Fast and Accurate Estimation of Typed Graphlets
Ryan A. Rossi\textsuperscript{1} | Anup Rao\textsuperscript{1}, Tung Mai\textsuperscript{1}, Nesreen K. Ahmed\textsuperscript{2}

\textsuperscript{1}Adobe Research
\textsuperscript{2}Intel Labs
Problem

- Typed graphlets = small typed (labeled) induced sub-graphs
- Generalization of graphlets to labeled and heterogeneous networks
- Useful for many applications including clustering, link prediction, network alignment, etc.

**Exact Problem.** Given a graph G with L types, the global typed graphlet counting problem is to find the set of all typed graphlets that occur in G along with their corresponding frequencies.

- Depending on the application constraints, speed may be more important than accuracy
  - e.g., applications requiring real-time response rates such as online recommendation, online advertisements, among many others

**QUESTION:** Can we instead obtain fast estimates with provable error guarantees?
Our Problem. Given a graph $G$ with $L$ types, the typed graphlet estimation problem is to accurately estimate the counts of all typed graphlets that occur in $G$ while achieving orders of magnitude speedup compared to exact algorithms.

- Typed graphlets = small typed (labeled) induced sub-graphs
- Generalization of graphlets to labeled and heterogeneous networks
- Useful for many applications including clustering, link prediction, network alignment, etc.
- Depending on the application constraints, speed may be more important than accuracy
Our Problem. Given a graph $G$ with $L$ types, the typed graphlet estimation problem is to accurately estimate the counts of all typed graphlets that occur in $G$ while achieving orders of magnitude speedup compared to exact algorithms.

- Typed graphlets = small typed (labeled) induced sub-graphs
- Generalization of graphlets to labeled and heterogeneous networks
- Useful for many applications including clustering, link prediction, network alignment, etc.
- Depending on the application constraints, speed may be more important than accuracy

Balance tradeoffs: tiny decrease in accuracy for significant improvement in runtime (100-1000x speedup)
Framework for Typed Graphlet Estimation

Introduce two general classes of typed graphlet estimation methods:

1. Typed *Edge* Sampling (TES) & Estimation

2. Typed *Path* Sampling (TPS) & Estimation
Typed Edge Sampling (TES)

- Typed Edge Sampling \( J \subseteq E \)

- Estimation
  \[
  X_H = \begin{bmatrix} x_1 & x_2 & \cdots \end{bmatrix} \in \mathbb{R}^{|J| \times |\mathcal{H}|}
  \]

  \[
  \hat{x}_H = \left( \frac{|J|}{|E|} \right)^{-1} \frac{e^T X_H}{|E(H)|}
  \]

  (typed graphlet counts that occur at each sampled edge in \( J \) for a specific induced subgraph \( H \) (e.g., 4-clique) and \( \mathcal{H} \) is the set of typed graphlets of \( H \))
Typed Path Sampling (TPS)

- Sampling

**Definition 1 (Typed Wedges).** Given an edge \((i, j) \in E\) with types \(\phi_i\) and \(\phi_j\), the \((i, j)\)-entry of the typed wedge matrix with types \(t\) and \(t'\) is:

\[
\Lambda_e^{tt'} = \Lambda_{ij}^{tt'} = \begin{cases} 
(d_i^t - 1)(d_j^{t'} - 1) & \text{if } t = \phi_j \land t' = \phi_i \\
(d_i^t - 1)d_j^{t'} & \text{if } t = \phi_j \land t' \neq \phi_i \\
d_i^t(d_j^{t'} - 1) & \text{if } t \neq \phi_j \land t' = \phi_i \\
d_i^t d_j^{t'} & \text{otherwise}
\end{cases}
\]  

(3)

**Algorithm 1** Typed Path Sample

**Output:** the four sampled nodes \((i', i, j, j')\) with types \((t_1, t, t', t_2)\) that form a typed 4-path with edges \(((i', i), (i, j), (j, j'))\)

1. Compute \(\Lambda_e^{tt'}\) (Eq. 3) for all typed edges and set
   \[
p_e^{tt'} = \Lambda_e^{tt'}/W^{tt'}
\]

2. Select \(e = (i, j)\) of type \(t_e = (t, t')\) with probability \(p_e^{tt'}\)

3. Select \(i' \in \Gamma_i^{t_1}\) with type \(t_1\) uniformly at random s.t. \(i' \neq j\) if \(\phi_j = t_1\).

4. Select \(j' \in \Gamma_j^{t_2}\) with type \(t_2\) uniformly at random s.t. \(j' \neq i\) if \(\phi_i = t_2\).
Typed Path Sampling (TPS)

- Estimation

\[ C_{i,t} = \text{# of occurrences of the i-th typed induced subgraph with types } t \]

\[ A_{ij} = \text{# of distinct copies of the i-th typed subgraph in the j-th subgraph} \]

\[ N_{i,t} = \text{count of the i-th typed non-induced subgraph with types } t \]

---

**Algorithm 2** Estimation via Typed Paths

**Input:** graph \( G \), # samples \( k \)

**Output:** estimated counts for all typed 4-node graphlets

1. Obtain \( k \) samples (sets of vertices) by running Alg. 1 \( k \) times where \( S_j \) denotes the \( j \)-th set of vertices.
2. \textbf{parallel for} \( j = 1, \ldots, k \) \textbf{do}
   - Determine subgraph induced by \( S_j \) (and type vector \( t \))
   - If this is the \( i \)-th graphlet with types \( t \), increment \( F_{i,t}^{t,t'} \)
   - Increment \( k^{t,t'} \) where \( t, t' \) are the other two node types
3. \textbf{for} \( i \in [2, 6] \) and type vector \( t \) \textbf{do}
4. \textbf{for all } \( t, t' \in \{1, \ldots, L\} \) \textbf{do}
   - Set \( \hat{C}_{i,t} = \hat{C}_{i,t} + (F_{i,t}^{t,t'} / k^{t,t'}) \cdot \frac{W_{t,t'}}{A_{2,i}} 
   - Set \( \hat{C}_{i,t} = \hat{C}_{i,t} / 2 \)
5. Set \( \hat{C}_{1,t} = N_{1,t} - \hat{C}_{3,t} - 2\hat{C}_{5,t} - 4\hat{C}_{6,t} \) \, \forall t \text{ s.t. } N_{1,t} \text{ is computed via Eq. 4} \]
## Results

Mean relative error of typed graphlet estimates.
We set $k = 50000$ and perform 100 runs.

<table>
<thead>
<tr>
<th>Data</th>
<th>Methods</th>
<th>(\text{i} )</th>
<th>(\text{y} )</th>
<th>(\text{m} )</th>
<th>(\text{n} )</th>
<th>(\text{m} \times \text{i} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>fb-political</td>
<td>TES</td>
<td>0.012</td>
<td>0.033</td>
<td>0.036</td>
<td>0.036</td>
<td>0.070</td>
</tr>
<tr>
<td></td>
<td>TPS</td>
<td><strong>0.002</strong></td>
<td><strong>0.021</strong></td>
<td><strong>0.010</strong></td>
<td><strong>0.024</strong></td>
<td><strong>0.034</strong></td>
</tr>
<tr>
<td>yahoo-msg</td>
<td>TES</td>
<td>0.774</td>
<td>0.867</td>
<td>1.233</td>
<td>2.784</td>
<td>0.430</td>
</tr>
<tr>
<td></td>
<td>TPS</td>
<td><strong>0.001</strong></td>
<td><strong>0.046</strong></td>
<td><strong>0.003</strong></td>
<td><strong>0.270</strong></td>
<td><strong>0.025</strong></td>
</tr>
<tr>
<td>web-polblogs</td>
<td>TES</td>
<td>0.010</td>
<td>0.083</td>
<td>0.011</td>
<td>0.100</td>
<td>0.164</td>
</tr>
<tr>
<td></td>
<td>TPS</td>
<td><strong>0.002</strong></td>
<td><strong>0.006</strong></td>
<td><strong>0.005</strong></td>
<td><strong>0.007</strong></td>
<td><strong>0.008</strong></td>
</tr>
<tr>
<td>soc-wiki-elec</td>
<td>TES</td>
<td>0.684</td>
<td>1.160</td>
<td>0.788</td>
<td>1.353</td>
<td>1.552</td>
</tr>
<tr>
<td></td>
<td>TPS</td>
<td><strong>0.002</strong></td>
<td><strong>0.003</strong></td>
<td><strong>0.003</strong></td>
<td><strong>0.005</strong></td>
<td><strong>0.008</strong></td>
</tr>
<tr>
<td>soc-digg</td>
<td>TES</td>
<td>0.460</td>
<td>0.412</td>
<td>0.890</td>
<td>0.713</td>
<td>1.474</td>
</tr>
<tr>
<td></td>
<td>TPS</td>
<td>$&lt;10^{-3}$</td>
<td><strong>0.004</strong></td>
<td><strong>0.003</strong></td>
<td><strong>0.006</strong></td>
<td><strong>0.011</strong></td>
</tr>
</tbody>
</table>
Variance of estimates

- The standard deviation/variance of TPS to be about an order of magnitude smaller than TES

### Table 2: Typed graphlet estimates and relative error using $k = 50000$ (typed 4-cliques).

| graph      | types | $C$  | TES | TPS | $\frac{|C-\hat{C}|}{C}$ | TES | TPS | Std |
|------------|-------|------|-----|-----|-------------------------|-----|-----|-----|
| fb-political | 1111  | 8.14K| 7.35K| 8.37K| 0.0973 | 0.0288 | 5K | 507 |
|            | 2111  | 7.64K| 8.13K| 7.86K| 0.0634 | 0.0285 | 3.4K | 545 |
|            | 2211  | 6.27K| 6.85K| 6.50K| 0.0924 | 0.0371 | 2.6K | 448 |
|            | 2221  | 6.12K| 6.55K| 6.29K| 0.0701 | 0.0281 | 2.3K | 455 |
|            | 2222  | 4.46K| 4.59K| 4.68K| 0.0274 | 0.0483 | 2.4K | 462 |
Convergence Results

- Convergence of estimates
- Increasing the sample size $k$ decreases error of TPS
- Error decreases towards zero as the sample size increases
Summary of Contributions

- Proposed estimation framework for Typed Graphlets

- Described two generic estimators
  - Typed Edge Sampling & Estimation (TES)
  - Typed Path Sampling & Estimation (TPS)

- TPS achieves better accuracy (lower rel. error) & lower variance than TES

- Convergence of estimates (error decreases towards zero as sample size increases)

- Speedup is significant taking less than a second for all graphs (100+ times faster than exact alg.)
Thanks for listening!