EFFICIENT APPROXIMATION ALGORITHMS FOR SPANNING CENTRALITY

Shiqi Zhang, Renchi Yang, Jing Tang, Xiaokui Xiao, Bo Tang

August 2023
SPANNING CENTRALITY

- Input:
  - an undirected and connected graph $G$

- Spanning centrality $s(e_{i,j}) \in (0,1]$ of an edge $e_{i,j}$:
  - the fraction of spanning trees of $G$ that contains $e_{i,j}$

- A higher SC $s(e_{i,j})$:
  - $e_{i,j}$ is more crucial for $G$ to ensure connectedness.
Spanning centrality $s(e_{i,j}) \in (0,1]$ of an edge $e_{i,j}$:
- the fraction of spanning trees of $G$ that contains $e_{i,j}$

$G$

spanning trees

\[ s(\text{blue edges}) = \frac{2}{3} \]

\[ s(\text{blue edges}) = 1 \]
SPANNING CENTRALITY

- Applications:
  - stability and robustness analysis
  - information propagation analysis
  - graph sparsification
  - collaborative recommendation
  - image segmentation
  - etc.
PROBLEM DEFINITION

- All Edge Spanning Centrality (AESC)
  - input:
    - an undirected & connected graph $G$ with $n$ nodes and $m$ edges
  - output:
    - $s(e_{i,j})$ for every edge $e_{i,j}$ in $G$
  - time complexity:
    - $O \left( mn^2 \right)$
PROBLEM DEFINITION

- $\epsilon$-approximate AESC
  - input:
    - an undirected & connected graph $G$
    - an absolute error $\epsilon$
  - output:
    - the estimated SC $\hat{s}(e_{i,j})$ for every edge $e_{i,j}$ in $G$ satisfying

\[ |s(e_{i,j}) - \hat{s}(e_{i,j})| \leq \epsilon \]
SOTA FOR $\epsilon$-APPROXIMATE AESC

- **Simple random walk from node $v_0$:**

  \[ p_\ell(v_i, v_j) = \Pr[\text{A simple random walk from } v_i \text{ visits } v_j \text{ at the } \ell\text{-th step}] \]

- **SC in a view of simple random walk [Peng et al. KKD’21]:**

  \[
  s(e_{i,j}) = \sum_{\ell=0}^{+\infty} \left( \frac{p_\ell(v_i, v_j)}{d(v_i)} + \frac{p_\ell(v_j, v_j)}{d(v_j)} - \frac{p_\ell(v_i, v_j)}{d(v_j)} - \frac{p_\ell(v_j, v_i)}{d(v_i)} \right) \frac{1}{\text{#neighbors of } v_i}
  \]
SOTA FOR $\epsilon$-APPROXIMATE AESC

- [Peng et al. KKD’21] uses the random walk interpretation

$$s(e_{i,j}) = \sum_{\ell=0}^{\tau} \frac{p_{\ell}(v_i, v_i)}{d(v_i)} + \frac{p_{\ell}(v_j, v_j)}{d(v_j)} - \frac{p_{\ell}(v_i, v_j)}{d(v_j)} - \frac{p_{\ell}(v_j, v_i)}{d(v_i)} + \sum_{\ell=\tau+1}^{+\infty} \frac{p_{\ell}(v_i, v_i)}{d(v_i)} + \frac{p_{\ell}(v_j, v_j)}{d(v_j)} - \frac{p_{\ell}(v_i, v_j)}{d(v_j)} - \frac{p_{\ell}(v_j, v_i)}{d(v_i)}$$

Estimate this by simple random walk sampling with at most an $\epsilon/2$ error

Derive a random walk length threshold $\tau$ s.t. the error of ignoring this part is at most $\epsilon/2$

- Expensive computational overhead
  - Large random walk length threshold $\tau$
  - Large number of random walks
OUR TECHNICAL CONTRIBUTIONS

\[ s(e_{i,j}) = \sum_{\ell=0}^{\tilde{\tau}} \frac{p_{\ell}(v_i, v_i)}{d(v_i)} + \frac{p_{\ell}(v_j, v_j)}{d(v_j)} - \frac{p_{\ell}(v_i, v_j)}{d(v_j)} - \frac{p_{\ell}(v_j, v_i)}{d(v_i)} \]

\[ + \sum_{\ell=\tilde{\tau}+1}^{\tau} \frac{p_{\ell}(v_i, v_i)}{d(v_i)} + \frac{p_{\ell}(v_j, v_j)}{d(v_j)} - \frac{p_{\ell}(v_i, v_j)}{d(v_j)} - \frac{p_{\ell}(v_j, v_i)}{d(v_i)} \]

\[ + \sum_{\ell=\tau+1}^{+\infty} \frac{p_{\ell}(v_i, v_i)}{d(v_i)} + \frac{p_{\ell}(v_j, v_j)}{d(v_j)} - \frac{p_{\ell}(v_i, v_j)}{d(v_j)} - \frac{p_{\ell}(v_j, v_i)}{d(v_i)} \]

Compute the first \( \tilde{\tau} \) steps by deterministic graph traversal

Estimate the rest \( \tau - \tilde{\tau} \) steps by random walk sampling

Tighten the random walk length threshold \( \tau \)
OUR TECHNICAL CONTRIBUTIONS

- Tightened length threshold $\tau_{i,j}$ personalized to each $e_{i,j}$

\[
s(e_{i,j}) = \sum_{\ell=0}^{+\infty} \frac{p_{\ell}(v_i, v_i)}{d(v_i)} + \frac{p_{\ell}(v_j, v_j)}{d(v_j)} - \frac{p_{\ell}(v_i, v_j)}{d(v_j)} - \frac{p_{\ell}(v_j, v_i)}{d(v_i)}
\]

- endpoints with larger degrees ↓
  - smaller SC values ↓
  - smaller $\tau$ to satisfy $\epsilon/2$ ↓
  - utilize the degree information of two endpoints

- can be decomposed by graph spectral property ↓
  - utilize the eigenvectors and eigenvalues of $G$
Our Technical Contributions

\[ s_{\tau}(e_{i,j}) = \sum_{\ell=0}^{\tau} p_\ell(v_i, v_i) \frac{d(v_i)}{d(v_j)} + \sum_{\ell=\tau+1}^{\tau} p_\ell(v_i, v_i) \frac{d(v_i)}{d(v_j)} - \sum_{\ell=\tau+1}^{\tau} p_\ell(v_i, v_i) \frac{d(v_i)}{d(v_j)} \]

deterministic graph traversal

random walk sampling

Switch to sampling when the cost of the former exceeds the latter
OUR TECHNICAL CONTRIBUTIONS

- Deterministic graph traversal

\[
\sum_{\ell=0}^{\bar{\ell}} \frac{p_\ell(v_i, v_i)}{d(v_i)} + \frac{p_\ell(v_j, v_j)}{d(v_j)} - \frac{p_\ell(v_i, v_j)}{d(v_j)} - \frac{p_\ell(v_j, v_i)}{d(v_i)}
\]
**OUR TECHNICAL CONTRIBUTIONS**

- Deterministic graph traversal

\[
\sum_{\ell=0}^{\bar{\ell}} \left( \frac{p_\ell(v_i, v_i)}{d(v_i)} - \frac{p_\ell(v_j, v_i)}{d(v_i)} \right) + \left( \frac{p_\ell(v_j, v_j)}{d(v_j)} - \frac{p_\ell(v_i, v_j)}{d(v_j)} \right)
\]

rely on \( p_\ell(v_*, v_i) \) of \( v_i \),
where \( v_* \) is \( v_i \) and its neighbors

deterministic graph traversal in a reverse manner

\[
p_\ell(v_x, v_i) = \frac{p_{\ell-1}(v_j, v_i)}{d(v_x)} = \frac{p_{\ell-1}(v_j, v_i)}{3}
\]

\[
p_\ell(v_y, v_i) = \frac{p_{\ell-1}(v_j, v_i)}{d(v_y)} = p_{\ell-1}(v_j, v_i)
\]
OUR TECHNICAL CONTRIBUTIONS

- Random walk sampling

\[ \sum_{\ell = \hat{\tau} + 1}^{\tau} \frac{p_\ell(v_i, v_i)}{d(v_i)} + \frac{p_\ell(v_j, v_j)}{d(v_j)} - \frac{p_\ell(v_i, v_j)}{d(v_j)} - \frac{p_\ell(v_j, v_i)}{d(v_i)} \]

\[ \sum_{v_x} \frac{p_{\hat{\tau}}(v_x, v_i)}{d(v_i)} \left( \sum_{\ell = 1}^{\tau - \hat{\tau}} p_\ell(v_i, v_x) - p_\ell(v_j, v_x) \right) \]

all \( p_{\hat{\tau}}(v_x, v_i) \) are known by traversal

estimate by generating random walks from \( v_i \) and \( v_j \)
# EXPERIMENTS

- **Dataset statistics**

<table>
<thead>
<tr>
<th>Name</th>
<th>#nodes</th>
<th>#edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook [30]</td>
<td>4,039</td>
<td>88,234</td>
</tr>
<tr>
<td>HepPh [20]</td>
<td>34,401</td>
<td>420,784</td>
</tr>
<tr>
<td>Slashdot [22]</td>
<td>77,360</td>
<td>469,180</td>
</tr>
<tr>
<td>Twitch [35]</td>
<td>168,114</td>
<td>6,797,557</td>
</tr>
<tr>
<td>Orkut [50]</td>
<td>3,072,441</td>
<td>117,185,082</td>
</tr>
</tbody>
</table>
EXPERIMENTS

- $\epsilon$-approximate AESC solutions
  - spanning tree sampling
    - ST-Edge [IJCAI’16]
  - random walk sampling with our $\tau$
    - MonteCarlo [KDD’21]
    - MonteCarlo-C [KDD’21]
- our proposal
  - TGT: our $\tau +$ reverse graph traversal
  - TGT+: our $\tau +$ reverse graph traversal + random walk
EXPERIMENTS

- Our $\tau$ vs. Peng et al.’s $\tau$

![Graphs showing truncated lengths and the number of walks](image)
EXPERIMENTS

- running time vs. absolute error $\epsilon$
SUMMARY

- Personalized random walk length
- TGT: deterministic graph traversal in a reverse manner
- TGT+: deterministic graph traversal + random walk sampling
THANK YOU! Q&A
- Tradeoff between running time and actual average absolute error

![Graphs showing tradeoff between running time and actual average absolute error for different datasets](image)

(a) *Facebook*  
(b) *HepPh*  
(c) *Slashdot*  
(d) *Twitch*