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

Pin-Yu Chen 1 Lingfei Wu 1 Sijia Liu 1 Indika Rajapakse 2

Abstract
The von Neumann graph entropy (VNGE) facilitates measurement of information divergence and distance between graphs in a graph sequence. It has been successfully applied to various learning tasks driven by network-based data. While effective, VNGE is computationally demanding as it requires the full eigenspectrum of the graph Laplacian matrix. In this paper, we propose a new computational framework, Fast Incremental von Neumann Graph Entropy (FINGER), which approaches VNGE with a performance guarantee. FINGER reduces the cubic complexity of VNGE to linear complexity in the number of nodes and edges, and thus enables online computation based on incremental graph changes. We also show asymptotic equivalence of FINGER to the exact VNGE, and derive its approximation error bounds. Based on FINGER, we propose efficient algorithms for computing Jensen-Shannon distance between graphs. Our experimental results on different random graph models demonstrate the computational efficiency and the asymptotic equivalence of FINGER. In addition, we apply FINGER to two real-world applications and one synthesized anomaly detection dataset, and corroborate its superior performance over seven baseline graph similarity methods.

1. Introduction
In recent years, graph-based learning has become an active research field (Shuman et al., 2013; Kalofolias, 2016; Luo et al., 2012; Shivanna & Bhattacharyya, 2014; Wang et al., 2016; Kipf & Welling, 2017; Wu et al., 2018a; Xu et al., 2018). Its success is rooted in the advanced capability of summarizing and representing phenomenal structural features embedded in graphs. In particular, evaluating similarity between graphs is crucial to network analysis and graph-based anomaly detection (Papadimitriou et al., 2010; Akoglu et al., 2015; Ranshous et al., 2015). For example, Yanardag and Vishwanathan used graph similarity for learning novel graph kernels (Yanardag & Vishwanathan, 2015), and Sharpnack et al. proposed the Lovasz extended scan statistic for anomaly detection in connected graphs (Sharpnack et al., 2013). Koutra et al. proposed DeltaCon, a state-of-the-art similarity algorithm in terms of its scalability and capability of handling weighted graphs using fast belief propagation (Koutra et al., 2016). However, these methods are sensitive to heuristic metrics and presumed models, and thus provide limited understanding on the general notion of variations between graphs. On the other hand, model-agnostic approaches such as graph entropy have been used to quantify the structural complexity of a single graph, which relates to the Shannon entropy of a probability distribution over a function of enumerated subgraphs in a graph (Simonyi, 1995; Shetty & Adibi, 2005; Li & Pan, 2016). However, graph entropy can be computationally demanding due to its use of exhaustive subgraph search.

Different from the aforementioned approaches and inspired by quantum information theory, the von Neumann graph entropy (VNGE) (Braunstein et al., 2006; Passerini & Severini, 2008; 2009) facilitates the measure of (quantum) Jensen-Shannon divergence and distance (Endres & Schindelin, 2003; Briët & Harremoës, 2009) between graphs. It associates with a model-agnostic information measure for quantifying variation between two quantum density matrices. In addition, the VNGE has been shown to be linearly correlated with classical graph entropy measures (Anand & Bianconi, 2009; Anand et al., 2011). The VNGE and the Jensen-Shannon distance have been successfully applied to structural reduction in multiplex networks (De Domenico et al., 2015), depth analysis in image processing (Han et al., 2012; Bai & Hancock, 2014), structure-function analysis in genetic networks (Seaman et al., 2017; Liu et al., 2018b), and network-ensemble comparison (Li et al., 2018). However, despite its effectiveness, the computation of VNGE requires (at most) cubic complexity in the number of nodes, thereby impeding its applicability to machine learning and data mining tasks involving a sequence of large graphs.
Contributions. To overcome the computational inefficiency of VNGE, we propose a Fast Incremental von Neumann Graph Entropy (FINGER) framework to approximate VNGE with a performance guarantee, reducing its cubic complexity to linear complexity in the number of nodes and edges. FINGER is a generic tool that applies to both batch and online graph sequences. It enables fast entropy computation when every single graph in a graph sequence is presented (e.g., a snapshot of a dynamic network, or a single-layer connectivity pattern of a multiplex network). For applications where changes in a graph (e.g., addition and deletion of nodes or edges over time) are continuously reported (e.g., streaming graphs), FINGER also allows online computation based on incremental graph changes. We prove that FINGER maintains an approximation guarantee and is asymptotically equivalent to the exact VNGE under some eigenspectrum conditions, which is further validated by different synthetic random graphs. We then apply FINGER to developing efficient algorithms for the computation of Jensen-Shannon distance between graphs. Comparing to the state-of-the-art graph similarity methods and two alternative approximate VNGE, FINGER yields superior and robust performance for anomaly detection in evolving Wikipedia networks and router communication networks, as well as bifurcation analysis in dynamic genomic networks. These applications show the effectiveness and potentials of Jensen-Shannon distance for network learning in a wide range of domains, which has not been rigorously explored owing to its high computation complexity in the absence of FINGER.

The contributions of this paper and the proposed framework (FINGER) are summarized as follows.

- Two types of approximate VNGE reducing its cubic complexity to linear complexity are proposed to support fast and incremental computation of VNGE. We derive their approximation error bounds and show asymptotic equivalence relative to the exact VNGE under mild conditions.
- FINGER achieves nearly 100% reduction in computation time for VNGE of different graphs and enables scalable Jensen-Shannon graph distance computation.
- On two real-world applications (anomaly detection and cellular bifurcation analysis) and one synthesized dataset, FINGER exhibits outstanding and robust performance over 7 baseline and state-of-the-art methods.

Related Work. The VNGE was firstly defined based on the combinatorial graph Laplacian matrix (Braunstein et al., 2006; Passerini & Severini, 2008; 2009; De Domenico et al., 2015; Li et al., 2018). Variants of VNGE and their approximations have been proposed in the literature, including the normalized graph Laplacian matrix (Shi & Malik, 2000) proposed in (Han et al., 2012) and the generalized graph Laplacian matrix of directed graphs (Chung, 2005) proposed in (Ye et al., 2014). However, these alternatives lack approximation justification and are shown to be suboptimal in Section 4. To the best of our knowledge, this paper is the first work that provides fast VNGE computation with a provable approximation analysis.

2. FINGER: Theory and Algorithms

2.1. Background and Preliminaries

Using terminology from quantum statistical mechanics, a density matrix $\Phi$ describing a quantum system in a mixed state can be cast as a statistical ensemble of several quantum states. The $n \times n$ matrix $\Phi$ is symmetric, positive semidefinite, and satisfies $\text{trace}(\Phi) = 1$. The von Neumann entropy of a quantum system is defined as $H = -\text{trace}(\Phi \ln \Phi)$ (Von Neumann, 1955), where $\ln \Phi$ denotes matrix logarithm. Let $\{\lambda_i\}_{i=1}^n$ be the sorted eigenvalues of $\Phi$ such that $0 \leq \lambda_n \leq \ldots \leq \lambda_1$. The definition of von Neumann entropy is equivalent to $H = -\sum_{i=1}^n \lambda_i \ln \lambda_i$, where the convention $\lim_{0 \ln 0 = 0}$ is used due to $\lim_{x \rightarrow 0^+} x \ln x = 0$. Moreover, since $\sum_i \lambda_i = 1$ and $\lambda_i \geq 0$ for all $i$, the von Neumann entropy can be viewed as the Shannon entropy associated with the eigenspectrum $\{\lambda_i\}_{i=1}^n$.

We consider the class of undirected weighted simple non-empty graphs with nonnegative edge weights, denoted by $G$. Let $G = (\mathcal{V}, \mathcal{E}, W)$ denote a single graph, where $\mathcal{V}$ and $\mathcal{E}$ denote its node and edge set with cardinality $|\mathcal{V}| = n$ and $|\mathcal{E}| = m$, respectively, and $W$ is an $n \times n$ matrix with entry $[W]_{ij} = w_{ij}$ denoting the weight of an edge $(i, j) \in \mathcal{E}$. A graph sequence $\{G_t\}_{t=1}^T$ refers to a set of $T$ graphs indexed by $t \in \{1, \ldots, T\}$ with known node-to-node correspondence, where $G_t \in G$ for all $t$. The combinatorial graph Laplacian matrix of $G$ is defined as $L = S - W$ (Luxburg, 2007), where $S = \text{diag}(s_1, \ldots, s_n)$ is a diagonal matrix and its diagonal entry $s_i = \sum_{j=1}^n w_{ij}$ is the nodal strength (weighted degree) of a node $i \in \mathcal{V}$. Connecting the von Neumann entropy to graphs, the VNGE, denoted by $H(G)$, is defined by replacing $\Phi$ with $L_N = c \cdot L$ (Braunstein et al., 2006; Passerini & Severini, 2008; 2009), where $c = 1/\text{trace}(L)$ is a trace normalization factor. It has been proved in (Passerini & Severini, 2008) that for any $G \in G$, $H(G) \leq \ln(n - 1)$, where the equality holds when $G$ is a complete graph. Note that since computing VNGE requires the entire eigenspectrum $\{\lambda_i\}_{i=1}^n$ of $L_N$, it incurs full eigenvalue decomposition on $L_N$ and has cubic complexity $O(n^3)$ (Horn & Johnson, 1990), making it computationally infeasible for large graphs.

$$f(n) = O(h(n)), \ f(n) = o(h(n)) \text{ and } f(n) = \Omega(h(n)) \text{ mean } \limsup_{n \rightarrow \infty} \frac{f(n)}{h(n)} < \infty, \liminf_{n \rightarrow \infty} \frac{f(n)}{h(n)} = 0, \text{ and } \limsup_{n \rightarrow \infty} \frac{f(n)}{h(n)} > 0, \text{ respectively.}$$

$^2$For computing all eigenvalues of large matrices, a viable solution is direct methods, possibly with parallel eigensolvers for acceleration. The complexity for computing $\{\lambda_i\}_{i=1}^n$ of $L_N$ is typically $O(n^2 + \frac{4}{3}n^3)$ (Bai et al., 2000).
In what follows, we propose two types of approximate VNGE (\(\hat{H}\) and \(\tilde{H}\)) for the exact VNGE \(H\), where \(\hat{H}\) and \(\tilde{H}\) possess linear computation complexity and satisfy \(H \leq \hat{H} \leq \tilde{H}\). Depending on the data format and problem setup, \(\hat{H}\) is designed for fast computation of \(H\) for a single graph, and \(\tilde{H}\) is designed for online computation of \(H\) based on incremental graph changes. Furthermore, we derive approximation error and prove asymptotic equivalence relative to \(H\) under mild conditions on the eigenspectrum \(\{\lambda_i\}_{i=1}^n\) of \(L_N\). Our proofs are given in the supplementary material.

2.2. Approximation of von Neumann Graph Entropy

Recall that computing \(H = -\sum_{i=1}^n \lambda_i \ln \lambda_i\) requires \(O(n^3)\) computation complexity. For computation acceleration, we first reduce its computation complexity by using the quadratic approximation of the term \(\lambda_i \ln \lambda_i\) in \(H\) via Taylor series expansion, leading to the following lemma.

**Lemma 1** (Quadratic approximation \(Q\) of \(H\)). For any \(G \in \mathcal{G}\), the quadratic approximation \(Q\) of the von Neumann graph entropy \(H\) via Taylor series expansion is equivalent to \(Q = 1 - c^2(\sum_{i \in V} s_i^2 + 2 \sum_{(i,j) \in E} w_{ij})\), where \(c = \frac{1}{2}\) and \(S = \text{trace}(L) = \sum_{i \in V} s_i = 2 \sum_{(i,j) \in E} w_{ij}\).

It is clear from Lemma 1 that \(Q\) only depends on the edge weights in \(G = (V,E,W)\), resulting in linear computation complexity\(^3\) \(O(n + m)\), where \(|V| = n\) and \(|E| = m\). We note that higher-order (beyond quadratic) approximation of \(H\) is plausible at the price of less computational efficiency and possibly excessive subgraph pattern searching. For example, the cubic approximation of \(H\) involves the computation of trace(\(W^3\)), which relates to the sum of edge weights of every triangle in \(G\). To identify the approximation accuracy and equivalence of \(Q\) with respect to \(H\), the following theorem shows the approximation bounds on \(H\) in terms of \(Q\) and the eigenspectrum \(\{\lambda_i\}_{i=1}^n\) of \(L_N\).

**Theorem 1** (Approximation bounds on \(H\)). For any \(G \in \mathcal{G}\), let \(\lambda_{\text{max}}\) and \(\lambda_{\text{min}}\) be the largest and smallest positive eigenvalues of \(L_N\), respectively. If \(\lambda_{\text{max}} < 1\), then \(-Q \frac{\ln \lambda_{\text{max}}}{\ln \lambda_{\text{min}}} \leq H \leq -Q \frac{\ln \lambda_{\text{min}}}{\ln \lambda_{\text{max}}}\). The bounds become exact and \(\tilde{H} = \ln(n - 1)\) when \(G\) is a complete graph with identical edge weight.

Note that Theorem 1 excludes the extreme case when \(\lambda_{\text{max}} = 1\), as the resulting VNGE is trivial (\(H = 0\)). The condition \(\lambda_{\text{max}} < 1\) holds for any graph \(G \in \mathcal{G}\) having a connected subgraph with at least 3 nodes. In addition to the approximation bounds presented in Theorem 1, the corollary below further shows asymptotic equivalence between \(Q\) and \(H\) under mild conditions on \(\lambda_{\text{max}}\) and \(\lambda_{\text{min}}\).

**Corollary 1** (Asymptotic equivalence of \(Q\)). For any \(G \in \mathcal{G}\), let \(n_+\) denote the number of positive eigenvalues of \(L_N\).

If \(n_+ = \Omega(n)\) and \(\lambda_{\text{min}} = \Omega(\lambda_{\text{max}})\), then \(\frac{H}{\ln n} - Q \to 0\) as \(n \to \infty\).

Corollary 1 suggests that the VNGE of large graphs with balanced eigenspectrum (i.e., \(\lambda_{\text{min}} = \Omega(\lambda_{\text{max}})\)) can be well approximated by \(Q\) and a factor \(\ln n\). The condition of balanced eigenspectrum holds in regular and homogeneous random graphs (Passerini & Severini, 2008; Du et al., 2010). Furthermore, since \(n_+\) equals to \(n - g\), where \(g\) is the number of connected components in \(G\) (Merris, 1994), the condition \(n_+ = \Omega(n)\) holds when \(g = o(n)^1\).

2.3. FINGER-\(\hat{H}\): Approximate von Neumann Graph Entropy \(\hat{H}\) Using \(Q\) and \(\lambda_{\text{max}}\)

Based on the derived lower bound of \(H\) in Theorem 1, we propose the first type of approximate VNGE \(\hat{H}\) using \(Q\) and \(\lambda_{\text{max}}\) for any \(G \in \mathcal{G}\), which is defined as

\[
\hat{H}(G) = -Q \ln \lambda_{\text{max}}.
\]

Comparing to the lower bound \(-Q \frac{\ln \lambda_{\text{max}}}{\ln \lambda_{\text{min}}}\) in Theorem 1, \(\hat{H}\) is a looser lower bound on \(H\) since \(1 - \lambda_{\text{min}}\) is negligible, especially for large graphs.

More importantly, since \(\lambda_{\text{max}}\) is the largest eigenvalue of \(L_N\) and by definition \(L_N\) has \(n + m\) nonzero entries, the computation of \(\lambda_{\text{max}}\) only requires \(O(m + n)\) operations via power iteration methods (Horn & Johnson, 1990; Wu et al., 2017; Liao et al., 2019), leading to the same complexity as \(Q\). Consequently, by only acquiring \(\lambda_{\text{max}}\) instead of the entire eigenspectrum \(\{\lambda_i\}_{i=1}^n\), the computation of \(H\) has linear complexity \(O(m + n)\), resulting in significant computation reduction when compared with the exact VNGE \(H\), which requires cubic complexity\(^2\) \(O(n^3)\). In addition to computational efficiency, the following corollary shows that the approximation error of \(\hat{H}\), defined as \(H - \hat{H}\), decays at the rate of \(\ln n\) under the same conditions as in Corollary 1. We note that the \(o(\ln n)\) approximation error rate is nontrivial since \(H \leq \ln(n - 1)\) for any \(G \in \mathcal{G}\) (Passerini & Severini, 2008; Du et al., 2010).

**Corollary 2** (Asymptotic approximation error of \(\hat{H}\)). For any \(G \in \mathcal{G}\), if \(n_+ = \Omega(n)\) and \(\lambda_{\text{min}} = \Omega(\lambda_{\text{max}})\), then the scaled approximation error error (SAE) \(\frac{H - \hat{H}}{\ln n}\) tends to 0 as \(n \to \infty\), implying \(H - \hat{H} = o(\ln n)\).

2.4. FINGER-\(\tilde{H}\): Approximate von Neumann Graph Entropy \(\tilde{H}\) Using \(Q\) and \(s_{\text{max}}\)

The proxy \(\tilde{H}\) in Section 2.3 enables fast computation of VNGE for a single graph. As the exact online update of the eigenvalue \(\lambda_{\text{max}}\) in \(\hat{H}\) based on incremental graph changes is challenging, we propose the second type of approximate
Fast Incremental von Neumann Graph Entropy Computation: Theory, Algorithm, and Applications

VNGE $\tilde{H}$ using $Q$ and the largest nodal strength $s_{\text{max}} = \max_{i \in V} s_i$ in a graph, which allows simple incremental update of $\tilde{H}$ based on graph changes but at the price of larger approximation error than that of $H$. The approximate VNGE $\tilde{H}$ is defined as

$$\tilde{H}(G) = -Q \ln(2c \cdot s_{\text{max}}), \quad (2)$$

where $c$ is the trace normalization constant. Using the definition $L_N = c \cdot L$ and the upper bound on the largest eigenvalue of $L$ in (Anderson Jr & Morley, 1985), we obtain $\tilde{H} < \tilde{H} < H$ since $\lambda_{\text{max}} \leq 2c \cdot s_{\text{max}}$, implying $\tilde{H}$ is a looser lower bound on $H$ when compared with $H$. Nonetheless, the following corollary shows the approximation error of $\tilde{H}$ also decays at the same rate $o(\ln n)$ as $H$.

Corollary 3 ($o(\ln n)$ approximation error of $\tilde{H}$). For any $G \in \mathcal{G}$, if $n_+ = \Omega(n)$ and $\lambda_{\text{min}} = \Omega(\lambda_{\text{max}})$, then the scaled approximation error (SAE) $\frac{H - \tilde{H}}{\ln n} \rightarrow 0$ as $n \rightarrow \infty$, implying $H - \tilde{H} = o(\ln n)$.

To enable incremental computation of VNGE using $\tilde{H}$, let $G = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ and $G' = (\mathcal{V}', \mathcal{E}', \mathcal{W}')$ be any two graphs from a graph sequence. Without loss of generality we assume $G$ and $G'$ have a common node set $\mathcal{V}_c$ with $|\mathcal{V}_c| = n$ nodes. In particular, the graph $\Delta G = (\Delta \mathcal{V}, \Delta \mathcal{E}, \Delta \mathcal{W})$ with $|\Delta \mathcal{V}| = \Delta n$ and $|\Delta \mathcal{E}| = \Delta m$ is introduced to represent the changes made from converting $G$ to $G'$, denoted by $G' = G \oplus \Delta G$. The terms $\{\Delta s_i\}_{i \in \Delta \mathcal{V}}$ and $\{\Delta w_{ij}\}_{(i,j) \in \Delta \mathcal{E}}$ denote the nodal strengths and edge weights of $\Delta G$, respectively, and $\Delta S = \sum_{i \in \Delta \mathcal{V}} \Delta s_i$. Let $Q'$ be the quadratic approximation of $H(G')$. The theorem below shows that $Q'$ can be efficiently updated based on $Q$ of $H(G)$, the values of $s_{\text{max}}$ and $c$ from $G$, and $\Delta G$, yielding competent complexity $O(\Delta n + \Delta m)$.

Theorem 2 (Incremental update of $Q'$). For any $G, G' \in \mathcal{G}$ such that $G' = G \oplus \Delta G$, given $Q$, $G$, and $\Delta G$, the term $Q'$ can be efficiently updated by incremental graph changes as

$$Q' = \frac{Q - G}{1 + \Delta S} - \left(\frac{1 + \Delta S}{1 + \Delta S}\right)^2 \Delta Q + 1, \quad \text{where} \quad \Delta Q = 2 \sum_{i \in \Delta \mathcal{V}} s_i \Delta s_i + \sum_{i \in \Delta \mathcal{V}} \Delta s_i^2 + 4 \sum_{(i,j) \in \Delta \mathcal{E}} w_{ij} \Delta w_{ij} + 2 \sum_{(i,j) \in \Delta \mathcal{E}} \Delta w_{ij}^2, \quad \text{and} \quad \Delta c = c \frac{s_{\text{max}}}{1 + \Delta S}.
$$

Furthermore, by the definition of $\tilde{H}$ in (2), $\tilde{H}(G \oplus \Delta G)$ can be efficiently updated by

$$\tilde{H}(G \oplus \Delta G) = -Q' \ln[2(c + \Delta c)(s_{\text{max}} + \Delta s_{\text{max}})] \quad (3)$$

given $Q$, $s_{\text{max}}$ and $c$ from $G$, and graph changes $\Delta G$, where $\Delta c$ is defined in Theorem 2, and $\Delta s_{\text{max}}$ is the maximum value of 0 and $\max_{i \in \Delta \mathcal{V}} (s_i + \Delta s_i) - s_{\text{max}}$. The computation complexity of $\tilde{H}(G \oplus \Delta G)$ is $O(\Delta n + \Delta m)$ since the incremental update formula of $Q'$ in Theorem 2 and the computation of $\Delta s_{\text{max}}$ only take $O(\Delta n + \Delta m)$ operations.

Algorithm 1 FINGER-JSzdist (Fast)

**Input:** Two graphs $G$ and $G'$ from a graph sequence

**Output:** JSdist($G, G'$)

1. Obtain $\overline{G} = \frac{G \oplus G'}{2}$ and compute $\tilde{H}(G), \tilde{H}(G')$, and $\tilde{H}(\overline{G})$ via FINGER-$\tilde{H}$ from (1)
2. JSdist($G, G'$) = \left(\tilde{H}(\overline{G}) - \frac{1}{2}[\tilde{H}(G) + \tilde{H}(G')]\right)^{1/2}

Algorithm 2 FINGER-JSzdist (Incremental)

**Input:** A graph $G$, graph changes $\Delta G$, and $\tilde{H}(G)$

**Output:** 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-$\tilde{H}$ from (3) and Theorem 2
2. JSdist($G, G \oplus \Delta G$) = \left(\tilde{H}(G \oplus \frac{\Delta G}{2}) - \frac{1}{2}[\tilde{H}(G) + \tilde{H}(G \oplus \Delta G)]\right)^{1/2}

2.5 Fast and Incremental Algorithms for Jensen-Shannon Distance between Graphs

As summarized in Algorithms 1 and 2, one major utility of VNGE is the computation of Jensen-Shannon distance (JSdist) between any two graphs from a graph sequence. Consider two graphs $G = (\mathcal{V}, \mathcal{E}, \mathcal{W}) \in \mathcal{G}$ and $G' = (\mathcal{V}', \mathcal{E}', \mathcal{W}') \in \mathcal{G}$, and let $\overline{G} = (\mathcal{V}, \overline{\mathcal{E}}, \overline{\mathcal{W}}) = \frac{G \oplus G'}{2}$ denote their averaged graph such that $\overline{\mathcal{W}} = \mathcal{W} + \mathcal{W}'$. Then the Jensen-Shannon divergence between $G$ and $G'$ can be computed by $\text{JSdiv}(G, G') = H(\overline{G}) - \frac{1}{2}[H(G) + H(G')]$ (De Domenico et al., 2015). Furthermore, the Jensen-Shannon distance between $G$ and $G'$ is defined as $\text{JSdist}(G, G') = \sqrt{\text{JSdiv}(G, G')}$, which has been proved to be a valid distance metric in (Endres & Schindelin, 2003; Briët & Harremoës, 2009). The exact computation of JSdist requires $O(n^3)$ computation complexity by the definition of $H$, where $|\mathcal{V}_c| = n$, which is computationally cumbersome for large graphs. To overcome its computational inefficiency, we apply the developed FINGER-$\tilde{H}$ and FINGER-$\tilde{H}$ to the computation of JSdist. If each graph $G_t$ in a graph sequence $\{G_t\}_{t=1}^T$ is given, then FINGER-JSzdist (Fast) allows fast computation of JSdist and features linear computation complexity inherited from $\tilde{H}$. If a graph sequence is presented by sequential graph changes $\{\Delta G_t\}_{t=1}^{T-1}$ such that $G_{t+1} = G_t \oplus \Delta G_t$, then FINGER-JSzdist (Incremental) allows online computation of JSdist relative to the incremental graph changes. Their superior performance will be discussed in Section 4.

Codes: https://github.com/pinyuchen/FINGER
3. Experiments

In this section, we conducted intensive experiments on the VNGE of three kinds of synthetic random graphs to study the effects of graph size, average degree, and graph regularity on the approximation error of FINGER and its computational efficiency. The three random graph models are: (i) Erdos-Renyi (ER) model (Erdős & Rényi, 1959) – every node pair is connected independently with probability \( p_{ER} \); (ii) Barabasi-Albert (BA) model (Barabási & Albert, 1999) – the degree distribution follows a power-law distribution; and (iii) Watts-Strogatz (WS) model (Watts & Strogatz, 1998) – an initially regular ring network with independent edge rewiring probability \( p_{WS} \) for simulating small-world networks. The parameter \( p_{WS} \) controls the regularity of graph connectivity, and smaller \( p_{WS} \) gives more regular graphs.

Since \( \hat{H} \leq \widetilde{H} \leq H \), the approximation error (AE) is defined as \( H - \hat{H} \) and \( H - \widetilde{H} \), respectively. The scaled approximation error (SAE) is defined as \( \frac{\text{AE}}{\text{time}(H)} \), which is a proper scaling according to our error analysis in Section 2, and it also makes a fair comparison of graphs with different number of nodes. The computation time reduction ratio (CTRR) is defined as \( \frac{\text{time}(H) - \text{time}(X)}{\text{time}(H)} \), where \( X \in \{ \hat{H}, \widetilde{H} \} \) and \( \text{time}(Y) \) denotes the computation time for \( Y \in \{ H, \hat{H}, \widetilde{H} \} \). All experiments (including Section 4) were conducted by Matlab R2016 on a 16-core machine with 128 GB RAM. The results in this section are averaged over 10 random trials. Additional results are reported in the supplementary material.

The effect of average degree \( \bar{d} \) and graph regularity parameter \( p_{WS} \). Figures 1 (a) and 1 (b) display the exact and the two approximate VNGE of ER and BA graphs and the corresponding CTRR under varying \( \bar{d} \). When fixing the number of nodes \( n \), both \( \hat{H} \) and \( \widetilde{H} \) better match \( H \) as \( \bar{d} \) increases, suggesting their AE decays with \( \bar{d} \). Comparing their CTRR, the computation of \( \hat{H} \) and \( \widetilde{H} \) enjoys at least 97% speed-up relative to \( H \). The drastic reduction in computation time can be explained by the efficient linear complexity of FINGER, as opposed to the high complexity in computing the entire eigenspectrum for calculating \( H \). The CTRR of \( \hat{H} \) slightly decays with \( \bar{d} \) due to the growing number of nonzero entries (edges) in \( L_N \), resulting in increasing operations for computing \( \lambda_{\text{max}} \). Although the AE of \( \hat{H} \) is always smaller than that of \( \widetilde{H} \) due to the fact that \( \hat{H} \leq \widetilde{H} \leq H \), the CTRR of \( \hat{H} \) has nearly 100% speed-up relative to \( H \) by simply requiring the information of \( s_{\text{max}} \) instead of \( \lambda_{\text{max}} \) from a graph.

Figure 1 (c) displays the AE and CTRR of \( \hat{H} \) and \( \widetilde{H} \) under
varying edge rewiring probability \( p_{WS} \) and different average degree \( \bar{d} \in \{6, 10, 20, 50\} \) of WS model. Similar to ER and BA graphs, when fixing \( n \) and \( p_{WS} \), the AE of \( \tilde{H} \) and \( \hat{H} \) decays as \( \bar{d} \) increases. When \( n \) and \( \bar{d} \) are fixed, smaller \( p_{WS} \) yields less AE for both \( \tilde{H} \) and \( \hat{H} \), suggesting that FINGER attains better approximation when graphs are more regular. Since the curves of CTRR for different \( \bar{d} \) in WS model have similar behavior, here we only report the results when \( \bar{d} = 50 \). Consistent with the observations in ER and BA graphs, in WS graphs the CTRR of \( \tilde{H} \) and \( \hat{H} \) achieves nearly 100% improvement relative to \( H \), and \( \hat{H} \) attains slightly better CTRR than \( \tilde{H} \) at the price of larger AE.

The effect of graph size \( n \). Figure 2 displays the SAE of FINGER under the three random graph models when varying the number of nodes \( n \). Since the results of \( \tilde{H} \) and \( H \) are similar, we show the SAE of \( \tilde{H} \) in Figure 2 and report the SAE of \( \hat{H} \) in the supplementary material. By the fact that ER and WS graphs have balanced eigenspectrum (Van Mieghem, 2010), for ER and WS models the SAE of both \( \tilde{H} \) and \( H \) decays as \( n \) increases, which verifies the \( o(\ln n) \) approximation error as stated in Corollaries 2 and 3. On the other hand, the SAE of BA graphs is observed to grow logarithmically in \( n \) due to the existence of extreme eigenvalues (imbalanced eigenspectrum) (Van Mieghem, 2010; Goh et al., 2001). Similar to the observations from fixed-size graphs, for a fixed \( n \) the SAE decays with \( \bar{d} \) and graph regularity in all cases. In addition, the CTRR attains nearly 100% speed-up relative to \( H \) for moderate-size graphs (\( n \geq 2000 \)).

4. Applications

Here we apply FINGER to the computation of Jensen-Shannon (JS) distance between graphs (Section 2.5) in two applications and one synthesized dataset and demonstrate its outstanding performance over seven baseline and state-of-the-art methods in terms of efficiency and effectiveness.

Anomaly detection in evolving Wikipedia hyperlink networks. Wikipedia is an online encyclopedia that allows editing and referencing between articles. By viewing an article as a node and a hyperlink as an edge, the evolution of Wikipedia forms a graph sequence \( \{G_t\}_{t=1}^T \) over time. Table 1 summarizes four evolving Wikipedia networks of different language settings collected in (Mislove, 2009; Preusse et al., 2013), where each graph \( G_t = (V_t, E_t, W_t) \) corresponds to a monthly snapshot of a hyperlink network. These datasets are presented in terms of addition and deletion of nodes or edges with timestamps (i.e., continuous graph changes \( \{\Delta G_t\}_{t=1}^{T-1} \)), which directly applies to incremental JS distance computation via FINGER (Algorithm 2). Fast JS distance computation via FINGER (Algorithm 1) can also be applied by computing \( G_{t+1} = G_t \oplus \Delta G_t \) to obtain \( \{G_t\}_{t=1}^T \). The task of anomaly detection is to identify noticeable changes (relative to the bulk network) in the consecutive monthly snapshots of these massive Wikipedia hyperlink networks.

Bifurcation detection in dynamic genomic networks. The genome-wide chromosome conformation capture (Hi-C) contact maps (Beloqui et al., 2009) for studying cell reprogramming from human fibroblasts to skeletal muscle can be viewed as a graph sequence consisting of 12 sampled spatial measurements, in which the cell reprogramming undergoes a space-time bifurcation at the 6th measurement as verified in (Liu et al., 2018a). The task is to identify this bifurcation instance based on the dynamic Hi-C contact maps. Additional descriptions of this dataset are given in the supplementary material.

Evaluation. We note that there are two major differences between these two applications: (i) unweighted v.s. weighted graphs and (ii) with v.s. without ground truth. In the Wikipedia case (unweighted graphs), our main goal is to use these large datasets to demonstrate the efficient computation of JS distance via FINGER owing to its linear complexity. Additionally, since there are no labels for verifying the detected changes, we conduct an ex post facto correlation analysis using an explicit and explainable anomaly metric – the vertex/edge overlapping (VEO) score (Papadimitriou et al., 2010). VEO is a properly normalized metric reflecting topological differences between two unweighted graphs, defined as \( 1 - 2|\Delta V_t|/|V_t||V_{t+1}| \), which is between [0, 1] and relates to the SorensenDice coefficient (Dice, 1945; Sørensen, 1948) for comparing the similarity of two samples. In the Wikipedia experiments, a high VEO score directly pinpoints the month when articles are edited by a relatively significant amount. Consequently, VEO can be used as an anomaly proxy for ex post facto analysis in our setting. In the genome case (weighted graphs), the ground-truth bifurcation instance was verified. Moreover, unlike the Wikipedia case, the genome dataset contains nonnegative edge weights indicating cell interaction strengths. Therefore, in this case VEO is not an appropriate anomaly proxy because by definition it is insensitive to edge weight changes.

Comparative methods. We compare the proposed method with the following baseline methods:

- DeltaCon (Koutra et al., 2016): DeltaCon uses the idea of fast belief propagation to compute graph similarity and outputs a similarity score \( \text{Sim}_{\text{DC}} \in [0, 1] \). We use \( 1 - \text{Sim}_{\text{DC}} \) as the anomaly score.
- RMD (Koutra et al., 2016): RMD is the Matusita distance deduced from DeltaCon, which is defined as \( 1 - \text{Sim}_{\text{RMD}} \).
- A distance (Bunke et al., 2007; Wilson & Zhu, 2008): The Euclidean distance between two sets of top \( k \) eigenvalues of a matrix. Here we use the weight matrix \( W \) (Adj.) and the graph Laplacian matrix \( L \) (Lap.), and set \( k = 6 \).
Table 1. 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>

Table 2. Computation time (seconds) and Pearson correlation coefficient (PCC) between the anomaly proxy and different methods. FINGER attains the best PCC and time efficiency. The Spearman’s rank correlation analysis is given in Table S1 of the supplement.

<table>
<thead>
<tr>
<th>Datasets (graph sequence)</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)</td>
<td>0.5593</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>0.7438</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)</td>
<td>0.9029</td>
<td>0.5583</td>
<td>-0.2411</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>13.975</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)</td>
<td>0.8183</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>23.667</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)</td>
<td>0.6764</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>32.647</td>
<td>4184.1</td>
<td>4184.1</td>
<td>9462.5</td>
<td>6013.7</td>
<td>716.31</td>
<td>1674.6</td>
<td></td>
</tr>
</tbody>
</table>

• GED (Bunke et al., 2007): graph edit distance (GED) for undirected unweighted graphs is the number of operations (node/edge additions and removals) required to convert a graph \( G_t \) to another graph \( G_{t+1} \).

• VNGE-NL (Han et al., 2012) / VNGE-GL (Ye et al., 2014): Two VNGE heuristics using the normalized/generalized graph Laplacian matrix. Unlike FINGER, they lack approximation error guarantee.

Wikipedia results. We compute the dissimilarity metrics of each method and compare them with the anomaly proxy in terms of the Pearson correlation coefficient (PCC). A higher PCC suggests a better match to the anomaly proxy for detecting abnormal monthly edit changes relative to the bulk network. The PCC and computation time of each method are reported in Table 2. For illustration, the dissimilarity metrics of Wikipedia-EN are shown in Figure 3. The plots of the other Wikipedia networks are given in the supplementary material. The statistics of the anomaly proxy meet the intuition that in the earlier stage the monthly evolution of Wikipedia is more drastic, and in the later stage it becomes stable (i.e., less anomalous) since the changes are subtle relative to the entire network. In Table 2, FINGER-JSdist (Fast) attains the best PCC (0.9029 PCC in Table 2 and 0.7973 SRCC in Table S1). FINGER-JSdist (Incremental) has efficient computation time and attains the second best PCC and SRCC among all methods. The plots of the other Wikipedia networks are given in the supplementary material. The statistics of the anomaly proxy meet the intuition that in the earlier stage the monthly evolution of Wikipedia is more drastic, and in the later stage it becomes stable (i.e., less anomalous) since the changes are subtle relative to the entire network. In Table 2, FINGER-JSdist (Fast) attains the best PCC (0.9029 PCC in Table 2 and 0.7973 SRCC in Table S1). FINGER-JSdist (Incremental) has efficient computation time and attains the second best PCC and SRCC among all methods.

As discussed in the “Evaluation” paragraph, the main purpose of the Wikipedia experiments (without ground truths) is to show the efficiency in fast JS distance computation of large real-world graphs, enabled by FINGER. Additionally, our ex post facto analysis shows high correlation of FINGER and the anomaly proxy.
Table 3. Detection rate on synthesized anomalous events in the dynamic communication networks.

<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>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>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>
</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>
</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>
</tr>
<tr>
<td>10 %</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>91%</td>
<td>81%</td>
<td>91%</td>
<td>91%</td>
</tr>
</tbody>
</table>

Figure 4. Bifurcation detection of cell reprogramming in dynamic genomic networks via the temporal difference score (TDS) of different methods (y-axis). The red squares indicate the detected bifurcation points. Among all the compared methods, FINGER-JSdist (Algorithm 1) is the only method that correctly detects the ground-truth bifurcation point (index 6), and its TDS resembles the shape of the ground-truth statistic.

Synthesized anomaly detection results. For further validation, we use another real-world dynamic peering network dataset at the autonomous system (AS) level (the Oregon-1 dataset (Leskovec et al., 2005)) to synthesize anomalous connectivity patterns that mimic the denial-of-service (DoS) attacks. Here each graph represents the router connectivity over a certain time period, leading to 9 such graphs. We synthesize anomalous events by first selecting one graph from the first 8 graphs at random, and then connecting X% of nodes to a randomly chosen node in the selected graph. This synthesized connection pattern mimics that of the DoS attack, in which multiple nodes (e.g., a botnet) aim to connect to the target node simultaneously. The task is to detect this synthesized anomalous event by comparing the dissimilarity metric between consecutive graphs. Table 3 reports the detection rate of different methods, where the detection rate is defined as the fraction of 100 random instances in which the anomalous event appears in the top-2 ranking based on a dissimilarity metric. Tested on X = \{1, 3, 5, 10\}%, FINGER-JS (Fast) consistently attains the best detection rate among all methods, suggesting the stability and superiority of the proposed method. On the other hand, the compared methods are not as robust as FINGER. Notably, when X is small (i.e., the more challenging case for detection as the attack becomes stealthier), the detection performance of FINGER is more sensible than other methods. As X becomes large, which means the DoS attack pattern is more apparent, the detection performance becomes similar.

5. Conclusion

In this paper, we proposed FINGER, a novel framework for efficiently computing von Neumann graph entropy (VNGE). FINGER reduces the computation of VNGE from cubic complexity to linear complexity for a given graph, and allows online computation based on incremental graph changes. In addition to bounded approximation error, our theory shows that FINGER is guaranteed to have asymptotic equivalence to the exact VNGE under mild conditions, which has been validated by extensive experiments on three different random graph models. The high efficiency of FINGER also leads to scalable network learning algorithms for computing Jensen-Shannon distance between graphs. Furthermore, we use two domain-specific applications and one synthesized dataset to corroborate the efficiency and effectiveness of FINGER compared to 7 baseline graph similarity methods. The results demonstrate the power of FINGER in tackling large network analysis and (unsupervised) learning problems in different domains. Our future work includes extension to directed graphs and negative edge weights.
Acknowledgment

Pin-Yu Chen, Lingfei Wu and Sijia Liu acknowledge the support from MIT-IBM Watson AI Lab. Indika Rajapakse is supported in part by the Lifelong Learning Machines program from DARPA/MTO.

References


Kalofolias, V. How to learn a graph from smooth signals. In International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 920929, 2016.


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


