EFFECTIVE AND EFFICIENT PAGERANK-BASED POSITIONING FOR GRAPH VISUALIZATION

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

June 2023
OUTLINE

- Background
- Existing Solutions
- PPRviz
- Experiments
- Conclusion
BACKGROUND: GRAPH VISUALIZATION

- Input: a graph $G$ with $n$ nodes and $m$ edges
- Output: a 2D position matrix $X$
- Drawing:
  - Position each node $v_i$ at its coordinate $X[i]$
  - Link two endpoints of each edge with a straight segment
- It helps to understand relational data
BACKGROUND: AESTHETIC CRITERIA

- An effective visualization should have good readability
- Evaluate the readability of $X$ by aesthetic criteria
- Node Distribution (ND): measure the distribution evenness of the nodes on the screen
- Uniform Length Coefficient Variance (ULCV): measure the length skewness of edge segments on the screen
EXISTING SINGLE-LEVEL SOLUTIONS

- Idea:
  - visualize all nodes and edges on the screen

- Steps:
  - compute a graph-theoretical distance matrix $D$:
    - adjacency-related matrix or the shortest distance matrix
  - embed $D$ into $X$:
    - minimize node pair’s difference between graph and Euclidean distance

- Cons:
  - Poor readability: aesthetically-unpleasing or hairball-like layout
  - Expensive computational cost
EXISTING MULTI-LEVEL FRAMEWORK

- Idea:
  - interactively show the partial view level by level

- Steps:
  - build a supergraph hierarchy \( H \) for \( G \)
  - use a single-level solution to visualize children in \( S \)

- Pros:
  - avoid hairball
  - reduce embedding overhead

- Cons:
  - the aesthetic issue remains
PPRVIZ: OVERVIEW

- Supergraph hierarchy construction:
  - Generate $H$ by Louvain [a] with balanced size

- Node distance computation:
  - Design a new distance measure $\text{PDist}$
  - Compute $\text{PDist}$ matrix $\mathbf{\Delta}$ for children in $S$ by our Tau-Push

- Position embedding:
  - Compute $\mathbf{X}$ by $\mathbf{\Delta}$
  - Make node pair’s Euclidean distance resemble its $\text{PDist}$

**PPRVIZ: PDIST FOR LEAF NODES**

- **Personalized PageRank (PPR)**
  - Input: a source $v_s$, a target $v_t$, and a stopping probability $\alpha$
  - Random walk with restart (RWR) from $v_s$:
    - At each step, stops with probability $\alpha$ at the current node,
    - With $1 - \alpha$ probability randomly jumps to one of the neighbors
  - PPR from $v_s$ to $v_t$: $\pi(v_s, v_t) = \mathbb{P}[\text{RWR from } v_s \text{ stops at } v_t]$

<table>
<thead>
<tr>
<th></th>
<th>$\pi(v_0, v_8)$</th>
<th>$\pi(v_2, v_0)$</th>
<th>$\pi(v_6, v_9)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.01</td>
<td>0.11</td>
<td>0.44</td>
</tr>
</tbody>
</table>

A large $\pi(v_s, v_t)$ indicates $v_s$ and $v_t$ are well-connected, which should be close on graph and screen.
PPRVIZ: PDIST FOR LEAF NODES

- PDist between any nodes $v_i, v_j$:
  - Degree-normalized PPR (DPPR): $\pi_d(v_i, v_j) = \pi(v_i, v_j) \cdot d(v_i)$
  - Convert DPPR to a distance: $1 - \log(\pi_d(v_i, v_j) + \pi_d(v_j, v_i))$

- Pros:
  - Preserve high-order information
  - Guarantee visualization quality in terms of ND and ULCV

![Graph showing RWR and ignored paths]

RWR from $v_3$ to $v_1$:
- $v_3 \rightarrow v_1$
- $v_3 \rightarrow v_4 \rightarrow v_1$
- $v_3 \rightarrow v_2 \rightarrow v_1$
- $v_3 \rightarrow v_5 \rightarrow v_4 \rightarrow v_1$
- ...
PPRviz: Tau-Push for Leaf Nodes

- **Tau-Push**
  - Compute the tau value $\tau_j$ for each $v_j$ and compute a constant $\tau$, where
    $$\tau_j = \frac{1}{m} \cdot \sum_i \pi_d(v_i, v_j)$$
  - Estimate $\Delta[i, j]$ for $v_j$ with $\tau_j < \tau$ by a deterministic version of RWR from $v_i$
  - Estimate $\Delta[i, j]$ for $v_j$ with $\tau_j \geq \tau$ by a reverse traversal from $v_j$

precompute and store as index
## EXPERIMENTS: DATASETS

<table>
<thead>
<tr>
<th>Dataset</th>
<th>( n )</th>
<th>( m )</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TwEgo</td>
<td>23</td>
<td>52</td>
<td>Ego network</td>
</tr>
<tr>
<td>FbEgo</td>
<td>52</td>
<td>146</td>
<td>Ego network</td>
</tr>
<tr>
<td>Wiki-ii</td>
<td>186</td>
<td>632</td>
<td>Authorship network</td>
</tr>
<tr>
<td>Physician</td>
<td>241</td>
<td>1.8K</td>
<td>Social network</td>
</tr>
<tr>
<td>FilmTrust</td>
<td>874</td>
<td>2.6K</td>
<td>User trust network</td>
</tr>
<tr>
<td>SciNet</td>
<td>1.5K</td>
<td>5.4K</td>
<td>Collaboration network</td>
</tr>
<tr>
<td>Amazon</td>
<td>334.9K</td>
<td>1.9M</td>
<td>Product network</td>
</tr>
<tr>
<td>Youtube</td>
<td>1.1M</td>
<td>6.0M</td>
<td>Social network</td>
</tr>
<tr>
<td>Orkut</td>
<td>3.1M</td>
<td>234.4M</td>
<td>Social network</td>
</tr>
<tr>
<td>DBLP</td>
<td>5.4M</td>
<td>17.2M</td>
<td>Collaboration network</td>
</tr>
<tr>
<td>It-2004</td>
<td>41.3M</td>
<td>2.3B</td>
<td>Crawled network</td>
</tr>
<tr>
<td>Twitter</td>
<td>41.7M</td>
<td>3.0B</td>
<td>Social network</td>
</tr>
</tbody>
</table>

Dataset statistics (\( K = 10^3, M = 10^6, B = 10^9 \))
EXPERIMENTS: COMPETITORS

- Single-level competitors
  - Stress methods: CMDS, PMDS
  - Node embedding methods: GFactor, SDNE, LapEig, LLE, Node2vec
  - A variant replacing DPPR in PDist with SimRank

- Multi-level competitors
  - OpenOrd, KDraw

- Most competitors have been applied in software and libraries like Gephi, Graphviz and NetworkX.
EXPERIMENTS: EFFECTIVENESS OF PPRVIZ

- 6 small datasets and 11 single-level competitors
- ULCV: the smaller the better

<table>
<thead>
<tr>
<th></th>
<th>TwEgo</th>
<th>FbEgo</th>
<th>Wiki-ii</th>
<th>Physician</th>
<th>FilmTrust</th>
<th>SciNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPRviz</td>
<td>0.22</td>
<td>0.39</td>
<td>0.35</td>
<td>0.45</td>
<td>0.48</td>
<td>0.34</td>
</tr>
<tr>
<td>FR</td>
<td>0.35</td>
<td>0.42</td>
<td>0.41</td>
<td>0.53</td>
<td>0.54</td>
<td>0.77</td>
</tr>
<tr>
<td>LinLog</td>
<td>0.57</td>
<td>0.67</td>
<td>1.09</td>
<td>0.90</td>
<td>1.99</td>
<td>4.70</td>
</tr>
<tr>
<td>ForceAtlas</td>
<td>0.37</td>
<td>0.49</td>
<td>0.64</td>
<td>0.55</td>
<td>0.96</td>
<td>1.52</td>
</tr>
<tr>
<td>CMDS</td>
<td>0.40</td>
<td>0.46</td>
<td>0.62</td>
<td>0.80</td>
<td>1.05</td>
<td>1.74</td>
</tr>
<tr>
<td>PMDS</td>
<td>0.23</td>
<td>0.45</td>
<td>0.78</td>
<td>0.47</td>
<td>0.69</td>
<td>0.74</td>
</tr>
<tr>
<td>GFactor</td>
<td>0.45</td>
<td>0.91</td>
<td>0.62</td>
<td>0.95</td>
<td>0.64</td>
<td>0.86</td>
</tr>
<tr>
<td>SDNE</td>
<td>1.96</td>
<td>0.94</td>
<td>0.94</td>
<td>1.67</td>
<td>1.31</td>
<td>1.72</td>
</tr>
<tr>
<td>LapEig</td>
<td>1.15</td>
<td>0.98</td>
<td>1.04</td>
<td>1.02</td>
<td>1.70</td>
<td>1.26</td>
</tr>
<tr>
<td>LLE</td>
<td>0.46</td>
<td>0.77</td>
<td>1.27</td>
<td>0.77</td>
<td>0.87</td>
<td>-</td>
</tr>
<tr>
<td>Node2vec</td>
<td>0.80</td>
<td>0.96</td>
<td>0.86</td>
<td>1.41</td>
<td>0.89</td>
<td>1.32</td>
</tr>
<tr>
<td>SimRank</td>
<td>0.84</td>
<td>0.75</td>
<td>0.53</td>
<td>0.53</td>
<td>1.78</td>
<td>1.98</td>
</tr>
</tbody>
</table>
EXPERIMENTS: EFFECTIVENESS OF PPRVIZ

- The best competitor FR (in terms of aesthetic criteria)
- Visualizations on FilmTrust
Experiments: Efficiency of PPRViz

Preprocessing time:
- compute $H$ and index of Tau-Push in PPRviz
- compute $H$ in multi-level methods

Response time:
- visualize $S$ in PPRviz and multi-level methods
- visualize $G$ in single-level methods
CONCLUSION

- PPRviz: graph visualization solution
- PDist: PPR-based distance measure
- Tau-Push: efficient PDist approximation algorithm
THANK YOU! Q&A
### BACKUP: TAU-PUSH FOR LEAF NODES

- **Forward Push [a]**
  - Deterministic version of RWR
  - Given a source $v_s$, each node $v_i$ maintains
    - estimation $\hat{\pi}_d(v_s, v_i)$ and residue $r(v_s, v_i)$
  - Invariant:
    \[
    \pi_d(v_s, v_t) = \hat{\pi}_d(v_s, v_t) + \sum_i \frac{1}{d(v_i)} \cdot r(v_s, v_i) \cdot \pi_d(v_i, v_t)
    \]

---

DPR-guided termination

Degree-normalized PageRank (DPR) for $v_j$:

$$\tau_j = \frac{1}{m} \cdot \sum_i \pi_d(v_i, v_j)$$

Given a source $v_s$ and a target $v_t$, stop Forward Push when each

$$r(v_s, v_i) \leq \frac{\epsilon \cdot \delta}{m \cdot \tau_t}$$

$\hat{\pi}_d(v_s, v_t)$ is $(\epsilon, \delta)$-approximate, since

$$\pi_d(v_s, v_t) - \hat{\pi}_d(v_s, v_t) = \sum_i \frac{1}{d(v_i)} \cdot r(v_s, v_i) \cdot \pi_d(v_i, v_t) \leq \epsilon \cdot \delta$$
Refinement

Intuition:

- only using Forward Push incurs redundant overhead
- Backward Push [a]:
  - perform push operations from a target $v_t$ in a reverse manner

For this $v_t$, the stop condition is extremely tough even if the estimations of others in $S$ are good

BACKUP: TAU-PUSH FOR LEAF NODES

- Summary
  - DPR Computation and identify large-DPR $v_t$
  - Forward Push: estimate for most small-DPR $v_t$
  - Backward Push: estimate for large-DPR $v_t$

<table>
<thead>
<tr>
<th>$\tau_1$</th>
<th>0.01</th>
<th>$\tau_7$</th>
<th>0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_2$</td>
<td>0.03</td>
<td>$\tau_8$</td>
<td>0.02</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>0.02</td>
<td>$\tau_9$</td>
<td>0.06</td>
</tr>
<tr>
<td>$\tau_4$</td>
<td>0.01</td>
<td>$\tau_{10}$</td>
<td>0.07</td>
</tr>
<tr>
<td>$\tau_5$</td>
<td>0.04</td>
<td>$\tau_{11}$</td>
<td>0.02</td>
</tr>
<tr>
<td>$\tau_6$</td>
<td>0.02</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$\tau = 0.05$

precompute and store as index