Centrality in Large Networks

Mostafa H. Chehreghani

May 14, 2017
Table of contents

Centrality notions

Exact algorithm

Approximate algorithms

Conclusion
Centrality notions

Exact algorithm

Approximate algorithms

Conclusion
Centrality in networks

- Graphs (networks) are a powerful tool to model data in different domains
  - The world wide web, road networks, social networks, ...
- Centrality notions identify the most important vertices within a graph
- Applications:
  - key intersections in a road network
  - Influential persons in a social network
  - Ranking in Web search
    - Query independent link-based score of importance of a web page
  - ...

Centrality in networks
Types of centralities

Starting point: the central vertex of a star is the most important!
Why?

1. the vertex with largest degree;
2. the vertex that is closest to the other vertexes (e.g., that has the smallest average distance to other vertexes);
3. the vertex through which all shortest paths pass;
4. the vertex with the largest number of incoming paths of length $k$, for every $k$;
5. the vertex that maximizes the dominant eigenvector of the graph adjacency matrix;
6. the vertex with highest probability in the stationary distribution of the natural random walk on the graph.

These observations lead to corresponding competing views of centrality.
A star graph
This observation leads to the following classes of indices of centrality:

1. measures based on **distances** [degree, closeness, Lin’s index];
2. measures based on **paths** [betweenness, Katz’s index];
3. **spectral** measures [dominant eigenvector, Seeley’s index, PageRank, HITS, SALSA].

The last two classes are largely the same (even if that wasn’t fully understood for a long time.)
Geometric centralities

- **degree** (folklore): \( c_{\text{deg}}(x) = d^-(x) \)
- **closeness** (Bavelas, 1950): \( c_{\text{clos}}(x) = c(x) = \frac{1}{\sum_y d(y,x)} \)
- **Lin** (Lin, 1976): \( c_{\text{Lin}}(x) = \frac{r(x)^2}{\sum_y d(y,x)} \) where \( r(x) \) is the number of vertexes that are co-reachable from \( x \)
- **harmonic** (Boldi and Vigna, 2013) \( c_{\text{harm}}(x) = \sum_{y \neq x} \frac{1}{d(y,x)} \)
Path-based centralities

- **betweenness** (Anthonisse, 1971):
  \[ c_{\text{bet}}(x) = b(x) = \sum_{y,z \neq x, \sigma_{yz} \neq 0} \frac{\sigma_{yz}(x)}{\sigma_{yz}} \]
  where \( \sigma_{yz} \) is the number of shortest paths \( y \rightarrow z \), and \( \sigma_{yz}(x) \) is the number of such paths passing through \( x \)

- **Katz** (Katz, 1951):
  \[ c_{\text{Katz}}(x) = \sum_{t \geq 0} \beta^t p_t(x) \]
  where \( p_t(x) \) is the number of paths of length \( t \) ending in \( x \), and \( \beta \) is a parameter \( (\beta < 1/\rho) \)
Spectral centralities

- **dominant** (Wei, 1953): $c_{dom}(x)$ is the dominant (right) eigenvector of $G$
- **Seeley** (Seeley, 1949): $c_{Seeley}(x)$ is the dominant (left) eigenvector of $G_r$
- **PageRank** (Brin, Page et al., 1999): $c_{PR}(x)$ is the dominant (left) eigenvector of $\alpha G_r + (1 - \alpha) \mathbf{1}^T \mathbf{1}/n$ (where $\alpha < 1$)
- **HITS** (Kleinberg, 1997): $c_{HITS}(x)$ is the dominant (left) eigenvector of $G^T G$
- **SALSA** (Lempel, Moran, 2001): $c_{SALSA}(x)$ is the dominant (left) eigenvector of $G_c^T G_r$

Where $G$ denotes the adjacency matrix of the graph, $G_r$ is the adjacency matrix normalized by row, and $G_c$ is the adjacency matrix normalized by column.
Closeness and Betweenness
Closeness centrality

Motivation
It measures the ability to quickly access or pass information through the graph;

Definition (Closeness Centrality)

- closeness centrality \( c(x) \) of a vertex \( x \)

\[
c(x) = \frac{1}{\sum_{y \neq x \in V} d(y, x)}.
\]

- \( d(y, x) \) is the length of a shortest path between \( y \) and \( x \).
- The closeness of a vertex is defined as the inverse of the sum of the Shortest Path (SP) distances between the vertex and all other vertexes of the graph.
- When multiplied by \( n - 1 \), it is effectively the inverse of the average SP distance.
Traffic analysis

Closeness centrality of Nantes, France. Limited traffic zone in the center.
Eccentricity

- A distance-based centrality notion
- Maximum of the shortest path sizes to all other nodes
- If graph is not connected, infinite eccentricity
- **Center** of a graph: set of all nodes with **minimum** eccentricity
- **Periphery** of a graph: set of all nodes with **maximum** eccentricity
Eccentricity
Radius, diameter, and average path length

- **Diameter**: the maximum distance between any pair of vertices in the graph.
  - Maximum eccentricity
- **Radius**: minimum eccentricity
- **Average path length**:

  \[
  L = \frac{\sum_{u \in V(G)} \sum_{v \in V(G)} d(u, v)}{|V(G)| \cdot (|V(G)| - 1)}
  \]

  where \(d(u, v)\) is the size of shortest path between \(u\) and \(v\)
Average path length

- In most of real-world networks:
  \[ L \propto \log |V(G)| \]

- More precisely,
  \[ L \propto \frac{\log |V(G)|}{\log \log |V(G)|} \]

- Small-world property
  - Most vertices are not neighbors of one another, but can be reached from every other vertex by a small number of steps
### Average path length

<table>
<thead>
<tr>
<th>Network</th>
<th>Size</th>
<th>$\langle k \rangle$</th>
<th>$\ell$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WWW, site level, undir.</td>
<td>153 127</td>
<td>35.21</td>
<td>3.1</td>
</tr>
<tr>
<td>Internet, domain level</td>
<td>3015–6209</td>
<td>3.52–4.11</td>
<td>3.7–3.76</td>
</tr>
<tr>
<td>Movie actors</td>
<td>225 226</td>
<td>61</td>
<td>3.65</td>
</tr>
<tr>
<td>LANL co-authorship</td>
<td>52 909</td>
<td>9.7</td>
<td>5.9</td>
</tr>
<tr>
<td>MEDLINE co-authorship</td>
<td>1 520 251</td>
<td>18.1</td>
<td>4.6</td>
</tr>
<tr>
<td>SPIRES co-authorship</td>
<td>56 627</td>
<td>173</td>
<td>4.0</td>
</tr>
<tr>
<td>NCSTRL co-authorship</td>
<td>11 994</td>
<td>3.59</td>
<td>9.7</td>
</tr>
<tr>
<td>Math. co-authorship</td>
<td>70 975</td>
<td>3.9</td>
<td>9.5</td>
</tr>
<tr>
<td>Neurosci. co-authorship</td>
<td>209 293</td>
<td>11.5</td>
<td>6</td>
</tr>
<tr>
<td><em>E. coli</em>, substrate graph</td>
<td>282</td>
<td>7.35</td>
<td>2.9</td>
</tr>
<tr>
<td><em>E. coli</em>, reaction graph</td>
<td>315</td>
<td>28.3</td>
<td>2.62</td>
</tr>
<tr>
<td>Ythan estuary food web</td>
<td>134</td>
<td>8.7</td>
<td>2.43</td>
</tr>
<tr>
<td>Silwood Park food web</td>
<td>154</td>
<td>4.75</td>
<td>3.40</td>
</tr>
<tr>
<td>Words, co-occurrence</td>
<td>460.902</td>
<td>70.13</td>
<td>2.67</td>
</tr>
<tr>
<td>Words, synonyms</td>
<td>22 311</td>
<td>13.48</td>
<td>4.5</td>
</tr>
<tr>
<td>Power grid</td>
<td>4941</td>
<td>2.67</td>
<td>18.7</td>
</tr>
<tr>
<td><em>C. Elegans</em></td>
<td>282</td>
<td>14</td>
<td>2.65</td>
</tr>
</tbody>
</table>
Betweenness centrality

Motivation
It measures the frequency with which a user appears in a shortest path between two other users.

Definition (Betweenness centrality)

- betweenness centrality $b(x)$ of a vertex $x$:
  \[
  b(x) = \sum_{\substack{s \neq x \neq t \in V \\ s \neq t}} \frac{\sigma_{st}(x)}{\sigma_{st}}
  \]

- $\sigma_{st}$: number of SPs from $s$ to $t$
- $\sigma_{st}(x)$: how many of them pass through $x$
Betweenness centrality

- Can be defined also for edges (similarly to vertexes)
- Edges with high betweenness are known as “weak ties”
- They tend to act as bridges between two communities

The strength of weak ties (Granovetter 1973)

- Dissemination and coordination dynamics are influenced by links established to vertexes of different communities.
- The importance of these links has become more and more with the rise of social networks and professional networking platforms.
Weak ties

Bakshy et al. 2012
Weak links have a greater potential to expose links to new contacts that otherwise would not have been discovered.
Grabowicz et al. 2012

- Personal interactions are more likely to occur in internal links within communities (strong links).
- Events or new information is propagated faster by intermediate links (weak links).
Hierarchical divisive clustering by recursively removing the “weakest tie”:

1. Compute edge betweenness centrality of all edges;
2. Remove the edge with the highest betweenness centrality;
3. Repeat from 1.
Traffic analysis

Edge betweenness centrality of Nantes, France. The beltway and bridges stand out.
Comparison

Which vertex is the most central?

- for Degree Centrality:
- for Closeness Centrality:
- for Betweenness Centrality:
Comparison

Which vertex is the most central?

- for Degree Centrality: user A
- for Closeness Centrality:
- for Betweenness Centrality:
Comparison

Which vertex is the most central?

- for Degree Centrality: user A
- for Closeness Centrality: users B and C
- for Betweenness Centrality:
Comparison

Which vertex is the most central?

- for Degree Centrality: user A
- for Closeness Centrality: users B and C
- for Betweenness Centrality: user D
Visual Comparison

A  Degree Centrality
B  Closeness Centrality
C  Betweenness Centrality
Axioms for centrality (Boldi and Vigna 2013)
Assessing

Question
Is there a robust way to convince oneself that a certain centrality measure is better than another?

Answer
Axiomatization...

- ...hard axioms (characterize a centrality measure completely)
- ...soft axioms (like the $T_i$ axioms for topological spaces)
Sensitivity to size

Idea: size matters!

$S_{k,p}$ be the union of a $k$-clique and a $p$-cycle.

- if $k \to \infty$, every vertex of the clique becomes ultimately strictly more important than every vertex of the cycle
- if $p \to \infty$, every vertex of the cycle becomes ultimately strictly more important than every vertex of the clique
Idea: density matters!

\( D_{k,p} \) be made by a \( k \)-clique and a \( p \)-cycle connected by a single bidirectional bridge:

- if \( k \to \infty \), the vertex on the clique-side of the bridge becomes more important than the vertex on the cycle-side.
Score monotonicity

Adding an edge $x \rightarrow y$ strictly increases the score of $y$.

Doesn’t say anything about the score of other vertexes!
Rank monotonicity

Adding an edge $x \rightarrow y$...

- if $y$ used to dominate $z$, then the same holds after adding the edge
- if $y$ had the same score as $z$, then the same holds after adding the edge
- strict variant: if $y$ had the same score as $z$, then $y$ dominates $z$ after adding the edge
## Rank monotonicity

<table>
<thead>
<tr>
<th>Centrality</th>
<th>General</th>
<th>Monotonicity</th>
<th>Other axioms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Score</td>
<td>Rank</td>
<td>Size</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Harmonic</td>
<td>yes</td>
<td>yes*</td>
<td>yes</td>
</tr>
<tr>
<td>Degree</td>
<td>yes</td>
<td>yes*</td>
<td>yes</td>
</tr>
<tr>
<td>Katz</td>
<td>yes</td>
<td>yes*</td>
<td>yes</td>
</tr>
<tr>
<td>PageRank</td>
<td>yes</td>
<td>yes*</td>
<td>yes</td>
</tr>
<tr>
<td>Seeley</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Closeness</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Lin</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Betweenness</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Dominant</td>
<td>no</td>
<td>no</td>
<td>?</td>
</tr>
<tr>
<td>HITS</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>SALSA</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
Kendall’s $\tau$

Hollywood collaboration network

<table>
<thead>
<tr>
<th>degree</th>
<th>Katz 1/4$\lambda$</th>
<th>Katz 1/2$\lambda$</th>
<th>Katz 3/4$\lambda$</th>
<th>SALSA</th>
<th>closeness</th>
<th>harmonic</th>
<th>Lin</th>
<th>HITS</th>
<th>between</th>
<th>PR 1/4</th>
<th>PR 1/2</th>
<th>PR 3/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>0.9053</td>
<td>0.9024</td>
<td>0.9000</td>
<td>0.9114</td>
<td>0.1950</td>
<td>0.2060</td>
<td>0.2060</td>
<td>0.2853</td>
<td>0.6419</td>
<td>0.6161</td>
<td>0.5784</td>
<td></td>
</tr>
<tr>
<td>0.9053</td>
<td>1.0000</td>
<td>0.9957</td>
<td>0.9922</td>
<td>0.8141</td>
<td>0.2059</td>
<td>0.2268</td>
<td>0.2265</td>
<td>0.2773</td>
<td>0.5917</td>
<td>0.5820</td>
<td>0.5595</td>
<td></td>
</tr>
<tr>
<td>0.9024</td>
<td>0.9957</td>
<td>1.0000</td>
<td>0.9966</td>
<td>0.8112</td>
<td>0.2078</td>
<td>0.2289</td>
<td>0.2286</td>
<td>0.2776</td>
<td>0.5914</td>
<td>0.5827</td>
<td>0.5611</td>
<td></td>
</tr>
<tr>
<td>0.9000</td>
<td>0.9922</td>
<td>0.9966</td>
<td>1.0000</td>
<td>0.8089</td>
<td>0.2094</td>
<td>0.2307</td>
<td>0.2303</td>
<td>0.2778</td>
<td>0.5911</td>
<td>0.5832</td>
<td>0.5622</td>
<td></td>
</tr>
<tr>
<td>0.9114</td>
<td>0.8141</td>
<td>0.8112</td>
<td>0.8089</td>
<td>1.0000</td>
<td>0.1782</td>
<td>0.1617</td>
<td>0.1619</td>
<td>0.1917</td>
<td>0.6145</td>
<td>0.6146</td>
<td>0.5717</td>
<td></td>
</tr>
<tr>
<td>closeness</td>
<td>0.1950</td>
<td>0.2059</td>
<td>0.2078</td>
<td>0.2094</td>
<td>0.1782</td>
<td>1.0000</td>
<td>0.8592</td>
<td>0.8566</td>
<td>0.3817</td>
<td>0.1518</td>
<td>0.1746</td>
<td>0.2004</td>
</tr>
<tr>
<td>harmonic</td>
<td>0.2060</td>
<td>0.2268</td>
<td>0.2289</td>
<td>0.2307</td>
<td>0.1617</td>
<td>0.8592</td>
<td>1.0000</td>
<td>0.9694</td>
<td>0.4253</td>
<td>0.1503</td>
<td>0.1770</td>
<td>0.2072</td>
</tr>
<tr>
<td>Lin</td>
<td>0.2060</td>
<td>0.2265</td>
<td>0.2286</td>
<td>0.2303</td>
<td>0.1619</td>
<td>0.8566</td>
<td>0.9694</td>
<td>1.0000</td>
<td>0.4272</td>
<td>0.1503</td>
<td>0.1768</td>
<td>0.2069</td>
</tr>
<tr>
<td>HITS</td>
<td>0.2853</td>
<td>0.2773</td>
<td>0.2776</td>
<td>0.2778</td>
<td>0.1917</td>
<td>0.3817</td>
<td>0.4253</td>
<td>0.4272</td>
<td>1.0000</td>
<td>0.1529</td>
<td>0.1484</td>
<td>0.1415</td>
</tr>
<tr>
<td>PR 1/4</td>
<td>0.6449</td>
<td>0.5917</td>
<td>0.5914</td>
<td>0.5911</td>
<td>0.6445</td>
<td>0.1518</td>
<td>0.1503</td>
<td>0.1503</td>
<td>0.1529</td>
<td>1.0000</td>
<td>0.9182</td>
<td>0.8289</td>
</tr>
<tr>
<td>PR 1/2</td>
<td>0.6161</td>
<td>0.5820</td>
<td>0.5827</td>
<td>0.5832</td>
<td>0.6146</td>
<td>0.1746</td>
<td>0.1770</td>
<td>0.1768</td>
<td>0.1484</td>
<td>0.9182</td>
<td>1.0000</td>
<td>0.9088</td>
</tr>
<tr>
<td>PR 3/4</td>
<td>0.5784</td>
<td>0.5595</td>
<td>0.5611</td>
<td>0.5622</td>
<td>0.5747</td>
<td>0.2004</td>
<td>0.2072</td>
<td>0.2069</td>
<td>0.1415</td>
<td>0.8289</td>
<td>0.9088</td>
<td>1.0000</td>
</tr>
</tbody>
</table>
Correlation

- most geometric indices and HITS are rather correlated to one another;
- Katz, degree and SALSA are also highly correlated;
- PageRank stands alone in the first dataset, but it is correlated to degree, Katz, and SALSA in the second dataset;
- Betweenness is not correlated to anything in the first dataset, and could not be computed in the second dataset due to the size of the graph (106M vertices).
Centrality notions

Exact algorithm

Approximate algorithms

Conclusion
In this section (and also in the next section), we mainly focus on *betweenness centrality computation*!

- Many of these techniques can be adapted for other notions such as closeness centrality
Betweenness centrality

- SPD rooted at vertex $s$
  - The directed acyclic graph (DAG) that contains all shortest paths starting from a vertex $s$

- Dependency score of vertex $s$ on vertex $v$:
  \[
  \delta_{s\rightarrow}(v) = \sum_{t \in V(G) \setminus \{v,s\}} \frac{\sigma_{st}(v)}{\sigma_{st}}
  \]

- We have:
  \[
  BC(v) = \sum_{s \in V(G) \setminus \{v\}} \delta_{s\rightarrow}(v)
  \]
An efficient algorithm for computing betweenness centrality of all vertices of a graph

It is based on the following recursive relation:

\[ \delta_{S\bullet}(v) = \sum_{w: v \in P_s(w)} \frac{\sigma_{sv}}{\sigma_{sw}} \cdot (1 + \delta_{S\bullet}(w)) \]
For each vertex \( v \), \( BC(v) \leftarrow 0 \)

For each vertex \( s \) in the graph

1. Form the SPD rooted at \( s \)
   - This is done by running BFS or Dijkstra's algorithm from \( s \) and storing \( \sigma_{sv} \) and \( P_s(v) \) for each vertex \( v \) and storing vertices in a queue \( Q \)

2. For each \( v \in V(G) \), \( \delta_s(v) \leftarrow 0 \)

3. While \( Q \) is not empty
   - \( w \leftarrow \) pop from \( Q \) the vertex with the largest distance from \( s \)
   - For each \( v \in P_s(w) \), update \( \delta_s(v) \) by
     \[
     \delta_s(v) + \frac{\sigma_{sv}}{\sigma_{sw}} (1 + \delta_s(w))
     \]
   - \( BC(w) \leftarrow BC(w) + \delta_s(w) \)
Brandes 2001

- Time complexity of the algorithm:
  - Unweighted graphs:
    \[ \Theta(|V(G)| \cdot |E(G)|) \]
  - Weighted graphs with positive weights:
    \[ \Theta(|V(G)| \cdot |E(G)| + |V(G)|^2 \log |V(G)|) \]
  - For weighted graphs where negative weights is allowed, the problem is NP-hard
    - By reducing to the longest shortest path and Hamiltonian path problems
Brandes 2001
Centrality notions

Exact algorithm

Approximate algorithms

Conclusion
Why approximation algorithms

1. Even though there are *polynomial* time and space algorithms for exact betweenness centrality computation, they are not tractable in practice
   - For large networks, *polynomial* does not imply efficiency!

2. In many applications, we only need to compute betweenness of a few vertices
   - For exact algorithms, this is not easier than computing betweenness of all vertices!

3. In many applications, we only need to have the *ratio* of betweenness scores
   - For exact algorithms, this is not easier than computing betweenness scores!
A (source) vertex sampling algorithm

For each vertex $v$, $\bar{BC}(v) \leftarrow 0$

Let $k$ be the number of samples

At each iteration $i$

- Choose a vertex $s$ uniformly at random
- Form SPD rooted at $s$ and compute dependency scores of $s$ on all other vertices
- For each vertex $v$, update $\bar{BC}(v)$ by $\bar{BC}(v) + \frac{|V(G)|}{k} \delta_{s•}(v)$
The number of samples \((k)\)

- If \(k\) is chosen such that

\[
k \geq \frac{1}{2\epsilon^2} \left( \ln |V(G)| + \ln 2 + \ln \frac{1}{\delta} \right)
\]

then the algorithm gives a \((\epsilon, \delta)\)-approximation

- This means

\[
\Pr \left( \exists v \in V(G) : |\bar{B}C(v) - BC(v)| > \epsilon \right) \leq \delta
\]

- Proof is done using Hoeffding inequality
In vertex sampling, betweenness centrality of vertices that are close to the sampled source vertices is over-estimated.

One solution: prevent vertices from profiting for being near a sampled vertex.

(Geisberger et al., 2008)
At each iteration:

- Choose uniformly at random a pair \( s, d \), where
  - \( s \) is a vertex
  - \( d \) is direction and can be forward or backward
- If \( d \) is forward, compute dependency scores of \( s \) on all other vertices
- If \( d \) is backward, if directed graph, virtually flip direction of edges and compute dependency scores of \( s \) on all other vertices
- The way dependency scores are aggregated should be revised!
(Geisberger et.al., 2008)
The generic framework to estimate $BC(v)$:

1. First the following probabilities are computed

\[ p_1, p_2, \ldots, p_n \geq 0 \text{ such that } \sum_{i=1}^{n} p_i = 1 \]

2. Then, at each iteration:
   - A vertex $i$ is selected with probability $p_i$,
   - The SPD rooted at $i$ is computed,
   - Dependency score of vertex $v$ on $i$ is computed,
   - $\frac{\delta_i \cdot (v)}{p_i}$ is the estimation of $BC(v)$ in the current iteration
Optimal sampling:

\[ p_i = \frac{\delta \cdot i(v)}{\sum_{j=1}^{n} \delta \cdot j(v)} \]

The condition that a promising sampling algorithm for estimating betweenness centrality of vertex \( v \) should satisfy:

\[ \forall i, i' \in V(G) \setminus \{v\} : p_i < p_{i'} \iff \delta \cdot i(v) < \delta \cdot i'(v) \]

Distance-based sampling:

\[ p_i = \frac{\frac{1}{d(i,v)}}{\sum_{j=1}^{n} \frac{1}{d(j,v)}} \]
<table>
<thead>
<tr>
<th>Database</th>
<th>Exact BC</th>
<th>Approximate BC</th>
<th>Distance-based sampling</th>
<th>Uniform sampling</th>
<th>Avg. time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg. BC score</td>
<td>Avg. time</td>
<td>Avg. BC score</td>
<td>Avg. time</td>
<td>Avg. error (%)</td>
</tr>
<tr>
<td>BA10^3</td>
<td>3503.83</td>
<td>7.87</td>
<td>41.77%</td>
<td>0.2</td>
<td>56.13%</td>
</tr>
<tr>
<td>Ba10^4</td>
<td>358789.85</td>
<td>743.49</td>
<td>16.54%</td>
<td>0.5</td>
<td>29.79%</td>
</tr>
<tr>
<td>Wiki Vote</td>
<td>76056.85</td>
<td>515.09</td>
<td>37.0%</td>
<td>0.5</td>
<td>41.13%</td>
</tr>
<tr>
<td>Email-Enron</td>
<td>2775100.8</td>
<td>9033.11</td>
<td>15.75%</td>
<td>0.5</td>
<td>25.28%</td>
</tr>
<tr>
<td>dblp0305</td>
<td>564246.41</td>
<td>19149.8</td>
<td>7.59%</td>
<td>1.5</td>
<td>64.73%</td>
</tr>
<tr>
<td>dblp0507</td>
<td>798125.00</td>
<td>35140</td>
<td>7.19%</td>
<td>1.5</td>
<td>50.17%</td>
</tr>
<tr>
<td>CA-CondMat</td>
<td>691667</td>
<td>3026.9</td>
<td>10.8%</td>
<td>1</td>
<td>20.81%</td>
</tr>
<tr>
<td>CA-HepTh</td>
<td>23747.85</td>
<td>341.8</td>
<td>26.99%</td>
<td>0.4</td>
<td>32.18%</td>
</tr>
</tbody>
</table>
The sample size $k$ should depend on a specific characteristic quantity of the graph (not just on the number of vertices).

For all $v \in V(G)$, $\bar{BC}(v) \leftarrow 0$

At each iteration:

- Choose a pair of vertices $s, t$ uniformly at random
- $S_{st} \leftarrow$ all shortest paths from $s$ to $t$
- $p \leftarrow$ a shortest path chosen from $S_{st}$ uniformly at random
- For each internal vertex $w$ of $p$: $\bar{BC}(w) \leftarrow \bar{BC}(w) + \frac{1}{k}$
The number of samples (iterations) $k$:

$$
\frac{1}{2\epsilon^2} \left( \lceil \log_2 (VD(G) - 2) \rceil + 1 + \ln\left(\frac{1}{\delta}\right) \right)
$$

- $VD(G)$ is the longest shortest path of $G$
  - Called vertex diameter
RK (Riondato and Kornaropoulos 2015)

email-Enron-u, $|V|=36,692$, $|E|=367,662$, $\delta = 0.1$, runs = 5

8 times faster than BP and 4 – 200 times faster than exact algorithm!
email-Enron-u, |V|=36,692, |E|=367,662, δ=0.1, runs= 5

**Absolute estimation error**

**epsilon**

- Avg (diam-2approx)
- Avg+Stddev (diam-2approx)
- Max (diam-2approx)
Weaknesses of RK:

1. For each pair of vertices sampled, it only uses one of the shortest paths
   - Waste of computation!
2. To determine the number of samples, it needs to compute $VD(G)$
   - Computing $VD(G)$ is as expensive as computing betweenness centrality
   - Approximate computation of $VD(G)$ can be done efficiently. However, it may yield a larger-than-necessary sample size
The ABRA algorithm tries to address these issues!

1. Instead of choosing one random shortest path, consider all shortest paths between \( u \) and \( v \)
   - Increase the estimation of any vertex that is on a shortest path between \( u \) and \( v \)

2. The number of samples is determined using an application of Rademacher Averages:
   - It depends on the richness of the vectors representing the current estimations of the betweenness scores of the vertices
ABRA (Riondato and Upfal 2016)

<table>
<thead>
<tr>
<th>Graph</th>
<th>$\varepsilon$</th>
<th>Runtime (sec.)</th>
<th>BA</th>
<th>RK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soc-Epinions1</td>
<td>0.005</td>
<td>483.06</td>
<td>1.36</td>
<td>2.90</td>
</tr>
<tr>
<td>Directed</td>
<td>0.010</td>
<td>124.60</td>
<td>5.28</td>
<td>3.31</td>
</tr>
<tr>
<td>$</td>
<td>V</td>
<td>= 75,879$</td>
<td>0.015</td>
<td>57.16</td>
</tr>
<tr>
<td>$</td>
<td>E</td>
<td>= 508,837$</td>
<td>0.020</td>
<td>32.90</td>
</tr>
<tr>
<td>$</td>
<td>V</td>
<td>= 36,682$</td>
<td>0.025</td>
<td>21.88</td>
</tr>
<tr>
<td></td>
<td>0.030</td>
<td>16.05</td>
<td>40.95</td>
<td>7.52</td>
</tr>
<tr>
<td>Email-Enron</td>
<td>0.010</td>
<td>202.43</td>
<td>1.18</td>
<td>1.10</td>
</tr>
<tr>
<td>Undirected</td>
<td>0.015</td>
<td>91.36</td>
<td>2.63</td>
<td>1.09</td>
</tr>
<tr>
<td>$</td>
<td>V</td>
<td>= 183,831$</td>
<td>0.020</td>
<td>53.50</td>
</tr>
<tr>
<td>$</td>
<td>E</td>
<td>= 183,831$</td>
<td>0.025</td>
<td>31.99</td>
</tr>
<tr>
<td></td>
<td>0.030</td>
<td>24.06</td>
<td>9.97</td>
<td>1.03</td>
</tr>
</tbody>
</table>

Much faster than RK:

- Using less samples
- No need to compute vertex-diameter
More than 10 times more accurate than guaranteed, on average

More than 100 times more accurate than guaranteed, in the best case

Close to the guarantee in the worst case
The basic idea is simple

- Sample shortest paths $\pi_1, \ldots, \pi_\tau$
- $\pi_i$ is chosen by selecting uniformly at random two vertices $s, t$ and then, selecting uniformly at random one of the shortest paths from $s$ to $t$
- Estimate $BC(v)$ with $\bar{BC}(v) = \frac{1}{\tau} \sum_{i=1}^{\tau} X_i(v)$, where
  - If $v \in \pi_i$, $X_i(v) = 1$
  - Otherwise, $X_i(v) = 0$
- $\mathbb{E}[\bar{BC}(v)] = BC(v)$

The key improvement is using balanced bidirectional BFS for sampling random paths
KADABRA (Borassi and Natale 2016)

- **Balanced Bidirectional BFS:**
  - Instead of performing a full BFS from $s$ until we reach $t$, we perform at the same time a BFS from $s$ and a BFS from $t$, until the two BFSs touch each other
  - Assume that we have visited up to level $l_s$ from $s$ and to level $l_t$ from $t$
  - If sum of the degrees of vertices at level $l_s$ is less than sum of the degrees of vertices at level $l_t$, we proceed with the BFS rooted at $s$
    - Otherwise, proceed with the BFS rooted at $t$
Why Balanced Bidirectional BFS is more efficient:

- $d$: the distance between $s$ and $t$
- $b$: maximum degree
- BFS will traverse $\Theta(b^d)$ vertices
- Balanced bidirectional BFS will traverse $\Theta(b^{d/2})$ vertices
- For large $b$ and $d$, balanced bidirectional BFS is much faster!
KADABRA (Borassi and Natale 2016)
Centrality notions

Exact algorithm

Approximate algorithms

Conclusion
In this talk, we presented:

- A brief survey on centrality notions
  - Betweenness centrality, closeness centrality, degree centrality, spectral centrality, ...
- A brief survey on the algorithms of betweenness centrality
- Brandes exact algorithm
  - Its time complexity is polynomial
    - However, it is not tractable in practice
    - For big data, polynomial does not mean efficiency!
- Several approximation algorithms
  - Much faster than exact algorithm
  - Usually have high accuracy
  - More suitable for big data!
Any question?

Some of the slides are from the tutorial of Bonchi et.al. at WWW’16!

Email: mostafa.chehreghani@gmail.com