Estimating the clustering coefficient using sample complexity analysis

Alane M. de Lima[1], Murilo V. G. da Silva[1], and André L. Vignatti[1]

Department of Computer Science, Federal University of Paraná, Curitiba – Brazil
{amlima,murilo,vignatti}@inf.ufpr.br

Abstract. In this work we present a sampling algorithm for estimating the local clustering of every vertex of a graph. Let G be a graph with n vertices, m edges, and maximum degree Δ. We present an algorithm that, given G and fixed constants $0 < \varepsilon, \delta, p < 1$, outputs the values for the local clustering coefficient within ε error with probability $1 - \delta$, for every vertex v of G, provided that the (exact) local clustering of v is not "too small". We use VC-dimension theory to give a bound for number of edges required to be sampled by the algorithm. We show that the algorithm runs in time $\mathcal{O}(\Delta \lg \Delta + m)$. We also show that the running time drops to, possibly, sublinear time if we restrict G to belong to some well known graph classes. In particular, for planar graphs the algorithm runs in time $\mathcal{O}(\Delta)$. In the case of bounded degree graphs the running time is $\mathcal{O}(1)$ if a bound for the value of Δ is given as a part of the input, and $\mathcal{O}(n)$ otherwise.

Keywords: clustering coefficient · approximation algorithm · sampling · VC-dimension.

1 Introduction

The occurrence of *clusters* in networks is a central field of investigation in the area of network theory [8, 24]. The existence of such phenomena motivated the creation of a variety of measures in order to quantify its prevalence, being the *clustering coefficient* [5, 2, 19] one of the most popular of these measures.

There are global and local versions of the clustering coefficient. In the *global clustering coefficient*, given a graph, the measure is a value that reflects the degree of clustering of the entire graph. In this case, the measure quantifies how close such graph is to being complete. However, if the objective is to analyze features of complex networks such as modularity, community structure, assortativity, and hierarchical structure, then the concept of *local clustering coefficient* is a better fit. This measure quantifies the degree in which a vertex is a part of a cluster in a graph. Simply speaking, the measure is related to the ratio of the number of triangles existing in the neighborhood of the target vertex to the total number of pair of nodes in the neighborhood. A precise definition for this measure is provided in Definition 1.

An exact algorithm for computing the local clustering coefficient for every vertex in a graph typically runs in cubic time. However, when dealing with large scale graphs, this is inefficient in practice, and high-quality approximations obtained with high confidence are usually sufficient. More specifically, given an accuracy or error parameter ε, a confidence parameter δ, and an adjustable lower bound parameter p, the idea is to sample a subset of edges in the graph that, for any fixed constants $0 < \varepsilon, \delta, p < 1$, the values for the local clustering coefficient can be estimated within ε error from the exact value with probability $1 - \delta$, for each vertex that respects a certain function of the parameter p. The details of the quality guarantees of the algorithm are better described in Section 2.

1.1 Related Works

The clustering coefficient was originally proposed by Watts and Strogatz (1998) [24] in order to determine if a graph have the property of being *small-world*. Intuitively, this coefficient is a measure of how close the neighborhood of a vertex is to being a clique. Over the years, many variants of this measure have being proposed in a way that it is somewhat difficult to provide an unified comparison between all these approaches under the light of algorithmic complexity.

One of these variations is the study of Soffer and Vázquez (2005) [23] on the influence of the degree of a vertex on the local clustering computation, proposing a modification on the original measure where the *degree-correlation* is filtered out. The work of Li et al. (2018) [14] provide a measure combining the local clustering coefficient and the local-degree sum of a vertex, but focused on the specific application of influence spreading. Other extensions of the measure and their applications in particular scenarios include link prediction [26, 7] and community detection [7]. In the theoretical front, working on random graphs, Kartun-Gilles and Bianconi (2019) [10] gives a statistical analysis of the topology of nodes in networks from different application scenarios. There are also many recent bounds for the average clustering of *power-law* graphs [12, 3, 9, 6], a graph model that represent many social and natural phenomena.

The algorithmic complexity of the exact computation of the local clustering coefficient for every vertex of a graph typically runs in cubic time. In our work, however, we are interested in faster approximation algorithms for obtaining good quality estimations. Let $n = |V|$ and $m = |E|$ in a graph G. In the work of Kutzkov and Pagh (2013) [13], the authors show an ε-approximation streaming algorithm for the local clustering coefficient of each vertex of degree at least d in expected time $\mathcal{O}(\frac{m}{\alpha\varepsilon^2} \log \frac{1}{\varepsilon} \log \frac{n}{\delta})$, where α is the local clustering coefficient of such vertex. This holds with probability at least $1 - \delta$. In the work of Zhang et al. (2017) [27], the authors propose an ε-approximation *MapReduce* based algorithm for the problem, and empirically compared its performance with other approximation algorithms designed using this type of approach [11, 21].

Results for the computation of the top-k vertices with the highest local clustering coefficient were also proposed [28, 15, 4]. In particular, Zhang et al. (2015) [28] use VC-dimension theory on their algorithm analysis, but in a different range

space than the one that we are dealing here, and for a scenario which is not exactly the one that we are tackling.

1.2 Our Results

In this paper we present an algorithm that samples edges from an input graph G and, for fixed constants $0 < \varepsilon, \delta, p < 1$, outputs an estimate $\tilde{l}(v)$ for the exact value $l(v)$ of the local clustering coefficient for each vertex $v \in V$, such that $|l(v) - \tilde{l}(v)| \leq \varepsilon l(v)$, with probability at least $1-\delta$ whenever $l(v)$ is at least $pm/\binom{\delta_v}{2}$, where δ_v is the degree of v. The main theme in our work is that, by using Vapnik–Chervonenkis (VC) theory, we can obtain an upper bound for sample size that is tighter than the ones given by standard Hoeffding and union-bound sampling techniques. In particular, we show that the sample size does not depend of the size of the G, but on a specific property of it, more precisely, its maximum degree Δ.

In Section 3.1 we give a definition for the VC-dimension of a graph and show in Theorem 2 that, for any graph, the VC-dimension is at most $\lfloor \lg(\Delta - 1) \rfloor + 1$. The sample size used in the algorithm depends, roughly speaking, on this value. In Corollary 1, we show that our analysis is tight by presenting an explicit construction of a class of graphs for which the VC-dimension reaches this upper bound. Even so, we also provide a tighter analysis for the case in which the input graph belongs to certain graph classes. In the class of *bounded degree graphs* the VC-dimension is bounded by a constant. In the case of *planar graphs*, we show, in Corollary 2, that the VC-dimension is at most 2.

In Section 3.2, we show that the running time for the general case of our algorithm is $\mathcal{O}(\Delta \lg \Delta + m)$. In Corollaries 3 and 4 we present an analysis for planar graphs and for bounded degree graphs, cases where the running time drops to, possibly, sublinear time. In the case of planar graphs, the Algorithm 1 has running time $\mathcal{O}(\Delta)$. In the case of bounded degree graphs the running time is $\mathcal{O}(1)$ if a bound for the value of Δ is given as a part of the input, and $\mathcal{O}(n)$ otherwise..

2 Preliminaries

In this section, we present the definitions, notation, and results that are the groundwork of our proposed algorithms. In all results of this paper, we assume w.l.o.g. that the input graph is connected, since otherwise the algorithm can be applied separately to each of its connected components.

2.1 Graphs and Local Clustering Coefficient

Let $G = (V, E)$ be a graph where V is the set of vertices and E the set of edges. We use convention that $n = |V|$ and $m = |E|$. For each vertex $v \in V$, let δ_v be the degree of v, and $\Delta_G = \max_{v \in V}\{\delta_v\}$ the maximum degree of the graph G. When the context is clear, we simply use Δ instead of Δ_G. We refer to a triangle

as being a complete graph with three vertices. Given $v \in V$, we let T_v be the number of triangles that contains v.

Definition 1. *(Local Clustering Coefficient) Given a graph $G = (V, E)$, the* local clustering coefficient *of a vertex $v \in V$ is*

$$l(v) = \frac{2T_v}{\delta_v(\delta_v - 1)}.$$

2.2 Sample Complexity and VC-dimension

In sampling algorithms, we typically want to estimate a certain quantity observing some parameters of quality and confidence. The sample complexity analysis relates the minimum size of a random sample required to obtain results that are consistent with the desired parameters. An upper bound to the Vapnik–Chervonenkis Dimension (VC-dimension) of a class of binary functions, a central concept in sample complexity theory, is especially defined in order to model the particular problem that we are dealing. An upper bound to the VC-dimension is also an upper bound to the sample size that respects the desired quality and confidence parameters.

Generally speaking, the VC-dimension measures the expressiveness of a class of subsets defined on a set of points [20]. An in-depth exposition of the definitions and results presented below can be found in the books of Anthony and Bartlett (2009) [1], Mohri *et al.* (2012) [18], Shalev-Shwartz and Ben-David (2014) [22], and Mitzenmacher and Upfal (2017) [17].

Definition 2 (Range space). *A* range space *is a pair $\mathcal{R} = (U, \mathcal{I})$, where U is a domain (finite or infinite) and $\mathcal{I}$ is a collection of subsets of U, called* ranges.

For a given $S \subseteq U$, the *projection* of $\mathcal{I}$ on S is the set $\mathcal{I}_S = \{S \cap I : I \in \mathcal{I}\}$. If $|\mathcal{I}_S| = 2^{|S|}$ then we say S is *shattered* by $\mathcal{I}$. The VC-dimension of a range space is the size of the largest subset S that can be shattered by $\mathcal{I}$, i.e.,

Definition 3 (VC-dimension). *The VC-dimension of a range space $\mathcal{R} = (U, \mathcal{I})$, denoted by VCDim($\mathcal{R}$), is*

$$VCDim(\mathcal{R}) = \max\{d : \exists S \subseteq U \text{ such that } |S| = d \text{ and } |\mathcal{I}_S| = 2^d\}.$$

The following combinatorial object, called a *relative (p, ε)-approximation*, is useful in the context when one wants to find a sample $S \subseteq U$ that estimates the size of ranges in $\mathcal{I}$, with respect to an adjustable parameter p, within $\varepsilon \Pr_\pi(I)$, for $0 < \varepsilon, p, \delta < 1$. This holds with probability at least $1 - \delta$, for $0 < \delta < 1$, where π is a distribution on U and $\Pr_\pi(I)$ is the probability of a sample from π belongs to I.

Definition 4 (relative (p, ε)-approximation, see [20], Def. 5). *Given $0 < p, \varepsilon < 1$, a set S is called a (p, ε)-approximation w.r.t. a range space $\mathcal{R} = (U, \mathcal{I})$, and a distribution π on U if for all $I \in \mathcal{I}$,*

(i) $\left|\Pr_\pi(I) - \frac{|S\cap I|}{|S|}\right| \leq \varepsilon \Pr_\pi(I), \quad$ *if* $\Pr_\pi(I) \geq p$,
(ii) $\frac{|S\cap I|}{|S|} \leq (1+\varepsilon)p$, *otherwise.*

An upper bound to the VC-dimension of a range space allows to build a sample S that is a (p,ε)-approximation set.

Theorem 1 (see [16], Theorem 5). *Given $0 < \varepsilon, \delta, p < 1$, let $\mathcal{R} = (U, \mathcal{I})$ be a range space with VCDim$(\mathcal{R}) \leq d$, a given distribution π on U, and let c be a universal positive constant. A collection of elements $S \subseteq U$ sampled w.r.t. π with*

$$|S| \geq \frac{c'}{\varepsilon^2 p}\left(d \log \frac{1}{p} + \log \frac{1}{\delta}\right)$$

is a relative (p,ε)-approximation with probability at least $1-\delta$, where c' is an absolute positive constant.

3 Estimation for the Local Clustering Coefficient

We first define the range space associated to the local clustering coefficient of a graph G and its corresponding VC-dimension, and then we describe the proposed approximation algorithm.

3.1 Range Space and VC-Dimension Results

Let $G = (V, E)$ be a graph. The range space $\mathcal{R} = (U, \mathcal{I})$ associated with G is defined as follows. The universe U is defined to be the set of edges E. We define a range τ_v, for every $v \in V$, as $\tau_v = \{e \in E :$ both endpoints of e are neighbors of v in $G\}$, and the range set corresponds to $\mathcal{I} = \{\tau_v : v \in V\}$. For the sake of simplicity, we often use VCDim(G) (instead of VCDim$(\mathcal{R})$) to denote the VC-dimension of the range space $\mathcal{R}$ associated with G.

Theorem 2 shows an upper bound forVCDim(G).

Theorem 2. *VCDim$(G) \leq \lfloor \lg(\Delta - 1) \rfloor + 1$.*

Proof. By definition, an edge $e \in E$ belongs to a range τ_v if both endpoints of e, say, a and b, are neighbors of v. That is, the number of ranges that contain e corresponds to the common neighbors of a and b. Let N be the set of such common neighbors. The maximum number of common neighbors a pair of vertices may have is Δ. Therefore, e is contained in at most $\Delta - 1$ ranges. Assuming that VCDim$(\mathcal{R}) = d$, then from Definition 3, the edge e must appear in 2^{d-1} ranges. We have

$$2^{d-1} \leq \Delta - 1 \Longrightarrow d - 1 \leq \lg(\Delta - 1) \Longrightarrow d \leq \lfloor \lg(\Delta - 1) \rfloor + 1.$$

□

One may ask if the bound given in Theorem 2 is tight. We now present an explicit construction of a family of graphs $\mathcal{G} = (G_d)_{d\geq 3}$ in order to show that this bound is tight with relation to Δ. A graph G_d, $d \geq 3$, of this family is constructed as follows. Initially, we create d disjoint edges $e_1, \ldots, e_d$. The endpoints of these edges are called *non-indexed vertices*. Let $1 \leq i_1 < i_2 < \ldots < i_k \leq d$. For every subset of k edges $e_{i_1}, \ldots, e_{i_k}$ we create a vertex $v_{(i_1,i_2,\ldots,i_k)}$ and connect it to both endpoints of each edge in the subset. These vertices are called *indexed vertices*. Figure 1 illustrates G_3 and G_4.

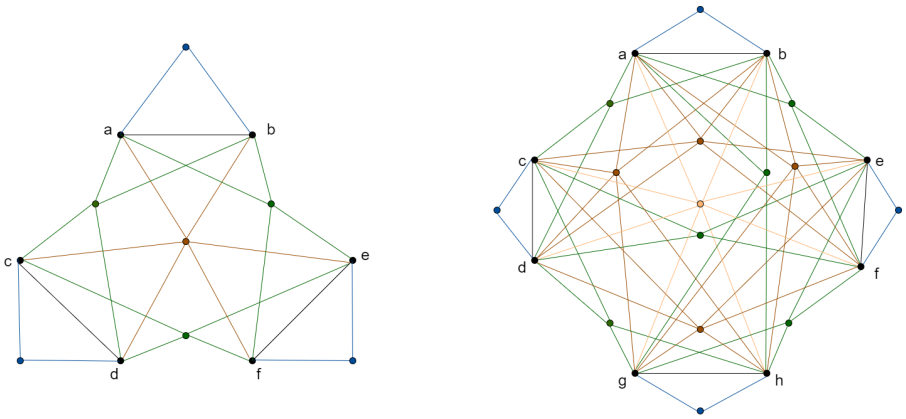

Fig. 1. The first two graphs of the construction of the family $\mathcal{G}$. In the case of G_3 (left), the edges of S are $e_1 = \{a, b\}$, $e_2 = \{c, d\}$, and $e_3 = \{e, f\}$. In the case of G_4 (right), the edges of S are $e_1 = \{a, b\}$. $e_2 = \{c, d\}$, $e_3 = \{e, f\}$, and $e_4 = \{(g, h\}$. Non-indexed vertices are labeled and depicted in black. We depict the indexed vertices in different colors, depending on the size of its neighborhood in S.

Claim. $\Delta_{G_d} = 2^{d-1} + 1$.

Proof. A vertex v in a graph $\mathcal{G}$ can be either indexed or non-indexed. We analyze each case separately.

Let v be a non-indexed vertex that is an endpoint of an edge e_j. W.l.o.g., we may assume that $j = 1$. The vertex v is adjacent to every indexed vertex with indices of the form $(1, i_1, \ldots, i_k)$. The first index is fixed, so there are 2^{d-1} indices of this form. So v is adjacent to 2^{d-1} indexed vertices. Also, v is adjacent to the other endpoint of e_1. Therefore, the degree of any non-indexed vertex is $2^{d-1} + 1$.

The degree of an indexed vertex cannot be larger than $2d$, since such vertex is adjacent to, at most, both endpoints of every edge $e_1, ..., e_d$. Since $2^{d-1}+1 \geq 2d$, the result follows. □

Theorem 3. *For every $d \geq 3$, $VCDim(G_d) \geq \lfloor \lg(\Delta_{G_d} - 1) \rfloor + 1$.*

Proof. Remember, $\mathcal{R} = (U, \mathcal{I})$, where U is the set of edges E, and $\mathcal{I} = \{\tau_v : v \in V\}$ where $\tau_v = \{e \in E :$ the endpoints of e are neighbors of v in $G\}$. First, we present a sample $S \subseteq U$, $|S| = d$ which is shattered, i.e., $|\mathcal{I}_S| = 2^d$, concluding that the VC-Dimension is at least d. After that, we show that $d = \lfloor \log(\Delta_{G_d} - 1) \rfloor + 1$, which proves the theorem.

Let $S = \{e_1, \ldots, e_d\}$. Consider an indexed vertex $v' = v_{(i_1, i_2, \ldots, i_k)}$. By the construction of the graph, we have that $S \cap \tau_{v'} = \{e_{i_1}, \ldots, e_{i_k}\}$, for all $\tau_{v'}$. That is, there is a one-to-one mapping of each v' to each $S \cap \tau_{v'}$. Since there are $2^d - 1$ indexed vertices v' (there is an indexed vertex for every subset except for the empty set), then there are $2^d - 1$ different intersections. Finally, the intersection that generates the empty set can be obtained by $S \cap \tau_{v''}$, where v'' is any non-indexed vertex. In other words,

$$|\{S \cap \tau_v | \tau_v \in \mathcal{I}\}| = |\mathcal{I}_S| = 2^d,$$

i.e., $\text{VCDim}(G_d) \geq d$. Now, using Claim 3.1, we have that

$$\lfloor \log(\Delta_{G_d} - 1) \rfloor + 1 = \lfloor \log(2^{d-1} + 1 - 1) \rfloor + 1 = \lfloor d - 1 \rfloor + 1 = d.$$

□

Combining Theorems 2 and 3, we conclude that the VC-Dimension of the range space is tight, as stated by Corollary 1.

Corollary 1. *For every $d \geq 3$, there is a graph G such that VCDim$(G) = d = \lfloor \lg(\Delta - 1) \rfloor + 1$.*

Next we define a more general property that holds for a graph G_d.

Property (*) We say that a graph $G = (V, E)$ has the *Property (*)* if exists $S \subseteq E$, $|S| \geq 3$, such that:

(i) For each $e = \{u, v\} \in S$, $|e \cap \{S \setminus \{e\}\}| \leq 1$.
(ii) For each subset $S' \subseteq S$, there is at least one vertex $v_{S'}$ that is adjacent to both endpoints of each edge of S'.

For every $d \geq 3$, Theorem 4 gives conditions based on Property (*) that a graph must obey in order to have VC-dimension at least d.

Theorem 4. *Let G be a graph. If VCdim$(G) \geq 3$, then G has Property (*).*

Proof. We prove the contrapositive of the statement, i.e., we show that if G does not have Property (*), then $\text{VCdim}(G) < 3$. Note that if we assume that G does not have Property (*), then for all $S \subseteq E, |S| \geq 3$, we have that either condition (i) or condition (ii) is false.

If it is the case that (ii) is false, then for all $S \subseteq E, |S| \geq 3$, there is a set $S' \subseteq S$ such that there is no $v_{S'} \in V$ which is adjacent to both endpoints of each edge in S'. We have that the number of subsets of S is $2^{|S|}$, so G must have at least $2^{|S|}$ vertices so that $\mathcal{I}_S = 2^{|S|}$. From the definition of shattering, if

$\mathcal{I}_S < 2^{|S|}$, then it is not possible that $\text{VCdim}(G) \geq |S|$. Since $|S| \geq 3$, it cannot be the case that $\text{VCdim}(G) \geq 3$.

Now consider the case where (i) is false. In this case, for all $S \subseteq E, |S| \geq 3$, there is an edge $e = \{u, v\} \in S$ where both u and v are endpoints of other edges in S (i.e., $|e \cap \{S \setminus \{e\}\}| = 2$). We name such edge $e_2 = \{b, c\}$. Suppose w.l.o.g. that e_2 share its endpoints with the edges $e_1 = \{a, b\}$ and $e_3 = \{c, d\}$. Then every triangle containing e_1 and e_3 necessarily contains e_2. Denote by z the vertex which forms triangles with e_1 and e_2. Then z also forms a triangle with e_2, since it is adjacent to both b and c, which are the endpoints of e_2. Hence, the subset $\{e_1, e_3\}$ cannot be generated from the intersection of $\mathcal{I}$ with e_1, e_2, and e_3. Therefore it cannot be the case that $\text{VCdim}(G) \geq 3$. □

Although Theorem 2 gives a tight bound for the VC-dimension, if we have more information about the type of graph that we are working, we can prove better results. In Corollary 2, we show that if G is a graph from the class of planar graphs, then the VC-dimension of G is at most 2. Another very common class of graphs where we can achieve a constant bound for the VC-dimension is the class of *bounded degree graphs*, i.e., graphs where Δ is bounded by a constant. For this class, the upper bound comes immediately from Theorem 2.

Note that, even though planar graphs and bounded degree graphs are both classes of sparse graphs, such improved bounds for the VC-dimension for these classes do not come directly from the sparsity of these graphs, since we can construct a (somewhat arbitrary) class of sparse graphs $\mathcal{G}'$ where the VC-dimension is as high as the one given by Theorem 2. The idea is that $\mathcal{G}' = (G'_d)_{d \geq 3}$, where each graph G'_d is the union of G_d with a sufficiently large sparse graph. In the other direction, one should note that dense graphs can have small VC-dimension as well, since complete graphs have VC-dimension at most 2. This comes from the fact that complete graphs do not have the Property (*). In fact, for a K_q, $q \geq 4$, the VC-dimension is exactly 2, since any set of two edges that have one endpoint in common can be shattered in this graph.

Corollary 2. *If G is a planar graph, then VCDim$(G) \leq 2$.*

Proof. We prove the VC-dimension of the range space of a planar graph is at most 2 by demonstrating the contrapositive statement. More precisely, from Theorem 4, we have that if $\text{VCDim}(G) \geq 3$, then G has *Property (*)*. In this case we show that G must contain a subdivision of a $K_{3,3}$, concluding that G cannot be planar, according to the Theorem of Kuratowski [25].

From Theorem 4, G has a subset of edges $\{e_1, e_2, e_3\}$ respecting conditions (i) and (ii) of *Property (*)*. Let $e_1 = \{a, b\}$, $e_2 = \{c, d\}$, and $e_3 = \{e, f\}$. Note that these three edges may have endpoints in common. By *condition (i)*, we may assume w.l.o.g. that $a \neq c \neq e$. By symmetry, w.l.o.g., there are three possibilities for the vertices b, d, and f: (1) they are all distinct vertices, (2) we have $d = f$, but $b \neq f$, and (3) they are the same vertex, i.e., $b = d = f$. In Figure 2 we show three graphs, one for each of these three possible configuration for the arrangement of edges e_1, e_2, e_3. By *condition (ii)* there are at least four vertices, say, u, v, w, x respecting the following:

– u is adjacent to all vertices of $\{a, b, c, d, e, f\}$;
– v is adjacent to all vertices of $\{a, b, c, d\}$ and not adjacent to both e and f;
– w is adjacent to all vertices of $\{a, b, e, f\}$ and not adjacent to both c and d;
– x is adjacent to all vertices of $\{c, d, e, f\}$ and not adjacent to both a and b.

Note that, even though every edge depicted in 2 is mandatory in G, there may be other edges in G that are not shown in the picture.

Since all of $\{v, c\}, \{c, x\}$ and $\{x, e\}$ are edges in G, then there is a path from v to e in G. Let $E(P)$ be the edges of this path. Let X be the set of edges with one endpoint in A and one endpoint in B. We can obtain a subgraph H of G that contains a subdivision of a bipartite graph $K_{3,3}$ with bipartition (A, B) in the following way. Let $A = \{a, b, e\}$ and $B = \{u, v, w\}$. The vertex set of H is $A \cup B \cup \{c, x\}$ and the edge set of H is $X \cup E(P)$. Therefore G cannot be a planar graph. □

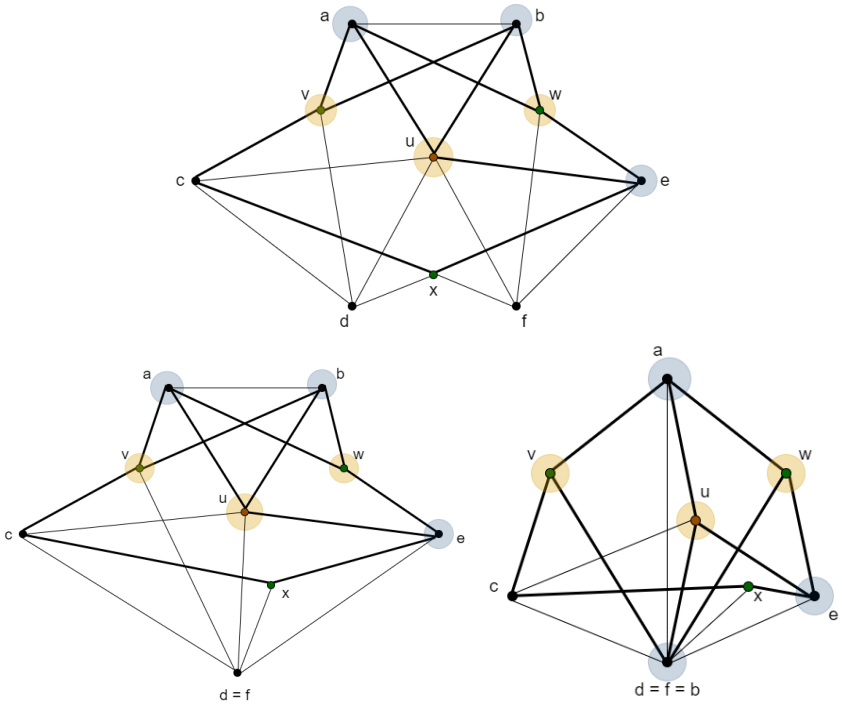

Fig. 2. Three possible arrangements for the edges $e_1 = \{a, b\}$, $e_2 = \{c, d\}$, and $e_3 = \{e, f\}$ from the proof of Corollary 2. The case where b, d, and f are distinct vertices is depicted above. The case where $d = f$, but $b \neq f$ is shown below in the left. Below in the right, we show the case where $b = d = f$. In all three cases, these edges are part of a subgraph H of G that contains a subdivision of a $K_{3,3}$.

3.2 Algorithm

The algorithm take as input a graph $G = (V, E)$, the quality and confidence parameters $0 < \varepsilon, \delta < 1$, and a parameter $0 < p < 1$, all parameters assumed to be constants. It outputs the estimation $\tilde{l}(v)$ for the exact value $l(v)$ of the local clustering coefficient for each vertex $v \in V$, such that

$$|l(v) - \tilde{l}(v)| \leq \varepsilon l(v), \text{ with prob. at least } 1 - \delta \text{ whenever } l(v) \geq \sigma_v(p),$$

where $\sigma_v(p) = pm/\binom{\delta_v}{2}$ is an adjustable function, depending on p. The idea, roughly speaking, is that if the neighborhood of v is not too small, then $l(v) \geq \sigma_v(p)$ holds.

Next we present Algorithm 1. At the beginning all $\tilde{T}_v$ are set to zero.

Algorithm 1: localClusteringEstimation(G,ε,δ,p)

input : Graph $G = (V, E)$ with m edges, parameters $0 < \varepsilon, \delta, p < 1$.
output: Local clustering coefficient estimation $\tilde{l}(v)$, $\forall v \in V$ s.t. $l(v) \geq \sigma_v(p)$.

```
r ← ⌈ (c'/(ε²p)) ( (⌊lg Δ − 1⌋ + 1) log(1/p) + log(1/δ) ) ⌉
for i ← 1 to r do
    sample an edge e = {a, b} ∈ E uniformly at random
    forall v ∈ N_a do
        if v ∈ N_b then
            T̃_v ← T̃_v + m/r
return l̃(v) ← 2T̃_v / (δ_v(δ_v − 1)), for each v ∈ V.
```

Theorem 5. *Let $S = \{e_1, \ldots, e_r\}$ be a sample of size*

$$r = \left\lceil \frac{c'}{\varepsilon^2 p} \left((\lfloor \lg \Delta - 1 \rfloor + 1) \log \frac{1}{p} + \log \frac{1}{\delta} \right) \right\rceil$$

for a given graph $G = (V, E)$ and for given $0 < p, \varepsilon, \delta < 1$. Algorithm 1 returns with probability at least $1 - \delta$ an approximation $\tilde{l}(v)$ to $l(v)$ within ε relative error, for each $v \in V$ such that $l(v) \geq \sigma_v(p)$.

Proof. For each $v \in V$, let $\mathbb{1}_v(e)$ be the function that returns 1 if $e \in \tau_v$ (and 0 otherwise). Thus, $T_v = \sum_{e \in E} \mathbb{1}_v(e)$. The estimate value $\tilde{T}_v$, computed by Algorithm 1, is incremented by m/r whenever an edge $e \in S$ belongs to τ_v, i.e.,

$$\tilde{T}_v = \sum_{e \in S} \frac{m}{r} \mathbb{1}_v(e).$$

Note that

$$\tilde{T}_v = \sum_{e \in S} \frac{m}{r} \mathbb{1}_v(e) = \frac{m}{r} \sum_{e \in S} \mathbb{1}_v(e) = m \cdot \frac{|S \cap \tau_v|}{|S|}.$$

Thus, assuming that we have a relative (p, ε)-approximation (Definition 4),

$$\frac{|T_v - \tilde{T}_v|}{T_v} = \frac{\left|m \cdot \Pr_\pi(\tau_v) - m \cdot \frac{|S \cap \tau_v|}{|S|}\right|}{m \cdot \Pr_\pi(\tau_v)} = \frac{\left|\Pr_\pi(\tau_v) - \frac{|S \cap \tau_v|}{|S|}\right|}{\Pr_\pi(\tau_v)} \leq \varepsilon.$$

Or, simply put, $|T_v - \tilde{T}_v| \leq \varepsilon T_v$. Therefore,

$$|l(v) - \tilde{l}(v)| = \frac{2|T_v - \tilde{T}_v|}{\delta_v(\delta_v - 1)} \leq \frac{2\varepsilon T_v}{\delta_v(\delta_v - 1)} = \varepsilon l(v).$$

Combining this with Theorems 1 and 2, and using a sample S with size

$$r = \left\lceil \frac{c'}{\varepsilon^2 p}\left((\lfloor \lg \Delta - 1 \rfloor + 1)\log\frac{1}{p} + \log\frac{1}{\delta}\right)\right\rceil,$$

we have that Algorithm 1 provides an ε-error estimation for $l(v)$ with probability $1 - \delta$ for all $v \in V$ s.t. $\Pr(\tau_v) \geq p$. But $\Pr(\tau_v) \geq p$ if and only if $l(v) \geq \sigma_v(p)$ since

$$l(v) = \frac{T_v}{\binom{\delta_v}{2}} = \frac{m \Pr(\tau_v)}{\binom{\delta_v}{2}}.$$

□

We remark that $\tilde{T}_v$ is an unbiased estimator for T_v, since

$$\mathbb{E}[\tilde{T}_v] = \mathbb{E}\left[\sum_{e \in S} \frac{m}{r} \mathbb{1}_v(e)\right] = \frac{m}{r} \sum_{e \in S} \Pr(e \in \tau_v) = \frac{m}{r} \sum_{e \in S} \frac{|\tau_v|}{m} = T_v.$$

Theorem 6. *Given a graph $G = (V, E)$ and a sample of size*

$$r = \left\lceil \frac{c'}{\varepsilon^2 p}\left((\lfloor \lg \Delta - 1 \rfloor + 1)\log\frac{1}{p} + \log\frac{1}{\delta}\right)\right\rceil,$$

Algorithm 1 has running time $\mathcal{O}(\Delta \lg \Delta + m)$.

Proof. In line 1, the value of Δ can be computed in time $\Theta(m)$. Given an edge $\{a, b\}$ we first store the neighbors of b in a directed address table. Then, lines 4, 5, and 6 take time $\mathcal{O}(\Delta)$ by checking, for each $v \in N_a$, if v is in the table. Hence, the total running time of Algorithm 1 is $\mathcal{O}(r \cdot \Delta + m) = \mathcal{O}(\Delta \lg \Delta + m)$. □

As mentioned before, for specific graph classes, the running time proved in 6 can be reduced. We can achieve this either by proving that graphs in such classes have a smaller VC-dimension, or by looking more carefully at the algorithm analysis for such classes. In Corollaries 3 and 4, we present results for two such classes.

Corollary 3. *If G is a planar graph, then Algorithm 1 has running time $\mathcal{O}(\Delta)$.*

Proof. By Corollary 2, $\text{VCDim}(G) \leq 2$. So, the sample size in the Algorithm 1 changes from a function of Δ to a constant. Note that, in particular, since we do not need to find the value of Δ, line 1 can be computed in time $\mathcal{O}(1)$. As with the proof of Theorem 6, lines 4, 5, and 6 still take time $\mathcal{O}(\Delta)$. Since r is constant, line 2 takes constant time. So, the total running time of Algorithm 1 is $\mathcal{O}(r \cdot \Delta) = \mathcal{O}(\Delta)$. □

Another case where we can provide a better running for the algorithm is the case for *bounded degree graphs*, i.e., the case where the maximum degree of any graph in the class is bounded by a constant.

Corollary 4. *Let G be a bounded degree graph, where d is such bound. Algorithm 1 has running time $\mathcal{O}(1)$ or $\mathcal{O}(n)$, respectively, depending on whether d is part of the input or not.*

Proof. If d is part of the input, then the number of samples r in line 1 can be computed in time $\mathcal{O}(1)$. Line 2 is executed $\mathcal{O}(1)$ times, and the remaining of the algorithm, in lines 4, 5, and 6, takes $\mathcal{O}(1)$ time, since the size of the neighborhood for every vertex is bounded by a constant.

On the other hand, if d is not part of the input, then Δ must be computed for the execution of line 1. In this case we check the degree of every vertex by traversing its adjacency list. All these adjacency lists have constant size. Performing this for all vertices takes time $\mathcal{O}(n)$. The other steps of the algorithm take constant time. □

4 Conclusion

We present a sampling algorithm for local clustering problem. In our analysis we define a range space associated to the input graph, and show how the sample size of the algorithm relates to the VC-dimension of this range space. This kind of analysis takes into consideration the combinatorial structure of the graph, so the size of the sample of edges used by the algorithm depends on the maximum degree of the input graph.

Our algorithm executes in time $\mathcal{O}(\Delta \lg \Delta + m)$ in the general case and guarantees, for given parameters ε, δ and p, that the approximation value has relative error ε with probability at least $1 - \delta$, for every node whose clustering coefficient is greater than a certain function adjusted by the parameter p. For planar graphs we show that the sample size can be bounded by a constant, an the running time in this case is $\mathcal{O}(\Delta)$. In the case of bounded degree graphs, where there is also a constant bound on the sample size, the running time drops to $\mathcal{O}(1)$ or $\mathcal{O}(n)$, depending on whether the bound on the degree is part of the input or not.

Bibliography

[1] Anthony, M., Bartlett, P.L.: Neural Network Learning: Theoretical Foundations. Cambridge University Press, New York, 1st edn. (2009)

[2] Barabási, A.L., Pósfai, M.: Network science. Cambridge University Press (2016)

[3] Bloznelis, M.: Degree and clustering coefficient in sparse random intersection graphs. The Annals of Applied Probability **23**(3), 1254–1289 (2013)

[4] Brautbar, M., Kearns, M.: Local algorithms for finding interesting individuals in large networks. In: ICS (2010)

[5] Easley, D.A., Kleinberg, J.M.: Networks, Crowds, and Markets - Reasoning About a Highly Connected World. Cambridge University Press (2010)

[6] Fronczak, A., Fronczak, P., Hołyst, J.A.: Mean-field theory for clustering coefficients in barabási-albert networks. Physical Review E **68**(4), 046126 (2003)

[7] Gupta, A.K., Sardana, N.: Significance of clustering coefficient over jaccard index. In: 2015 Eighth International Conference on Contemporary Computing (IC3). pp. 463–466. IEEE (2015)

[8] Holland, P.W., Leinhardt, S.: Transitivity in structural models of small groups. Comparative Group Studies **2**(2), 107–124 (1971)

[9] Iskhakov, L., Kamiński, B., Mironov, M., Prałat, P., Prokhorenkova, L.: Local clustering coefficient of spatial preferential attachment model. Journal of Complex Networks **8**(1), cnz019 (2020)

[10] Kartun-Giles, A.P., Bianconi, G.: Beyond the clustering coefficient: A topological analysis of node neighbourhoods in complex networks. Chaos, Solitons & Fractals: X **1**, 100004 (2019)

[11] Kolda, T.G., Pinar, A., Plantenga, T., Seshadhri, C., Task, C.: Counting triangles in massive graphs with mapreduce. SIAM Journal on Scientific Computing **36**(5), S48–S77 (2014)

[12] Krot, A., Ostroumova Prokhorenkova, L.: Local clustering coefficient in generalized preferential attachment models. In: International Workshop on Algorithms and Models for the Web-Graph. pp. 15–28. Springer (2015)

[13] Kutzkov, K., Pagh, R.: On the streaming complexity of computing local clustering coefficients. In: Proceedings of the sixth ACM international conference on Web search and data mining. pp. 677–686 (2013)

[14] Li, M., Zhang, R., Hu, R., Yang, F., Yao, Y., Yuan, Y.: Identifying and ranking influential spreaders in complex networks by combining a local-degree sum and the clustering coefficient. International Journal of Modern Physics B **32**(06), 1850118 (2018)

[15] Li, X., Chang, L., Zheng, K., Huang, Z., Zhou, X.: Ranking weighted clustering coefficient in large dynamic graphs. World Wide Web **20**(5), 855–883 (2017)

[16] Li, Y., Long, P.M., Srinivasan, A.: Improved bounds on the sample complexity of learning. Journal of Computer and System Sciences **62**(3), 516 – 527 (2001)
[17] Mitzenmacher, M., Upfal, E.: Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis. Cambridge University Press, New York, 2nd edn. (2017)
[18] Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. The MIT Press, Cambridge (2012)
[19] Newman, M.E.J.: Networks: an introduction. Oxford University Press (2010)
[20] Riondato, M., Kornaropoulos, E.M.: Fast approximation of betweenness centrality through sampling. Data Mining and Knowledge Discovery **30**(2), 438–475 (2016)
[21] Seshadhri, C., Pinar, A., Kolda, T.G.: Fast triangle counting through wedge sampling. In: Proceedings of the SIAM Conference on Data Mining. vol. 4, p. 5. Citeseer (2013)
[22] Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, New York (2014)
[23] Soffer, S.N., Vazquez, A.: Network clustering coefficient without degree-correlation biases. Physical Review E **71**(5), 057101 (2005)
[24] Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world'networks. nature **393**(6684), 440–442 (1998)
[25] West, D.B.: Introduction to Graph Theory. Prentice Hall, 2 edn. (September 2000)
[26] Wu, Z., Lin, Y., Wang, J., Gregory, S.: Link prediction with node clustering coefficient. Physica A: Statistical Mechanics and its Applications **452**, 1–8 (2016)
[27] Zhang, H., Zhu, Y., Qin, L., Cheng, H., Yu, J.X.: Efficient local clustering coefficient estimation in massive graphs. In: International Conference on Database Systems for Advanced Applications. pp. 371–386. Springer (2017)
[28] Zhang, J., Tang, J., Ma, C., Tong, H., Jing, Y., Li, J.: Panther: Fast top-k similarity search on large networks. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1445–1454 (2015)