Fast Incremental von Neumann Graph Entropy Computation: Theory, Algorithm, and Applications

Pin-Yu Chen
IBM Research AI

joint work with
Lingfei Wu (IBM Research AI)
Sijia Liu (IBM Research AI)
Indika Rajapakse (Univ. Michigan Ann Arbor)

Poster: Tuesday 6:30-9:00 pm, Pacific Ballroom #265

June 10, 2019
Graph as a Data Representation

Social Network

Power Grid

Communication Network

Information System

Bio Informatics

Cyber-Physical System
Information-Theoretic Measures between Graphs

- Structural reducibility of multilayer networks (unsupervised learning)
Von Neumann Graph Entropy (VNGE): Introduction

- Quantum information theory: $\Phi$ is a $n \times n$ density matrix that is symmetric, positive semidefinite, and $\text{trace}(\Phi) = 1$
  \[ \{\lambda_i\}_{i=1}^{n} : \text{eigenvalues of } \Phi \]

- Von Neumann entropy $H = -\text{trace}(\Phi \ln \Phi) = - \sum_{i: \lambda_i > 0} \lambda_i \ln \lambda_i$
  \[ \rightarrow \text{Shannon entropy over eigenspectrum } \{\lambda_i\}_{i=1}^{n}, \text{ since } \sum_i \lambda_i = 1 \]
  \[ \Rightarrow \text{Generally requires } O(n^3) \text{ computation complexity for } H \]

- Graph $G = (\mathcal{V}, \mathcal{E}, W) \in G$: undirected weighted graphs with nonnegative edge weights. $G$ has $|\mathcal{V}| = n$ nodes and $|\mathcal{E}| = m$ edges.

  \[ D = \text{diag}(\{\lambda_i\}) : \text{diagonal degree matrix. } [W]_{ij} = w_{ij} : \text{edge weight.} \]

- Von Neumann graph entropy (VNGE): $\Phi = L_N = c \cdot L$, where
  \[ c = \frac{1}{\text{trace}(L)} = \frac{1}{\sum_{i \in \mathcal{V}} d_i} = \frac{1}{2 \sum_{(i,j) \in \mathcal{E}} w_{ij}} \]

- $H \leq \ln(n - 1)$, “=” when $G$ is a complete graph with identical edge weight

Von Neumann Graph Entropy (VNGE): Introduction

- VNGE characterizes structural complexity of a graph and enables computation of Jensen-Shannon distance (JSdist) between graphs.
- Applications in network learning, computer vision and data science:
  1. Structural reducibility of multilayer networks (hierarchical clustering)
  2. Depth-analysis for image processing
  3. Network-ensemble comparison via edge rewiring
  4. Structure-function analysis in genetic networks
- High consistency with classical Shannon graph entropy that is defined as a probability distribution of a function on subgraphs of $G$.

The main challenge of exact VNGE computation: it generally requires cubic complexity $O(n^3)$ for obtaining the full eigenspectrum → NOT scalable to large graphs

Our solution: FINGER, a scalable and provably asymptotically correct approximate computation framework of VNGE

FINGER supports two different data modes: batch and online

(a) Batch mode: $O(n + m)$

(b) Online mode: $O(\Delta n + \Delta m)$

New applications:

1. Anomaly detection in evolving Wikipedia hyperlink networks
2. Bifurcation detection of cellular networks during cell reprogramming
3. Synthesized denial of service attack detection in router networks
Efficient VNGE Computation via FINGER

- Recall $H = - \sum_{i=1}^{n} \lambda_i \ln \lambda_i \Rightarrow O(n^3)$ cubic complexity
- FINGER enables fast and incremental computation of $H$ with asymptotic approximation guarantee

Lemma (Quadratic approximation of $H$)

The quadratic approximation of the von Neumann graph entropy $H$ via Taylor expansion is equivalent to $Q = 1 - c^2 (\sum_{i \in V} d_i^2 + 2 \cdot \sum_{(i,j) \in E} w_{ij}^2)$

- $d_i$: degree (sum of edge weights) of node $i$
- $w_{ij}$: edge weight of edge $(i, j)$
- $c = \frac{1}{2 \sum_{(i,j) \in E} w_{ij}}$
- $O(n + m)$ linear complexity. $|V| = n$, $|E| = m$
- $Q$ can be incremental updated given graph changes $\Delta G$ $\Rightarrow O(\Delta n + \Delta m)$ complexity
Approximate VNGE with Asymptotic Guarantees

- Let $\lambda_{\text{max}}$ ($\lambda_{\text{min}}$) be the largest (smallest) positive eigenvalue in $\{\lambda_i\}$
- Approx. VNGE for batch graph sequence: $\hat{H}(G) = -Q \ln \lambda_{\text{max}}$
- Approx. VNGE for online graph sequence: $\tilde{H}(G) = -Q \ln (2c \cdot d_{\text{max}})$
- Relation: $\tilde{H} \leq \hat{H} \leq H$

### Theorem ($o(\ln n)$ approximation error with balanced eigenspectrum)

If the number of positive eigenvalues $n_+ = \Omega(n)$ and $\lambda_{\text{min}} = \Omega(\lambda_{\text{max}})$, the scaled approximation error (SAE) $\frac{H - \hat{H}}{\ln n} \to 0$ and $\frac{H - \tilde{H}}{\ln n} \to 0$ as $n \to \infty$.

$f(n) = o(h(n))$ and $f(n) = \Omega(h(n))$ mean $\lim_{n \to \infty} \frac{f(n)}{h(n)} = 0$, and $\limsup_{n \to \infty} \left| \frac{f(n)}{h(n)} \right| > 0$, respectively.

- Computing $\lambda_{\text{max}}$ only requires $O(n + m)$ operations via power iteration $\Rightarrow O(n + m)$ linear complexity for $\hat{H}$.

### Theorem (Incremental update of $\tilde{H}$ with $O(\Delta n + \Delta m)$ complexity)

The VNGE $\tilde{H}(G \oplus \Delta G)$ can be updated by $\tilde{H}(G \oplus \Delta G) = F(\tilde{H}(G), \Delta G)$
Figure: Scaled approximation error (SAE) and computation time reduction ratio

- scaled approximation error (SAE) = \( \frac{H - H_{\text{approx}}}{\ln n} \)
- computation time reduction ratio = \( \frac{\text{Time}_H - \text{Time}_{H_{\text{approx}}}}{\text{Time}_H} \)
- almost 100% speed-up (\(O(n^3)\) v.s. \(O(n + m)\))
- approximation error decreases as average degree increases
- regular (random) graphs have smaller (larger) approximation error
Two graphs $G$ and $\tilde{G}$ of the same node set $\mathcal{V}$.

KL divergence $D_{KL}(G|\tilde{G}) = \text{trace}(L_N(G) \cdot [\ln L_N(G) - \ln L_N(\tilde{G})])$ (not symmetric)

Let $\overline{G} = \frac{G \oplus \tilde{G}}{2}$ denote the averaged graph of $G$ and $\tilde{G}$, where $L_N(\overline{G}) = \frac{L_N(G) + L_N(\tilde{G})}{2}$.

The Jensen-Shannon divergence is defined as $\text{DIV}_{JS}(G, \tilde{G}) = \frac{1}{2} D_{KL}(G|\tilde{G}) + \frac{1}{2} D_{KL}(\tilde{G}|G) = H(\overline{G}) - \frac{1}{2} [H(G) + H(\tilde{G})]$ (symmetric)

The Jensen-Shannon distance is defined as $\text{JSdist}(G, \tilde{G}) = \sqrt{\text{DIV}_{JS}}$

which is proved to be a valid distance metric.

Jensen-Shannon distance computation via FINGER-$\hat{H}$ (batch mode):

**Input:** Two graphs $G$ and $\tilde{G}$

**Output:** $\text{JSdist}(G, \tilde{G})$

1. Obtain $G = \frac{G \oplus \tilde{G}}{2}$ and compute $\hat{H}(G)$, $\hat{H}(\tilde{G})$, and $\hat{H}(G)$ via FINGER (Fast)
2. $\text{JSdist}(G, \tilde{G}) = \hat{H}(\tilde{G}) - \frac{1}{2}[\hat{H}(G) + \hat{H}(\tilde{G})]$  

$\Rightarrow O(n + m)$ complexity inherited from $\hat{H}$

Jensen-Shannon distance computation via FINGER-$\tilde{H}$ (online mode):

**Input:** Graph $G$ and its changes $\Delta G$, Approx VNGE $\tilde{H}(G)$ of $G$

**Output:** $\text{JSdist}(G, G \oplus \Delta G)$

1. compute $\tilde{H}(G \oplus \frac{\Delta G}{2})$ and $\tilde{H}(G \oplus \Delta G)$ via FINGER (Inc.)
2. $\text{JSdist}(G, G \oplus \Delta G) = \tilde{H}(G \oplus \frac{\Delta G}{2}) - \frac{1}{2}[\tilde{H}(G) + \tilde{H}(G \oplus \Delta G)]$

$\Rightarrow O(\Delta n + \Delta m)$ complexity inherited from $\tilde{H}$

$o(\sqrt{\ln n})$ approximation guarantee of JSdist via FINGER (see paper)
Application I: Anomaly Detection in Wikipedia Networks

- Compare dissimilarity metrics of consecutive graphs via FINGER and other baseline methods:
  1. DeltaCon & RMD
  2. $\lambda$ distance (6 leading eigenvalues) & graph edit distance (GED)
  3. VNGE-NL & VNGE-GL
  4. divergence based on degree distribution

Table: Summary of four evolving Wikipedia hyperlink networks

<table>
<thead>
<tr>
<th>Datasets (graph sequence)</th>
<th>maximum # of nodes</th>
<th>maximum # of edges</th>
<th># of graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wikipedia - simple English (sEN)</td>
<td>100,312 (0.1 M)</td>
<td>746,086 (0.7 M)</td>
<td>122</td>
</tr>
<tr>
<td>Wikipedia - English (EN)</td>
<td>1,870,709 (1.8 M)</td>
<td>39,953,145 (39 M)</td>
<td>75</td>
</tr>
<tr>
<td>Wikipedia - French (FR)</td>
<td>2,212,682 (2.2 M)</td>
<td>24,440,537 (24 M)</td>
<td>121</td>
</tr>
<tr>
<td>Wikipedia - German (GE)</td>
<td>2,166,669 (2.1 M)</td>
<td>31,105,755 (31 M)</td>
<td>127</td>
</tr>
</tbody>
</table>

- Node: article. Edge: existence of hyperlinks.
  Graph: monthly hyperlink network.

- Anomaly proxy: vertex/edge overlapping dissimilarity

$$\text{VEO} (G, \tilde{G}) = 1 - \frac{2(|V \cap \tilde{V}| + |E \cap \tilde{E}|)}{|V| + |\tilde{V}| + |E| + |\tilde{E}|}$$
Table: Computation time (sec.) and Pearson correlation coefficient (PCC) of anomaly proxy and different methods. FINGER attains the best PCC and efficiency.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>FINGER-JS (Fast)</th>
<th>FINGER-JS (Inc.)</th>
<th>DeltaCon</th>
<th>RMD</th>
<th>λ dist. (Adj.)</th>
<th>λ dist. (Lap.)</th>
<th>GED</th>
<th>VNGE-NL</th>
<th>VNGE-GL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiki (sEN) PCC</td>
<td><strong>0.5593</strong></td>
<td>0.3382</td>
<td>0.1596</td>
<td>0.1718</td>
<td>0.1871</td>
<td>-0.0095</td>
<td>-0.2036</td>
<td>0.2065</td>
<td>0.2462</td>
</tr>
<tr>
<td>time</td>
<td>26.065</td>
<td><strong>0.7438</strong></td>
<td>44.952</td>
<td>44.952</td>
<td>150.16</td>
<td>99.905</td>
<td>1.666</td>
<td>13.574</td>
<td>30.483</td>
</tr>
<tr>
<td>Wiki (EN) PCC</td>
<td><strong>0.9029</strong></td>
<td>0.5583</td>
<td><strong>-0.2411</strong></td>
<td>-0.1167</td>
<td>-0.0175</td>
<td>-0.1759</td>
<td>-0.3429</td>
<td>-0.0442</td>
<td>0.1519</td>
</tr>
<tr>
<td>time</td>
<td>603.98</td>
<td><strong>13.975</strong></td>
<td>1846.1</td>
<td>1846.1</td>
<td>4417.7</td>
<td>2898.3</td>
<td>47.299</td>
<td>335.66</td>
<td>858.22</td>
</tr>
<tr>
<td>Wiki (FR) PCC</td>
<td><strong>0.8183</strong></td>
<td>0.592</td>
<td>-0.1503</td>
<td>-0.1203</td>
<td>0.0133</td>
<td>-0.1877</td>
<td>-0.4915</td>
<td>0.0552</td>
<td>0.2349</td>
</tr>
<tr>
<td>time</td>
<td>1038.6</td>
<td><strong>23.667</strong></td>
<td>2804.5</td>
<td>2804.5</td>
<td>6664.5</td>
<td>4411.4</td>
<td>83.398</td>
<td>474.42</td>
<td>1129.1</td>
</tr>
<tr>
<td>Wiki (GE) PCC</td>
<td><strong>0.6764</strong></td>
<td>0.4619</td>
<td>-0.2035</td>
<td>-0.1542</td>
<td>0.0182</td>
<td>-0.3814</td>
<td>-0.4677</td>
<td>0.2194</td>
<td>0.2679</td>
</tr>
<tr>
<td>time</td>
<td>1457.3</td>
<td><strong>32.647</strong></td>
<td>4184.1</td>
<td>4184.1</td>
<td>9462.5</td>
<td>6013.7</td>
<td>115.923</td>
<td>716.31</td>
<td>1674.6</td>
</tr>
</tbody>
</table>

P.-Y. Chen
ICML 2019
June 10, 2019 13 / 16
Application II: Detection of Bifurcation Time Instance in Dynamic Cellular Networks

- Genome-wide chromosome conformation capture contact maps among 3K cells with 12 observations
- Cellular reprogramming from human fibroblasts to skeletal muscle at some critical time instance (index 6) - Liu et al., iScience (2018)
- Temporal difference score $TDS(G_t) = \frac{\text{dist}(G_t, G_{t-1}) + \text{dist}(G_t, G_{t+1})}{2}$
Application III: Synthesized Attacks in Router Networks

- Connectivity pattern of 9 real-world autonomous system level router communication graph
- Synthesize the connectivity pattern of distributed denial of service (DoS) attacks by randomly selecting one graph and then connecting $X\%$ of nodes to a randomly chosen node in the selected graph

Table: Average detection rate on synthesized anomalous events

<table>
<thead>
<tr>
<th>DoS attack ($X%$)</th>
<th>FINGER-JS (Fast)</th>
<th>FINGER-JS (Inc.)</th>
<th>DeltaCon</th>
<th>RMD</th>
<th>$\lambda$ dist. (Adj.)</th>
<th>$\lambda$ dist. (Lap.)</th>
<th>GED</th>
<th>VNGE-NL</th>
<th>VNGE-GL</th>
<th>VEO</th>
<th>Cosine distance</th>
<th>Bhattacharyya distance</th>
<th>Hellinger distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 %</td>
<td>24 %</td>
<td>10%</td>
<td>14%</td>
<td>14%</td>
<td>10%</td>
<td>24%</td>
<td>14%</td>
<td>22%</td>
<td>22%</td>
<td>14%</td>
<td>12%</td>
<td>10%</td>
<td>12%</td>
</tr>
<tr>
<td>3 %</td>
<td>75%</td>
<td>62%</td>
<td>58%</td>
<td>58%</td>
<td>12%</td>
<td>23%</td>
<td>36%</td>
<td>39%</td>
<td>39%</td>
<td>36%</td>
<td>35%</td>
<td>14%</td>
<td>16%</td>
</tr>
<tr>
<td>5 %</td>
<td>90%</td>
<td>77%</td>
<td>90%</td>
<td>90%</td>
<td>12%</td>
<td>28%</td>
<td>41%</td>
<td>67%</td>
<td>67%</td>
<td>41%</td>
<td>37%</td>
<td>37%</td>
<td>34%</td>
</tr>
<tr>
<td>10 %</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>46%</td>
<td>46%</td>
<td>67%</td>
<td>71%</td>
<td></td>
</tr>
</tbody>
</table>

- FINGER consistently outperforms other dissimilarity metrics for different $X$
- When $X$ is small (difficult case for detection), JSdist via FINGER is more sensible than other methods
- When $X$ is large (easy case), the performance becomes similar
Conclusion and Future Work

- An efficient framework (FINGER) for fast and incremental computation of von Newman Graph Entropy and Jensen-Shannon graph distance.
- For batch graph mode, FINGER features linear complexity $O(n + m)$. For online graph mode, FINGER features incremental complexity $O(\Delta n + \Delta m)$. Both modes have asymptotic approximation guarantee.
- New applications in anomaly detection and bifurcation detection.
- Code: https://github.com/pinyuchen/FINGER
- Future work:
  1. Stochastic computation of Jensen-Shannon distance via sampling.
  2. Extension to directed graphs, and graphs with negative weights.
  3. Applications involving graph distance: e.g., brain networks, traffic networks, unsupervised and active learning.
- Contact: pin-yu.chen at ibm.com; pinyuchenTW (Twitter)
- Poster: Tuesday 6:30-9:00 pm, Pacific Ballroom #265