E-CLoG: Counting Edge-Centric Local Graphlets. (Algorithms)

*Vachik S. Dave

1. IMPLEMENTED ALGORITHMS

Here in this section, we provide Algorithms 1-6, which are detailed algorithms (as implemented) to count frequency of each 5 sized local graphlets shown in Figure 1ii. These algorithms are generated using fixed template algorithms 10,11,12 and 13 with variable values mentioned in Table I. For quick understanding of these algorithms, we discuss notations first. (for more detail check our paper "E-CLoG: Counting Edge-Centric Local Graphlets" published at BigData-2017.)

1.1. Notations

T : Set of nodes creating triangles with edge (u, v).
N_u : Set of nodes only neighbor of u not v.
N_v : Set of nodes only neighbor of v not u.
Vertices u and v are not included in any of these sets, i.e. $u, v \notin (T \cup N_u \cup N_v)$, also notice that T, N_u and N_v are disjoint sets i.e.

$$N_u \cap T = \phi \qquad \& \qquad N_v \cap T = \phi \qquad \& \qquad N_u \cap N_v = \phi$$

We calculate frequency of specific graphlet types using above sets (T, N_u, N_v) and frequency of other graphlet types using below equations:

$$f2 \leftarrow |N_u| \times |N_v| - f5 \tag{1}$$

$$f4 \leftarrow \binom{|N_u|}{2} + \binom{|N_v|}{2} - f6 \tag{2}$$

$$f8 \leftarrow |T| \times f0 - f9 \tag{3}$$

$$f10 \leftarrow \binom{|T|}{2} - f11 \tag{4}$$

For any 5 size local graphlet, we select three nodes (i, j and k) other than u, v from $(T \cup N_u \cup N_v)$. A careful selection of these nodes $i, j, k \in (T \cup N_u \cup N_v)$, leads to a specific graphlet type.

$N_{Ti} = Neighbor(i) \cap T$ For any node $i \in (T \cup N_u \cup N_v)$
$N_{ui} = Neighbor(i) \cap N_u$ For any node $i \in (T \cup N_u \cup N_v)$
$N_{vi} = Neighbor(i) \cap N_v$ For any node $i \in (T \cup N_u \cup N_v)$

Algorithm 1: $i, j \in T$, $k \in T$, $k \in (N_u \cup N_v)$

for all $i \in T$ **do**
    for all $j \in N_{Ti}$ **do**
        // $k \in T$
        $N_{Tij} \leftarrow N_{Ti} \cap N_{Tj}$
        $f45_{ij} \leftarrow f45_{ij} + |N_{Tij}|$
        $f44_{ij} \leftarrow f44_{ij} + (|N_{Tj}| - |N_{Tij}| - 1)$ // 1 is deducted for node i
        $f39_{ij} \leftarrow f39_{ij} + (|T| - |N_{Ti}| - |N_{Tj}| + |N_{Tij}|)$
        // $k \in N_u$
        $N_{uij} \leftarrow N_{ui} \cap N_{uj}$
        $f31_{uij} \leftarrow f31_{uij} + (|N_u| - |N_{ui}| - |N_{uj}| + |N_{uij}|)$
        // $k \in N_v$
        $N_{vij} \leftarrow N_{vi} \cap N_{vj}$
        $f31_{vij} \leftarrow f31_{vij} + (|N_v| - |N_{vi}| - |N_{vj}| + |N_{vij}|)$
    end for
    $f45_i \leftarrow f45_i + f45_{ij}/2$
    $f44_i \leftarrow f44_i + f44_{ij}$
    $f39_i \leftarrow f39_i + f39_{ij}$
    $f31_i \leftarrow f31_i + f31_{uij} + f31_{vij}$
end for
$f45 \leftarrow f45 + f45_i/3$
$f44 \leftarrow f44 + f44_i/2$
$f39 \leftarrow f39 + f39_i/2$
$f31 \leftarrow f31 + f31_i/2$

Algorithm 2: $i \in T$, $j \in N_u$, $k \in T$

for all $i \in T$ **do**
  for all $j \in N_{ui}$ **do**
    $N_{Tij} \leftarrow N_{Ti} \cap N_{Tj}$
    $f43_{ij} \leftarrow f43_{ij} + |N_{Tij}|$
    $f42_{ij} \leftarrow f42_{ij} + (|N_{Tj}| - |N_{Tij}| - 1)$ // 1 is deducted for node i
    $f40_{ij} \leftarrow f40_{ij} + (|N_{Ti}| - |N_{Tij}|)$
    $f34_{ij} \leftarrow f34_{ij} + (|T| - |N_{Ti}| - |N_{Tj}| + |N_{Tij}|)$
  end for
  $f43_i \leftarrow f43_i + f43_{ij}$
  $f42_i \leftarrow f42_i + f42_{ij}$
  $f40_i \leftarrow f40_i + f40_{ij}$
  $f34_i \leftarrow f34_i + f34_{ij}$
end for
$f43 \leftarrow f43 + f43_i/2$
$f42 \leftarrow f42 + f42_i/2$
$f40 \leftarrow f40 + f40_i$
$f34 \leftarrow f34 + f34_i$

Algorithm 3: $i \in T$, $j \in N_u$, $k \in (N_u \cup N_v)$

for all $i \in T$ **do**
  for all $j \in N_{ui}$ **do**
    // $k \in N_u$
    $N_{uij} \leftarrow N_{ui} \cap N_{uj}$
    $f38_{ij} \leftarrow f38_{ij} + |N_{uij}|$
    $f33_{ij} \leftarrow f33_{ij} + (|N_{uj}| - |N_{uij}|)$
    $f29_{ij} \leftarrow f29_{ij} + (|N_{ui}| - |N_{uij}|)$
    $f22_{ij} \leftarrow f22_{ij} + (|N_u| - |N_{ui}| - |N_{uj}| + |N_{uij}|)$
    // $k \in N_v$
    $N_{vij} \leftarrow N_{vi} \cap N_{vj}$
    $f41_{ij} \leftarrow f41_{ij} + |N_{vij}|$ // to avoid duplicate counting, dont count for $j \in N_{vi}$
    $f36_{ij} \leftarrow f36_{ij} + (|N_{vj}| - |N_{vij}|)$
    $f35_{ij} \leftarrow f35_{ij} + (|N_{vi}| - |N_{vij}|)$ // to avoid duplicate counting, dont count for $j \in N_{vi}$
    $f25_{ij} \leftarrow f25_{ij} + (|N_v| - |N_{vi}| - |N_{vj}| + |N_{vij}|)$
  end for
  $f38_i \leftarrow f38_i + f38_{ij}/2$
  $f33_i \leftarrow f33_i + f33_{ij}$
  $f29_i \leftarrow f29_i + f29_{ij}/2$
  $f22_i \leftarrow f22_i + f22_{ij}$
  $f41_i \leftarrow f41_i + f41_{ij}$
  $f36_i \leftarrow f36_i + f36_{ij}$
  $f35_i \leftarrow f35_i + f35_{ij}$
  $f25_i \leftarrow f25_i + f25_{ij}$
end for
$f38 \leftarrow f38 + f38_i$
$f33 \leftarrow f33 + f33_i$
$f29 \leftarrow f29 + f29_i$
$f22 \leftarrow f22 + f22_i$
$f41 \leftarrow f41 + f41_i$
$f36 \leftarrow f36 + f36_i$
$f35 \leftarrow f35 + f35_i$
$f25 \leftarrow f25 + f25_i$

Algorithm 4: $i, j \in N_u$, $k \in (N_u \cup N_v \cup T)$

for all $i \in N_u$ **do**
  for all $j \in N_{ui}$ **do**
    // $k \in N_u$
    $N_{uij} \leftarrow N_{ui} \cap N_{uj}$
    $f32_{ij} \leftarrow f32_{ij} + |N_{uij}|$
    $f23_{ij} \leftarrow f23_{ij} + (|N_{uj}| - |N_{uij}| - 1)$ // deduct 1 for node i
    $f18_{ij} \leftarrow f18_{ij} + (|N_u| - |N_{ui}| - |N_{uj}| + |N_{uij}|)$
    # $k \in N_v$
    $N_{vij} \leftarrow N_{vi} \cap N_{vj}$
    $f37_{ij} \leftarrow f37_{ij} + |N_{vij}|$
    $f28_{ij} \leftarrow f28_{ij} + (|N_{vj}| - |N_{vij}|)$
    $f16_{ij} \leftarrow f16_{ij} + (|N_v| - |N_{vi}| - |N_{vj}| + |N_{vij}|)$
    # $k \in T$
    $N_{Tij} \leftarrow N_{Ti} \cap N_{Tj}$
    $f24_{ij} \leftarrow f24_{ij} + (|T| - |N_{Ti}| - |N_{Tj}| + |N_{Tij}|)$
  end for
  $f32_i \leftarrow f32_i + f32_{ij}/2$
  $f23_i \leftarrow f23_i + f23_{ij}$
  $f18_i \leftarrow f18_i + f18_{ij}$
  $f37_i \leftarrow f37_i + f37_{ij}$
  $f28_i \leftarrow f28_i + f28_{ij}$
  $f16_i \leftarrow f16_i + f16_{ij}$
  $f24_i \leftarrow f24_i + f24_{ij}$
end for
$f32 \leftarrow f32 + f32_i/3$
$f23 \leftarrow f23 + f23_i/2$
$f18 \leftarrow f18 + f18_i/2$
$f37 \leftarrow f37 + f37_i/2$
$f28 \leftarrow f28 + f28_i$
$f16 \leftarrow f16 + f16_i/2$
$f24 \leftarrow f24 + f24_i/2$

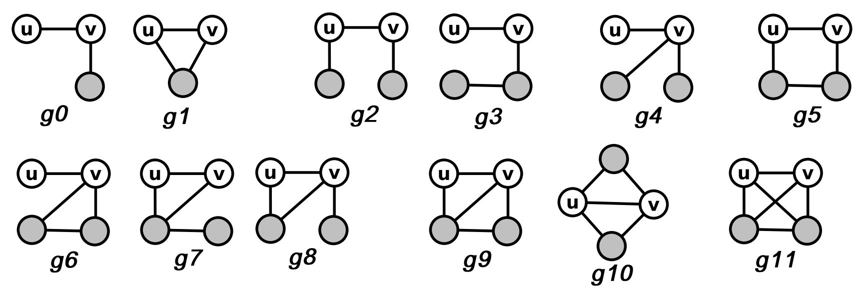
Algorithm 5: $i \in N_u$, $j \in \overline{N_{ui}}$, $k \in (N_u \cup N_v)$

for all $i \in N_u$ **do** # $i \in N_v$
    $\overline{N_{ui}} \leftarrow N_u - (N_{ui} \cup \{i\})$
    for all $j \in \overline{N_{ui}}$ **do** # $j \in \overline{N_{ui}}$
        // $k \in N_u$ // # $k \in N_v$
        $N_{uij} \leftarrow N_{ui} \cap N_{uj}$
        $f14_{ij} \leftarrow f14_{ij} + (|N_u| - |N_{ui}| - |N_{uj}| + |N_{uij}| - 2)$ // 2 is deducted for nodes i&j
        // $k \in N_v$ // # $k \in N_u$
        $N_{vij} \leftarrow N_{vi} \cap N_{vj}$
        $f26_{vij} \leftarrow f26_{vij} + |N_{vij}|$
        $f20_{vij} \leftarrow f20_{vij} + (|N_{vj}| - |N_{vij}|)$
        $f13_{vij} \leftarrow f13_{vij} + (|N_v| - |N_{vi}| - |N_{vj}| + |N_{vij}|)$
    end for
    $f14_i \leftarrow f14_i + f14_{ij}/2$
    $f13_i \leftarrow f13_i + f13_{ij}$
    $f26_i \leftarrow f26_i + f26_{ij}$
    $f20_i \leftarrow f20_i + f20_{ij}$
end for
$f14 \leftarrow f14 + f14_i/3$
$f13 \leftarrow f13 + f13_i/2$
$f26 \leftarrow f26 + f26_i/2$
$f20 \leftarrow f20 + f20_i$

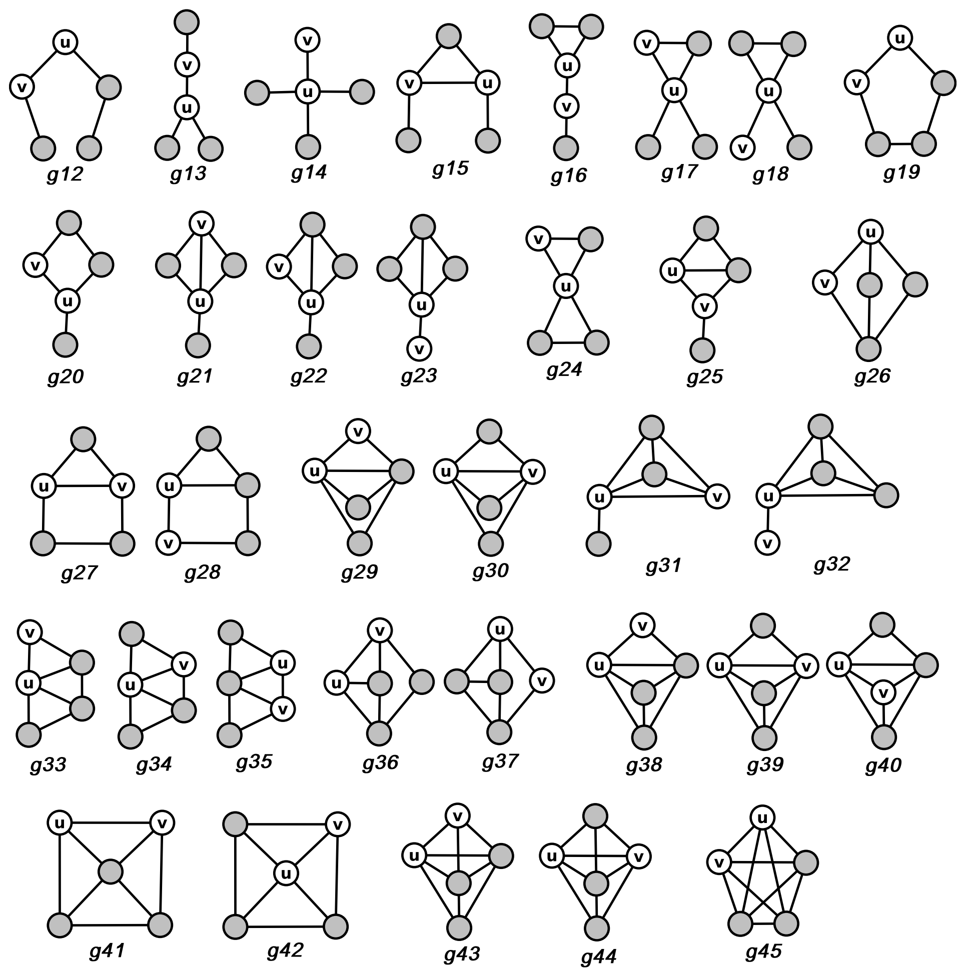
Algorithm 6: $i \in T$, $j \in (\overline{N_{Ti}} \cup \overline{N_{ui}} \cup \overline{N_{vi}})$, $k \in (N_u \cup N_v)$

for all $i \in T$ **do**
  // $j \in \overline{N_{Ti}}$
  $\overline{N_{Ti}} \leftarrow T - (N_{Ti} \cup \{i\})$
  for all $j \in \overline{N_{Ti}}$ **do**
    // $k \in T$
    $N_{Tij} \leftarrow N_{Ti} \cap N_{Tj}$
    $f30_{ij} \leftarrow f30_{ij} + (|T| - |N_{Ti}| - |N_{Tj}| + |N_{Tij}| - 2)$ // deduct counts for i & j
    // $k \in N_u$
    $N_{uij} \leftarrow N_{ui} \cap N_{uj}$
    $f21_{uij} \leftarrow f21_{uij} + (|N_u| - |N_{ui}| - |N_{uj}| + |N_{uij}|)$
    // $k \in N_v$
    $N_{vij} \leftarrow N_{vi} \cap N_{vj}$
    $f21_{vij} \leftarrow f21_{vij} + (|N_v| - |N_{vi}| - |N_{vj}| + |N_{vij}|)$
  end for
  $f30_i \leftarrow f30_i + f30_{ij}/2$
  $f21_i \leftarrow f21_i + f21_{uij} + f21_{vij}$
  // $j \in \overline{N_{ui}}$
  $\overline{N_{ui}} \leftarrow N_u - N_{ui}$
  for all $j \in \overline{N_{ui}}$ **do**
    // $k \in N_u$
    $N_{uij} \leftarrow N_{ui} \cap N_{uj}$
    $f17_{uij} \leftarrow f17_{uij} + (|N_u| - |N_{ui}| - |N_{uj}| + |N_{uij}| - 1)$ // deduct count for j
    // $k \in N_v$
    $N_{vij} \leftarrow N_{vi} \cap N_{vj}$
    $f27_{ij} \leftarrow f27_{ij} + (|N_{vj}| - |N_{vij}|)$
    $f15_{ij} \leftarrow f15_{ij} + (|N_v| - |N_{vi}| - |N_{vj}| + |N_{vij}|)$ // to avoid duplicate counting, dont count $j \in \overline{N_{vi}}$
  end for
  // $j \in \overline{N_{vi}}$
  $\overline{N_{vi}} \leftarrow N_v - N_{vi}$
  for all $j \in \overline{N_{vi}}$ **do**
    // $k \in N_v$
    $N_{vij} \leftarrow N_{vi} \cap N_{vj}$
    $f17_{vij} \leftarrow f17_{vij} + (|N_v| - |N_{vi}| - |N_{vj}| + |N_{vij}| - 1)$
  end for
  $f17_i \leftarrow f17_i + f17_{uij}/2 + f17_{vij}/2$
  $f15_i \leftarrow f15_i + f15_{ij}$
  $f27_i \leftarrow f27_i + f27_{ij}$
end for
$f30 \leftarrow f30 + f30_i/3$
$f21 \leftarrow f21 + f21_i/2$
$f17 \leftarrow f17 + f17_i$
$f15 \leftarrow f15 + f15_i$
$f27 \leftarrow f27 + f27_i$

2. ALL LOCAL GRAPHLETS WITH IDS



i: 3-4 size local graphlets



ii: 5 size local graphlets

Fig. 1: Local graphlets

3. PROBLEM FORMULATION

Let $G(V, E)$ be a simple undirected graph, where V is set of vertices and E is set of edges. Given vertex $u \in V$, neighbors of u is represented by $\Gamma(u) = \{v|(u, v) \in E\}$ and degree of u is represented by $d(u) = |\Gamma(u)|$.

An induced graphlet is a fixed size induced subgraph of a given graph (definition 3.1).

Definition 3.1 (*Induced graphlet*). For an undirected graph $g'(V', E')$, where V' is set of vertices and E' is set of edges; if $V' \subset V$, $E' = \{\forall(u, v) \in E|u, v \in V'\}$ and $|V'| = k$ then g' is called k sized induced graphlet of G. □

In this study we concentrate on counting frequency of edge centric local graphlets, which are induced graphlets such that it always include the given edge and only local neighbors of the edge. Formally we defined the edge centric local graphlet below in *definition* 3.2.

Definition 3.2 (*Edge centric local graphlet*). Given an edge (u, v), any k sized induced graphlet which includes both vertices u and v (edge (u, v)) and all other vertices in the graphlet are neighbors of u and/or v, is called k sized edge centric local graphlet. If G'_k is the set of all k sized induced graphlets of G, then for an edge (u, v), set of k sized edge centric local graphlets $G'_k(u, v) = \{\forall g'(V', E') \in G'_k|u, v \in V' \& \forall w \in V', w \in (\Gamma(u) \cup \Gamma(v))\}$ □

Figure 1, shows all 3, 4 and 5 sized local graphlets for given edge (u, v). Notice that, we separated a same graphlet with different orbits for given edge (u, v).

Definition 3.3 (*Graphlet Isomorphism*). A graphlet $g(V, E)$ is said to be isomorphic to another graphlet $g'(V', E')$, if there exists a bijective function $f_{iso} : V \to V'$ such that $(u, v) \in E \Leftrightarrow (f_{iso}(u), f_{iso}(v)) \in E'$ and the bijection function f_{iso} is called graphlet isomorphism from g to g'. □

Definition 3.4 (*Graphlet Automorphism*). A graphlet automorphism is graphlet isomorphism with itself, i.e. a bijection function f_{auto} that map the set of vertices V back to V with different permutation and satisfy the prosperities of graphlet isomorphism. □

Definition 3.5 (*Edge orbit*). For a given graphlet $g(V, E)$, edges $(u, v), (u', v') \in E$ are in same orbit if an automorphism f_{auto} of g exists such that $u = f_{auto}(u')$ and $v = f_{auto}(v')$. □

While counting the edge centric local graphlets, we consider the orbit of the given edge (u, v) and count the frequency separately for a same graphlet with different orbit for the given edge. Notice that, if two edges are in the same orbit they must have the same pair of degrees for their incident vertices. For example. in Figure 1i graphlets $g10$ and $g11$ are same 4 size graphlet (lets call it $2 - triangles$), however orbit of an edge (u, v) is different i.e. $d(u) = 2, d(v) = 3$ in $g10$ and $d(u) = 3, d(v) = 3$ in $g11$. Hence we count graphlet frequency for both $g10$ and $g11$, separately for given graph. For simplicity, we refer k sized edge centric local graphlets as k sized local graphlets and (u, v) as corresponding given edge.

Note: In Figure 1, graphlet types $g3, g7, g12$ and $g19$ are the exceptions to the k sized local graphlets, because for each of these graphlet types, there exist one vertex which is neither neighbor of u nor neighbor of v.

3.1. Problem definition

Given an edge (u, v) of an undirected graph $G(V, E)$, our goal is to count number of appearance of each k $\{3, 4, 5\}$ sized local graphlet for the edge (u, v) in the graph G.

Algorithm 7: GET_GRAPHLETCOUNT(G, u, v)

initialize frequencies // $f0 - f45 \leftarrow 0$
initialize unique neighbor sets of u and v // $N_u \leftarrow \{\}$, $N_v \leftarrow \{\}$
initialize common neighbor set // $N_{uv} \leftarrow \{\}$
initialize status of each vertex in G // **for each** x do $status\_map(x) \leftarrow \phi$
for each $x \in \Gamma(u)$ **do**
    if $x \neq v$ **then**
        $N_u \leftarrow N_u \cup x$, $status\_map(x) \leftarrow \alpha$
    end if
end for
for each $x \in \Gamma(v)$ **do**
    if $x \neq u$ **then**
        if $status\_map(x) = \alpha$ **then**
            $T \leftarrow T \cup x$, $status\_map(x) \leftarrow \gamma$
            $N_u \leftarrow N_u \setminus x$ // remove the element x from N_u
        else
            $N_v \leftarrow N_v \cup x$, $status\_map(x) \leftarrow \beta$
        end if
    end if
end for
$f0 \leftarrow |N_u| + |N_v|$ // number of unique neighbors (no triangles)
$f1 \leftarrow |T|$ // number of triangles
for each $x \in T$ **do**
    for each $y \in \Gamma(x)$ **do**
        if $status\_map(y) == \gamma$ **then**
            $f11 \leftarrow f11 + 1$ // 4clique
        else if $status\_map(y) == \phi$ **then**
            $f7 \leftarrow f7 + 1$ // tail-triangle with tail at common neighbor
        end if
    end for
    $status\_map(x) \leftarrow \rho$
end for
for each $x \in N_u$ **do**
    for each $y \in \Gamma(x)$ **do**
        if $status\_map(y) == \alpha$ **then**
            $f6 \leftarrow f6 + 1$ // tail-triangle with (u, v) as tail
        else if $status\_map(y) == \beta$ **then**
            $f5 \leftarrow f5 + 1$ // 4cycle
        else if $status\_map(y) == \rho$ **then**
            $f9 \leftarrow f9 + 1$ // 2triangles with (u, y) as diagonal link
        else if $status\_map(y) == \phi$ **then**
            $f3 \leftarrow f3 + 1$ // 4path with (u, v) as tail link
        end if
    end for
    $status\_map(x) \leftarrow \varphi$
end for
for each $x \in N_v$ **do**
    for each $y \in \Gamma(x)$ **do**
        if $status\_map(y) == \beta$ **then**
            $f6 \leftarrow f6 + 1$ // tail-triangle with (u, v) as tail
        else if $status\_map(y) == \rho$ **then**
            $f9 \leftarrow f9 + 1$ // 2triangles with (v, y) as diagonal link
        else if $status\_map(y) == \phi$ **then**
            $f3 \leftarrow f3 + 1$ // 4path with (u, v) as tail link
        end if
    end for
    $status\_map(x) \leftarrow \varphi$
end for
calculate $f2, f4, f8, f10$ using eqs. (1) to (4)
return $f, T, N_u, N_v, status\_map$

Algorithm 8: For a given graph $G = (V, E)$ and edge $e = (u, v)$, this algorithm returns local graphlet counts ($f0 - f45$) for each graphs $g0 - g45$ described in Figure 1.

1: $f, T, N_u, N_v, status\_map \leftarrow$ GET_GRAPHLETCOUNT(G, u, v)
2: $f \leftarrow$ GET_5SIZED-PATH-CYCLE$(G, u, v, f, T, N_u, N_v, status_map)$
3: calculate all 5 sized local graphlets using template algorithms 10,11,12 and 13 with values mentioned in Table I.
4: **return** f.

4. PROPOSED METHOD

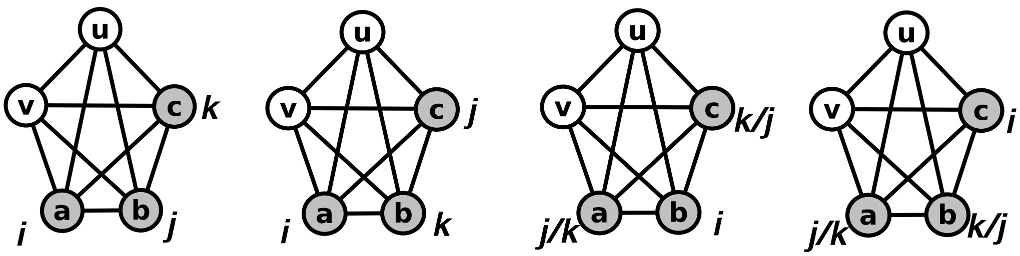
Algorithm 8 returns graphlet frequency (counts) for each graphlet types depicted in Figure 1. Here, line 1 calls function GET_GRAPHLETCOUNT(G, u, v) (Algorithm 7), which returns graphlet frequencies for each 3-4 sized graphlet types shown in the Figure 1i. Second line of Algorithm 8, calculates frequency of exception type 5 sized graphlets ($g12, g19$) using Algorithm 9 (GET_5SIZED-PATH-CYCLE()). In the algorithms, frequency for a graphlet type gi is represented as f_i and we use notations described in sub-section 1.1.

Algorithm 9: GET_5SIZED-PATH-CYCLE$(G, u, v, f, T, N_u, N_v, status_map)$

$f19_i, f12_i \leftarrow 0$
for each $i \in N_u$ **do**
  $N'_{vi} \leftarrow N_v - N_{vi}$
  $f19_{ij}, f12_{ij} \leftarrow 0$
  for each $j \in \Gamma(i)$ **do**
    if $status\_map(j) == \phi$ **then**
      $N'_{vij} \leftarrow N'_{vi} \cap N_{vj}$
      $f19_{ij} \leftarrow f19_{ij} + |N'_{vij}|$
      $f12_{ij} \leftarrow f12_{ij} + (|N'_{vi}| - |N'_{vij}|)$
    end if
  end for
  $f19_i \leftarrow f19_i + f19_{ij}$
  $f12_i \leftarrow f12_i + f12_{ij}$
end for
for each $i \in N_v$ **do**
  $N'_{ui} \leftarrow N_u - N_{ui}$
  $f12_{ij} \leftarrow 0$
  for each $j \in \Gamma(i)$ **do**
    if $status\_map(j) == \phi$ **then**
      $N'_{uij} \leftarrow N'_{ui} \cap N_{uj}$
      $f12_{ij} \leftarrow f12_{ij} + (|N'_{ui}| - |N'_{uij}|)$
    end if
  end for
  $f12_i \leftarrow f12_i + f12_{ij}$
end for
$f12 \leftarrow f12_i$
$f19 \leftarrow f19_i$
return f

4.1. Template Algorithm

Algorithm 10 is a template algorithm for finding frequency of different 5 sized local graphlets. In the algorithm, $S1, S2, k-var, t, b_i, d_i$ and d are template variables and a specific set of values of these variables gives frequency of a specific graphlet. The detailed information is available in Table I.



i: 5 clique graphlet $(i = a)$ ii: 5 clique graphlet $(i = b/c)$

Fig. 2: 5 clique graphlet counting

Example: Frequency of the 5 size clique i.e. graphlet $g45$ can be calculated using template variable values $S1 = T$, $S2 = N_{Ti}$, $k - var = 0$, $t = 1$, $b_i = 0$,$d_i = 2$ and $d = 3$. As shown in the Figure 2, three vertices (a, b, c) other than u and v need to be from set T, hence $S1 = T$ and $k - var = 0$. Also all three vertices need to be interconnected, so $S2 = N_{Ti}$ and $t = 1$. Now, as we can see for each i selection of j and k are interchangeable i.e. in Figure 2i j and k is interchangeable between vertices b and c. Therefore this graphlet will be counted twice for each i, which leads to $d_i = 2$. Similarly, from Figure 2ii, we can see that we select all of the three vertices (a, b, c) as an i one by one (Algorithm 10: line 1), hence we need to divide the total count by 3 i.e. $d = 3$.

Algorithm 10: Template Algorithm

1: **for all** $i \in S1$ **do**
2:   **for all** $j \in S2$ **do**
3:     **if** k-var == 0 **then** # $k \in T$
4:       $f1_{ij}, f2_{ij}, f3_{ij}, f4_{ij} \leftarrow$ get_freq_k_inT(i, j)
5:     **else if** k-var == 1 **then** # $k \in N_u$
6:       $f1_{ij}, f2_{ij}, f3_{ij}, f4_{ij} \leftarrow$ get_freq_k_inNu(i, j)
7:     **else** # $k \in N_v$
8:       $f1_{ij}, f2_{ij}, f3_{ij}, f4_{ij} \leftarrow$ get_freq_k_inNv(i, j)
9:     **end if**
10:   **end for** // end of loop for j
11:   // select $t \in \{1, 2, 3, 4\}$
12:   $ft_i \leftarrow ft_i + (ft_{ij} - b_i)/d_i$
13: **end for** // end of loop for i
14: $ft \leftarrow ft + ft_i/d$

Algorithm 11: get_freq_k_inT(i, j)

$N_{Tij} \leftarrow N_{Ti} \cap N_{Tj}$
$f1_{ij} \leftarrow f1_{ij} + |N_{Tij}|$
$f2_{ij} \leftarrow f2_{ij} + (|N_{Ti}| - |N_{Tij}|)$
$f3_{ij} \leftarrow f3_{ij} + (|N_{Tj}| - |N_{Tij}|)$
$f4_{ij} \leftarrow f4_{ij} + (|T| - |N_{Ti}| - |N_{Tj}| + |N_{Tij}|)$
return $f1_{ij}, f2_{ij}, f3_{ij}, f4_{ij}$

Algorithm 12: get_freq_k_inNu(i, j)

$N_{uij} \leftarrow N_{ui} \cap N_{uj}$
$f1_{ij} \leftarrow f1_{ij} + |N_{uij}|$
$f2_{ij} \leftarrow f2_{ij} + (|N_{ui}| - |N_{uij}|)$
$f3_{ij} \leftarrow f3_{ij} + (|N_{uj}| - |N_{uij}|)$
$f4_{ij} \leftarrow f4_{ij} + (|N_u| - |N_{ui}| - |N_{uj}| + |N_{uij}|)$
return $f1_{ij}, f2_{ij}, f3_{ij}, f4_{ij}$

Algorithm 13: get_freq_k_inNv(i, j)

$N_{vij} \leftarrow N_{vi} \cap N_{vj}$
$f1_{ij} \leftarrow f1_{ij} + |N_{vij}|$
$f2_{ij} \leftarrow f2_{ij} + (|N_{vi}| - |N_{vij}|)$
$f3_{ij} \leftarrow f3_{ij} + (|N_{vj}| - |N_{vij}|)$
$f4_{ij} \leftarrow f4_{ij} + (|N_v| - |N_{vi}| - |N_{vj}| + |N_{vij}|)$
return $f1_{ij}, f2_{ij}, f3_{ij}, f4_{ij}$

4.2. Counting frequency of non-symmetric local graphlets

Definition 4.1 (*Vertex orbit*). For a given graphlet $g(V, E)$, vertexes $u, v \in V$ are in same orbit if an automorphism f_{auto} of g exists such that $u = f_{auto}(v)$. □

Definition 4.2 (*Symmetric local graphlets*). For a given edge (u, v) a local graphlet g is called symmetric if vertices u and v are in the same orbit for g. □

For symmetric local graphlets we do not need to count frequency for reverse sequence of the vertices v, u (instead of u, v). For a symmetric local graphlet, $d(u) = d(v)$ and graphlet structure remains same after exchanging neighbourhood of the vertices u and v. For example, as shown in Figure 3i for graphlet type $g35$, $d(u) = d(v) = 3$ and when we exchange neighbors of u and v the graphlet structure remains same. On the other hand, Figure 3ii shows two examples of non-symmetric graphlets of type $g23$ and $g31$.

To calculate correct frequency for non-symmetric graphlets, we need to count occurrence of the graphlet with reverse sequence of the vertices. For example, in Figure 3ii(1) a, b and c all are neighbors of u and $d(u) = 4$. However to count the correct frequency of the graphlet type $g23$, we need to count frequency for the graphlet shown in Figure 3ii(2) where $d(v) = 4$ and a', b' and c' all are neighbors of v. To count the correct frequency for the graphlet in Figure 3ii(1), template variables take values $S1 = N_u$, $S2 = N_{ui}$ and $k - var = 1$. While to count frequency of the graphlet in Figure 3ii(2), the variable values become $S1_r = N_v$, $S2_r = N_{vi}$ and $k - var_r = 2$, where $S1_r$ represents reverse version of $S1$ as shown in Table I. Template variables t, b_i, d_i and d remains same for both cases. For graphlet type $g23$, when we calculate $f3_{ij}$ using Algorithm 12,

Table I: Template Table

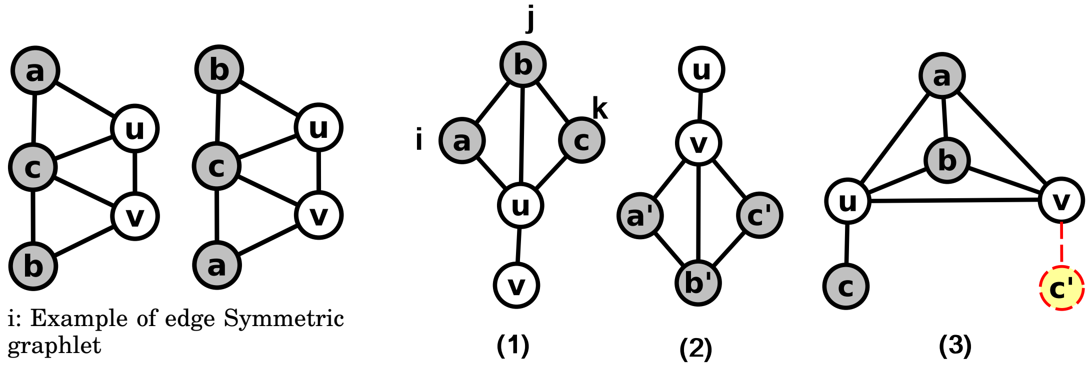
graphlet	$S1$	$S2$	$k - var$	t	b_i	d_i	d	$S1_r$	$S2_r$	$k - var_r$
13	N_u	$\overline{N_{ui}}$	2	4	0	1	2	N_v	$\overline{N_{vi}}$	1
14	N_u	$\overline{N_{ui}}$	1	4	$\|S2\| * 2$	2	3	N_v	$\overline{N_{vi}}$	2
15	T	$\overline{N_{ui}}$	2	4	0	1	1	–	–	–
16	N_u	N_{ui}	2	4	0	1	2	N_v	N_{vi}	1
17	T	$\overline{N_{ui}}$	1	4	$\|S2\|$	2	1	T	$\overline{N_{ui}}$	2
18	N_u	N_{ui}	1	4	0	1	2	N_v	N_{vi}	2
20	N_u	$\overline{N_{ui}}$	2	3	0	1	1	N_v	$\overline{N_{vi}}$	1
21	T	$\overline{N_{Ti}}$	$1 + 2$	4	0	1	2	–	–	–
22	T	N_{ui}	1	4	0	1	1	T	N_{vi}	2
23	N_u	N_{ui}	1	3	$\|S2\|$	1	2	N_v	N_{vi}	2
24	N_u	N_{ui}	0	4	0	1	2	N_v	N_{vi}	0
25	T	N_{ui}	2	4	0	1	1	T	N_{vi}	1
26	N_u	$\overline{N_{ui}}$	2	1	0	1	2	N_v	$\overline{N_{vi}}$	1
27	T	$\overline{N_{ui}}$	2	3	0	1	1	–	–	–
28	N_u	N_{ui}	2	3	0	1	1	N_v	N_{vi}	1
29	T	N_{ui}	1	2	0	2	1	T	N_{vi}	2
30	T	$\overline{N_{Ti}}$	0	4	$\|S2\| * 2$	2	3	–	–	–
31	T	N_{Ti}	$1 + 2$	4	0	1	2	–	–	–
32	N_u	N_{ui}	1	1	0	2	3	N_v	N_{vi}	2
33	T	N_{ui}	1	3	0	1	1	T	N_{vi}	2
34	T	N_{ui}	0	4	0	1	1	T	N_{vi}	0
35	T	N_{ui}	2	2	0	1	1	–	–	–
36	T	N_{ui}	2	3	0	1	1	T	N_{vi}	1
37	N_u	N_{ui}	2	1	0	1	2	N_v	N_{vi}	1
38	T	N_{ui}	1	1	0	2	1	T	N_{vi}	2
39	T	N_{Ti}	0	4	0	1	2	–	–	–
40	T	N_{ui}	0	2	0	1	1	T	N_{vi}	0
41	T	N_{ui}	2	1	0	1	1	–	–	–
42	T	N_{ui}	0	3	$\|S2\|$	1	2	T	N_{vi}	0
43	T	N_{ui}	0	1	0	1	2	T	N_{vi}	0
44	T	N_{Ti}	0	3	$\|S2\|$	1	2	–	–	–
45	T	N_{Ti}	0	1	0	2	3	–	–	–

N_{uj} also includes vertex represented by i (here a) hence we need to subtract 1 from $f3_{ij}$ for each j. We subtract this value combined using bias $b_i = |S2|$. We also count the same graphlet two times considering $i = a$ and $i = c$, hence we divide the global count by $d = 2$.

Another example of non-symmetric graphlet is graphlet type $g31$. To count correct frequency for this type, we just need to sum up possible unique neighbors of u and unique neighbors of v (with red dotted line). To calculate both cases as shown in Table I, we just need to put two different values (1 and 2) of template variable $k - var$ and sum the resultant frequency values.

4.3. Time complexity

For 5-size local graphlets counting, any regular algorithm generally takes $O(\Delta^3)$ time for each edge [GRAFT]. However, time complexity of our method (Algorithm 10) is $O((T^{max} + N_u^{max} + N_v^{max})^3)$, where T^{max} is the largest value of $|T|$ out of all edges of the



i: Example of edge Symmetric graphlet

ii: Examples of edge non-symmetric graphlets

Fig. 3: Example of edge symmetric and non-symmetric graphlets

graph. Similarly N_u^{max} and N_v^{max} represents largest value of $|N_u|$ and $|N_v|$ respectively. For any edge (u, v), $|N_u| = d(u) - |T|$ and $|N_v| = d(v) - |T|$, and for any real-world sparse network $0 \leq T^{max} \ll \Delta$, hence $N_u^{max} \leq \Delta$ and $N_v^{max} \leq \Delta$. However, for any node with highest degree there is very low probability that $|T| = 0$ [need reference / proof]