036,538	34,681,189	360	327	4.43E+18	3.06E+23	$\approx$ 1.47E+19
`com-orkut`	3,072,627	117,185,083	253	92	5.05E+06	1.54E+25	$\approx$ 3.03E+13

Table 1. Results for a number of real-life networks

*Clique counts are given by estimates in [3].

We observe `com-orkut` behaves particularly bad, as it has many edges, but relatively low degeneracy, meaning the suffix is not overwhelmingly dense.

References

[1] PETER FRANKL, ZOLTÁN FÜREDI, and GIL KALAI. Shadows of colored complexes. *Mathematica Scandinavica*, 63(2):169–178, 1988.

[2] Shweta Jain and C. Seshadhri. The power of pivoting for exact clique counting. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 268–276. Association for Computing Machinery, 2020.

[3] Shweta Jain and C. Seshadhri. Provably and efficiently approximating near-cliques using the turán shadow: Peanuts. In *Proceedings of The Web Conference 2020*, WWW '20, page 1966–1976. Association for Computing Machinery, 2020.

[4] Shweta Jain and Hanghang Tong. *YACC: A Framework Generalizing T<sc>urán</sc>S<sc>hadow</sc> for Counting Large Cliques*, pages 684–692.

[5] Donald Ervin Knuth. *The art of computer programming, Volume 4A: Combinatorial Algorithms, part 1*. Addison-Wesley, 2011.

[6] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[7] David W. Matula and Leland L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *J. ACM*, 30(3):417–427, jul 1983.

[8] Xiaowei Ye, Rong-Hua Li, Qiangqiang Dai, Hongzhi Chen, and Guoren Wang. Lightning fast and space efficient k-clique counting. WWW '22, page 1191–1202. Association for Computing Machinery, 2022.

hzxi@student.unimelb.edu.au