DENSE SUBGRAPH DISCOVERY IN LARGE GRAPHS
Charalampos (Babis) E. Tsourakakis
babis@seas.harvard.edu

Motivation
- Anomaly detection in “who-chose-who” network, large sets of vertices which look like cliques are suspicious.
- Vertices correspond to humans
- Edges denote at least one phone call exchange
- Many more applications rely on dense subgraph discovery, correlation mining, graph visualization, mining Twitter data, bioinformatics.

Main contributions
Theorem 1 (STOC’15) Let \( \epsilon \in (0, 1) \), \( \lambda > 1 \) constant and \( T = \lfloor n^\lambda \rfloor \).
- There is an algorithm that processes the first \( T \) updates in the dynamic stream such that:
  - It uses \( \tilde{O}(n) \) space (Space efficiency)
  - It maintains a value \( \text{OUTPUT}(t) \) at each \( t \in [T] \) such that for all \( t \in [T] \) whp
    \[ \text{OUTPUT}(t)/(4 + \Theta(\epsilon)) \leq \text{OPT}(t) \leq \text{OPT}(t). \]

Theorem 2 (STOC’15) We can process a dynamic stream of updates in the input graph \( G \in O(n) \) space, with a single pass and with high probability return a \((2 + \Theta(\epsilon))\)-approximation of \( d^* = \max_{S \subseteq V} \rho(S) \) at the end of the stream.

Theorem 3 (KDD’15) Sample each edge \( e \in E_H \) independently with probability \( p = \frac{\log n}{G} \). Then, the following statements hold simultaneously with high probability:
- For all \( U \subseteq V \) such that \( p(U) \geq D, \rho(U) \geq (1 - \epsilon)\log n \) for any \( \epsilon > 0 \).
- For all \( U \subseteq V \) such that \( p(U) < (1 - 2\epsilon)D, \rho(U) < (1 - \epsilon)\log n \) for any \( \epsilon > 0 \).

Corollary 1 (KDD’15) We improve the approximation guarantee of the above dynamic streaming algorithm to \( (1 + \Theta(\epsilon)) \).

Theorem 4 (WWW’15) Consider the following generalization of the DSP, the \( k \)-clique DSP. The goal is to maximize the \( k \)-clique density \( h_k(S), k \geq 2 \) as \( h_k(S) = \frac{|S|^k}{2^k} \).
- For any constant \( K \), the \( K \)-clique densest subgraph problem can be solved exactly in polynomial time.
- Furthermore, we can \( 1/3 \)-approximate it using any \( K \)-clique counting algorithm as subroutine.

Key concept – \((\alpha, d, L)\)-decomp
Definition 1 Fix any \( \alpha \geq 1, d \geq 0, \) and any positive integer \( L \). Consider a family of subsets \( Z_i \subseteq \ldots \subseteq Z_{i_L} \). The tuple \((Z_1, \ldots, Z_L)\) is an \((\alpha, d, L)\)-decomposition of the input graph \( G = (V, E) \) if \( Z_i = V \) and, for every \( i \in [L - 1] \), we have \( Z_{i+1} \supseteq \{v \in Z_i : D_{Z_i}(Z_{i+1}) > \alpha d \} \) and \( Z_{i+1} \cap \{v \in Z_i : D_{Z_i}(Z_{i+1}) < d \} = \emptyset \).

Two key properties of the \((\alpha, d, L)\)-decomposition follow.
Theorem 5 Fix any \( \alpha \geq 1, d \geq 0, \epsilon \in (0, 1) \), \( L \leq 2 + \lfloor \log_{1+\epsilon} n \rfloor \). Let \( d^* \leftarrow \max_{S \subseteq V} \rho(S) \) be the maximum density of any subgraph in \( G = (V, E) \), and let \((Z_1, \ldots, Z_L)\) be an \((\alpha, d, L)\)-decomposition of \( G = (V, E) \). We have: (1) If \( d > 2(1 + \epsilon)d^* \), then \( Z_L = \emptyset \), and (2) if \( d < \alpha d^* \), then \( Z_L \neq \emptyset \).

(Rough) Idea of how to turn the previous theorem into an algorithm.
- Discretize the range of \( d^* \) as \( d_k \leftarrow (1 + \epsilon)^{k-1} \frac{m}{n} \), \( k \in [K] \) where \( K = O(\log_{1+\epsilon} n) \).
- For every \( k \in [K] \), construct an \((\alpha, d_k, L)\)-decomposition \((Z_1(k), \ldots, Z_L(k))\), where \( L = O(\log_{1+\epsilon} n) \).

Then we have the following guarantees:
1. \( d^*/(1 + \epsilon) \leq d_k \leq 2(1 + \epsilon) \cdot d^* \).
2. There exists an index \( j^* \in [L] \) such that \( \rho(Z_{j^*}) \geq d_k/(2(1 + \epsilon)) \).

Experimental results

<table>
<thead>
<tr>
<th>k-cliques</th>
<th>( k = 2 )</th>
<th>( k = 3 )</th>
<th>( k = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_x )</td>
<td>( \frac{</td>
<td>S</td>
<td>}{</td>
</tr>
<tr>
<td>0.12</td>
<td>1.012</td>
<td>0.26</td>
<td>432</td>
</tr>
<tr>
<td>0.11</td>
<td>1.8686</td>
<td>0.80</td>
<td>76</td>
</tr>
<tr>
<td>0.19</td>
<td>16.714</td>
<td>0.54</td>
<td>102</td>
</tr>
<tr>
<td>0.13</td>
<td>553</td>
<td>0.38</td>
<td>167</td>
</tr>
</tbody>
</table>

(p,q)-bicliques

<table>
<thead>
<tr>
<th></th>
<th>( (p, q) = (1, 1) )</th>
<th>( (p, q) = (2, 2) )</th>
<th>( (p, q) = (3, 3) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_x )</td>
<td>( \frac{</td>
<td>S</td>
<td>}{</td>
</tr>
<tr>
<td>0.001</td>
<td>9.177</td>
<td>0.06</td>
<td>181</td>
</tr>
<tr>
<td>0.001</td>
<td>6.437</td>
<td>0.41</td>
<td>18</td>
</tr>
</tbody>
</table>

Experimental results.
- Effect of sampling on Epinions network.

Open problems
- Can we improve the \((4 + \epsilon)\) approximation guarantee? What about weighted graphs?
- Space- and time-efficient fully dynamic algorithm for other graph problems, e.g., single-source shortest paths?

References