Analysis and Approximation Algorithms for Large-Scale
Networks Using All Eigenvalues
全ての固有値を利用した大規模ネットワークの分析と近似ア
ルゴリズム

by

Naoki Murakami
村上直輝

A Senior Thesis
卒業論文

Submitted to
the Department of Information Science
the Faculty of Science, the University of Tokyo
on January 29, 2021
in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Science

Thesis Supervisor: Hiroshi Imai　今井浩
Professor of Information Science

ABSTRACT

Spectral graph theory studies the relationship between the properties of graphs and eigenvalues. Spectra, or eigenvalues, are useful for analyzing the properties and structure of networks. However, since computing the full eigendecomposition is expensive, large networks have been analyzed using part of the spectrum. Research using the overall spectra of a large-scale network is still an underdeveloped field.

Firstly, we propose two approximation algorithms to compute the sum of powers of eigenvalues $\sum_{i=0}^{n-1} \lambda_i^k$ that scale to large networks. The sum of powers of eigenvalues is an important property, and in recent years, research on related inequalities has been conducted. However, the conventional simple algorithms for calculating the sum of powers of eigenvalues do not scale to large graphs. Our proposed method makes it possible to calculate the sum of powers of eigenvalues for networks with millions or billions of vertices in a practical time. This makes it possible to use the sum of powers of eigenvalues as a new measure for the network.

In this research, we also study the spectral distribution obtained by an algorithm based on a method similar to the proposed method. Recently, a method to calculate the distribution of the entire spectrum has been devised, but since there are still few studies that visualize the spectral distribution of real-world networks, the correspondence between the spectral distribution and the properties of the network has not yet been considered deeply. We first discuss the quantitative evaluation of spectra and their application methods. For the quantitative evaluation and comparison of spectral distributions, we propose to use the concepts of divergence and distance for discrete probability distributions. As an application method, we also mention a method for fast retrieval of similar spectra using cosine distance. Next, we present the spectral distributions obtained from the synthesized graph and the real-world network. Next, we analyze the obtained spectral distributions and discuss the relationship between the network properties and the spectra. The results show that for sparse networks, the local structure of the network, called motifs, and the distribution of eigenvalues are related. Besides, graphs in which nearby vertices are adjacent to each other have a sharp spectrum, as seen in complex networks. Furthermore, we exemplify that many graphs where vertices are randomly adjacent to each other have a semicircular spectral distribution as seen in Erdős-Rényi random graphs.

論文要旨

Spectral graph theory ではグラフの性質と固有値の関係について研究されるなど、ネットワークの性質や構造を分析する上でスペクトルは非常に重要な指標となり得る。しかし、大規模ネットワークについては、計算量の問題からスペクトルの一部を利用した分析しか行われていなかった。大規模ネットワークの全てのスペクトルを利用した研究はまだ未発展の分野である。

まず、本研究では大規模ネットワークにスケールするような、固有値の累乗の合計値 $\sum_{i=0}^{n-1} \lambda_i^k$ をもとめる近似アルゴリズムを 2 つ提案する。固有値の累乗の合計値は重要な性質を持っており、近年は関係した不等式の研究なども行われている。しかし、固有値の累乗の合計値を計算する従来の単純なアルゴリズムでは大規模グラフにスケールしなかった。本研究の提案手法により実用的な時間で、数百万や数十億単位の頂点を持つネットワークについて固有値の累乗の合計値を計算することが可能になった。これにより、ネットワークに対する新たな指標として、固有値の累乗の合計値というものが利用可能となる。

本研究は、提案手法と同様の手法に基づいたアルゴリズムによって得ることができるスペクトル分布についても研究する。近年スペクトル全体の分布を計算する手法が考案されたが、実世界のネットワークのスペクトル分布を可視化した研究はまだ少ないため、スペクトル分布とネットワークの性質の対応関係についてはまだ深く考えられていない。我々はまず、スペクトル分布の定量的評価とその応用方法について議論する。スペクトル分布の定量的な評価や比較のために、離散確率分布に対するダイバージェンスや距離の概念を流用することを提案する。応用方法として cosine distance を用いた類似のスペクトルを高速に検索する手法についても言及する。次に、合成されたグラフと実世界のネットワークから得られたスペクトル分布を提示する。次に、得られたスペクトル分布を分析し、ネットワークの性質とスペクトルの関係について考察する。結果として、疎なネットワークについては、モチーフと呼ばれるネットワークの局所的な構造と固有値の分布が関係していることが分かった。また、近くの頂点同士が隣接するようなグラフは、複雑ネットワークに見られるような鋭いスペクトルを持つ。さらに、頂点同士がランダムに隣接する多くのグラフは、Erdős-Rényi ランダムグラフに見られる半円状のスペクトルになることを例示する。

Acknowledgements

Contents

Chapter 1

Introduction

1.1 Large-Scale Network

Large-scale networks appear in many places in our daily lives. For example, IP address networks, map graphs, web graphs, and so on. Algorithms that have been used in the past do not scale to the huge networks that appear in the real world. Therefore, research is being done to develop algorithms that can be applied to large-scale networks and to elucidate the properties of large-scale networks.

1.2 Spectra and Graphs

Spectral graph theory is one powerful tool for understanding the properties of networks. This theory is based on the fact that the properties of a network and its eigenvalues, when expressed as a matrix, are interrelated. However, when considering eigenvalues for real-world networks, only a subset of the eigenvalues have been analyzed due to computational complexity issues, since the computational complexity of obtaining all the eigenvalues is $O(n^3)$ when the number of vertices is n. In recent years, approximation algorithms that use all eigenvalues for analysis have been proposed, but this is still an undeveloped field.

1.2.1 Sum of Powers of Eigenvalues

1.2.2 Spectral Distributions

Recently, an approximation algorithm for the distribution of eigenvalues that scales to large graphs has been proposed. This has made it possible to visualize the distribution of eigenvalues of various networks in the real world. However, a comprehensive analysis of the synthesized graphs and real-world networks has not yet been done.

1.3 Our Contributions

We study the spectral distribution and the sum of powers of eigenvalues as an analysis using all eigenvalues. First, we propose an approximation algorithm for the sum of powers of eigenvalues that scales to large networks. By applying methods such as stochastic trace estimator and stochastic Lanczos quadrature, the sum of powers of eigenvalues can be calculated in a practical time. Experimental results for the analysis of relative errors are also shown.

Second, we discuss the use of the spectral distribution obtained by the method of Kun Dong et al [8]. Although it is currently used for visualization of the distribution, various concepts of a discrete probability distribution can be used

for quantitative evaluation. Next, we show the results of an experiment in which we obtained the spectral distribution for a real-world network. It can be inferred from the experimental results that networks with similar properties have similar spectral distributions. The results of this experiment will be a stepping stone to understanding the properties of networks using spectral distribution.

Chapter 2

Preliminaries

2.1 General Notation

Let $G = (V, E, w)$ be a wighted undirected graph where $V = (v_0, v_2, ..., v_{n-1})$ is a set of vertices, $E \subseteq V \times V$ is a set of undirected edges, and $w : E \to \mathbb{R}$ is a weight function. The adjacency matrix $\mathbf{A}$ is a $n \times n$ matrix of G, (i.e., $A_{ij} = w(v_i, u_i)$).

The laplacian matrix $\mathbf{L}$ and the normalized laplacian matrix $\mathcal{L}$ are defined by the weight function w and the degree $d_v := \sum_u w(u, v)$.

$$
L(u, v) = \begin{cases} d_v - w(u, v) & \text{if } u = v, \\ -w(u, v) & \text{if } u \text{ and } v \text{ are adjacent}, \\ 0 & \text{otherwise}. \end{cases}
$$

$$
\mathcal{L}(u, v) = \begin{cases} 1 - \frac{w(u,v)}{d_v} & \text{if } u = v, \\ -\frac{w(u,v)}{\sqrt{d_u d_v}} & \text{if } u \text{ and } v \text{ are adjacent}, \\ 0 & \text{otherwise}. \end{cases}
$$

Let $\mathbf{M}$ denote a real symmetric matrix that can be $\mathbf{A}$, $\mathbf{L}$, or $\mathcal{L}$.

Symbol	Description
$n \in \mathbb{N}$	$|V|$:the number of vertices
$m \in \mathbb{N}$	$|E|$:the number of edges
$\mathbf{M} \in \mathbb{R}^{n \times n}$	Real symmetric matrix which can be $\mathbf{A}$, $\mathbf{L}$, or $\mathcal{L}$
$\mathbf{A} \in \mathbb{R}^{n \times n}$	Adjacency matrix of a graph G
$\mathbf{L} \in \mathbb{R}^{n \times n}$	Laplacian matrix of a graph G
$\mathcal{L} \in \mathbb{R}^{n \times n}$	Normalized Laplacian matrix of a graph G
$n_v \in \mathbb{R}^+$	Number of SLQ random vectors
$s \in \mathbb{R}^+$	Number of Lanczos algorithm steps
$\mathbf{x} \in \mathbb{R}^n$	Random vector which satisfies $\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}\mathbf{x}^{\mathbf{T}}] = \mathbf{I}$

Table 2.1: Summary of notion

2.2 The Laplacian and Eigenvalues

The relationship between graphs and eigenvalues has been studied in spectral graph theory. A particular focus has been on eigenvalues of normalized laplacian matrices, and Fan Chung has written a comprehensive textbook on the subject[7], but there are also studies on laplacian matrices.

Let $\mathbf{D}$ denote the diagonal matrix, such that

$$\mathbf{D}(u, v) = \begin{cases} d_v & \text{if } u = v, \\ 0 & \text{otherwise.} \end{cases}$$

There are some relationships among the adjacency matrix, laplacian matrix, and normalized laplacian matrix.

By definitions, we can write

$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}}$$
$$= \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$$

Theorem 1. *If G has no negative edges, $\mathbf{L}$ and $\mathcal{L}$ are positive semi-definite.*

Proof. For all $\mathbf{x} \in \mathbb{R}^n$,

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{(u,v) \in E} w(u, v)(x_u - x_v)^2 \geq 0$$

$$\mathbf{x}^T \mathcal{L} \mathbf{x} = \sum_{(u,v) \in E} w(u, v) \left(\frac{x_u}{\sqrt{d_u}} - \frac{x_v}{\sqrt{d_v}} \right)^2 \geq 0$$

$\square$

2.3 Matrix and Spectra

Firstly, we define some concepts regarding matrix and its eigenvalues and prove some theorems which are used later.

Definition 1. *Let $\lambda_0 \leq \lambda_2 \leq ... \leq \lambda_{n-1}$ be the eigenvalues of a n-dimensional matrix $\mathbf{M}$. The spectrum $\sigma(\mathbf{M})$ of $\mathbf{M}$ is the set of its overall eigenvalues.*

$$\sigma(\mathbf{M}) := \{\lambda_i | i = 0, 1, ..., n - 1\}$$

Definition 2. *The spectral radius $\rho(\mathbf{M})$ of a n-dimensional matrix $\mathbf{M}$ is the nonnegative number*

$$\rho(\mathbf{M}) := \sup_{\lambda \in \sigma(A)} |\lambda|.$$

Theorem 2. *Let $\lambda_0 \leq \lambda_2 \leq ... \leq \lambda_{n-1}$ be the eigenvalues of a real symmetric matrix $\mathbf{M}$ and the eigendecomposition of $\mathbf{M}$ be $\mathbf{M} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T$. Then $\lambda_0^k, \lambda_2^k, ..., \lambda_{n-1}^k$ are eigenvalues of $\mathbf{M}^k$.*

Proof.

$$\mathbf{M}^k = (\mathbf{Q} \mathbf{\Lambda} \mathbf{Q^T})^k$$
$$= (\mathbf{Q} \mathbf{\Lambda} \mathbf{Q^{-1}})^k \qquad \mathbf{Q} \text{ is a orthogonal matrix}$$
$$= \mathbf{Q} \mathbf{\Lambda^k} \mathbf{Q^{-1}}$$
$$= \mathbf{Q} \mathbf{\Lambda^k} \mathbf{Q^T}$$

$\square$

Theorem 3. *Let the eigendecomposition of a real symmetric matrix $\mathbf{M}$ be $\mathbf{M} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. If $f(x)$ is given by $f(x) = \sum_{k=0}^{\infty} a_k x^k$ with radius of convergence greater than the spectral radius $\rho(\mathbf{M})$ and $f(\mathbf{M})$ is defined by $f(\mathbf{M}) = \sum_{k=0}^{\infty} a_k \mathbf{M}^k$, then*

$$f(\mathbf{M}) = \mathbf{Q}f(\mathbf{\Lambda})\mathbf{Q}^T.$$

Proof. $\mathbf{\Lambda}$ is a diagonal matrix. Therefore $f(\mathbf{\Lambda})$ is easy to calculate:

$$[f(\mathbf{\Lambda})]_{ij} = \begin{cases} f(\lambda_i) & (i = j) \\ 0 & (otherwise) \end{cases}$$

By applying theorem 2 to each degree, we can get $f(\mathbf{M}) = \mathbf{Q}f(\mathbf{\Lambda})\mathbf{Q}^T$. $\qquad\square$

2.4 Approximation Methods for Matrix Traces and Bilinear Forms

2.4.1 Stochastic Trace Estimator

For trace estimation of large implicit matrices, we can use randomized estimator described by Hunchinson [11, 1].

Theorem 4. *(Proposition 4.1 [1]) Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a square matrix and $\mathbf{x} \in \mathbb{R}^n$ be a random vector such that $\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}\mathbf{x}^T] = \mathbf{I}$. Then $\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}^T\mathbf{M}\mathbf{x}] = \text{tr}(\mathbf{M})$.*

Proof.

$$\begin{aligned} \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}^T\mathbf{M}\mathbf{x}] &= \mathbb{E}_{p(\mathbf{x})}[\text{tr}(\mathbf{x}^T\mathbf{M}\mathbf{x})] \\ &= \mathbb{E}_{p(\mathbf{x})}[\text{tr}(\mathbf{M}\mathbf{x}\mathbf{x}^T)] && \text{invariance to cyclic permutation} \\ &= \text{tr}(\mathbb{E}_{p(\mathbf{x})}[\mathbf{M}\mathbf{x}\mathbf{x}^T]) && \text{linearity of trace} \\ &= \text{tr}(\mathbf{M}\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}\mathbf{x}^T]) && \text{linearity of expectation} \\ &= \text{tr}(\mathbf{M}) \end{aligned}$$

$\qquad\square$

The requirement $\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}\mathbf{x}^T] = \mathbf{I}$ for the random vector $\mathbf{x}$ can be easily satisfied. For example, paractical choices of $p(\mathbf{x})$ include Ramemacher or standard normal distributions with zero mean and one variance.

The following corollary holds.

Corollary 4.1. *If $f(\mathbf{M}) \in \mathbb{R}^{n \times n}$ is a square matrix, and $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{n_v} \in \mathbb{R}^n$ are random vectors drawn from a distribution $p(\mathbf{x})$ such that $\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}\mathbf{x}^T] = \mathbf{I}$, then*

$$\text{tr}(f(\mathbf{M})) = \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}^T f(\mathbf{M})\mathbf{x}] \approx \frac{1}{n_v} \sum_{i=0}^{n_v - 1} \mathbf{x}_i^T f(\mathbf{M})\mathbf{x}_i$$

Unit vectors $\mathbf{z}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2}$ can be used instead of $\mathbf{x}_i$. We can rewrite the formula of Corollary 4.1.

Theorem 5. *Let $\mathbf{z}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2}$ be a unit vector, then*

$$\text{tr}(f(\mathbf{M})) \approx \frac{n}{n_v} \sum_{i=0}^{n_v - 1} \mathbf{z}_i^T f(\mathbf{M})\mathbf{z}_i$$

Proof.

$$\text{tr}(f(\mathbf{M})) \approx \frac{1}{n_v} \sum_{i=0}^{n_v-1} \mathbf{x}_i^T f(\mathbf{M}) \mathbf{x}_i$$

$$= \frac{1}{n_v} \sum_{i=0}^{n_v-1} \|\mathbf{x}_i\|_2^2 \cdot \mathbf{z}_i^T f(\mathbf{M}) \mathbf{z}_i$$

$$\approx \frac{1}{n_v} \sum_{i=0}^{n_v-1} n \cdot \mathbf{z}_i^T f(\mathbf{M}) \mathbf{z}_i \qquad \mathbb{E}_{p(\mathbf{x})}[\|\mathbf{x}_i\|_2^2] \text{ is equal to } n$$

$$= \frac{n}{n_v} \sum_{i=0}^{n_v-1} \mathbf{z}_i^T f(\mathbf{M}) \mathbf{z}_i$$

$\square$

For Randemacher vectors, $\|\mathbf{x}_i\|_2^2$ is always n. For other random vectors, $\mathbb{E}_{p(\mathbf{x})}[\|\mathbf{x}_i\|_2^2]$ is equal to n as long as $\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}\mathbf{x}^T] = \mathbf{I}$.

For accurate computation, n_v should approaches infinity. However using large n_v is expensive. For practical purposes, we can use small n_v. The convergence rate of such estimator was studied by Avron and Toledo[2].

2.4.2 Estimation of bilinear form $\mathbf{z}^T f(\mathbf{M})\mathbf{z}$

There are relationships between matrix, orthogonal polynomials, quadrature rules and the Lanczos algorithms. Gene H Golub and Gérard Meurant compiled a book of previous research on such relationships and on estimation and bounds of bilinear form $\mathbf{u}^T f(\mathbf{M})\mathbf{v}$ [10]. In this paper, we focus on a special case that $\mathbf{u} = \mathbf{v} = \mathbf{z}$.

Firstly, the bilinear form $\mathbf{z}^T f(\mathbf{M})\mathbf{z}$ is reformed as a Riemann-Stieltjes integral and approximated by applying Gauss quadrature rule. Then, the Lanczos algorithm is applied for calculation.

The Riemann–Stieltjes integral is a generalization of the Riemann integral. It is defined to be the limit of a sum.

Definition 3. *A Riemann–Stieltjes integral of a real valued function f of a real variable on the infinite interval $[a, b]$ with respect to a real function α is denoted by*

$$\int_a^b f(x) d\alpha(x).$$

This integral is defined to be the limit, as the length of the subinterval of the partition $\pi = \{a = x_0 < x_1 < ... < x_n = b\}$ goes to zero, of the approximating sum

$$\sum_{x_i \in \pi} f(c_i)(\alpha(x_{i+1}) - \alpha(x_i)).$$

where c_i is in the i-th subinterval $[x_i, x_{i+1}]$.

Now, we consider the approximation of this integral by Gauss quadrature rule. This is a relation:

$$I[f] = \int_a^b f(x) d\alpha(x) = \sum_{k=0}^{s-1} w_k f(\theta_k) + R[f] \qquad (2.1)$$

6

where $\{w_k\}$ are the unknowns weights and $\{\theta_k\}$ are the unknowns nodes. In the right-hand side, the sum $\sum_{k=0}^{s-1} w_k f(\theta_k)$ is the approximation of the Riemann-Stieltjes integral $I[f]$, and $R[f]$ is the remainder. If $R[p] = 0$ for all polynomials p of degree d and $R[q] \neq 0$ for any polynomials q of degree $d+1$, the rule is called *exact*.

The value of $R[f]$ is known [10, 19]. If the measure $\alpha(x)$ is a positive nondecreasing function and $f \in C^{2s}[a, b]$, then

$$R[f] = \frac{f^{2s}(\xi)}{(2s)!} \int_a^b \left[\prod_{k=0}^{s-1} (x - \theta_k) \right]^2 d\alpha(x) \tag{2.2}$$

for some $\xi \in (a, b)$. Therefore, the Gauss rule is exact for polynomials f of degree less than $2s$.

Finally, we can reform and compute the bilinear form $\mathbf{z}^T f(\mathbf{M})\mathbf{z}$ using the Gauss quadrature rule.

$$\begin{aligned}
\mathbf{z}^T f(\mathbf{M})\mathbf{z} &= \mathbf{z}^T \mathbf{Q} f(\mathbf{\Lambda}) \mathbf{Q}^T \mathbf{z} \\
&= \sum_{j=0}^{n-1} f(\lambda_i) \mu_j^2 \\
&= \int_a^b f(t) d\mu(t) && \text{Riemann–Stieltjes integral} \\
&\approx \sum_{k=0}^{s-1} \omega_k f(\theta_k) && \text{the Gauss quadrature rule}
\end{aligned}$$

where $\mu_j = [\mathbf{Q}^T \mathbf{z}]_j$ and $\mu(t)$ is a piecewise constant function defined as

$$\mu(t) = \begin{cases} 0 & \text{if } t < a = \lambda_0 \\ \sum_{j=0}^{i-1} \mu_j^2 & \text{if } \lambda_{i-1} \leq t < \lambda_i, i = 1, \cdots, n-1 \\ \sum_{j=0}^{n-1} \mu_j^2 & \text{if } b = \lambda_{n-1} \leq t \end{cases}$$

and $\{\omega_k\}$ are the weights and $\{\theta_k\}$ are the nodes of the s-point Gauss quadrature.

One choise for computing the nodes and the weights of the Gauss quadrature rule is the Lanczos algorithm. Let $\mathbf{M}$ be a real symmetric matrix, $\mathbf{w}_0$ be an arbitrary starting unit-vector, and $\mathcal{K}$ be the Krylov subspace spanning vectors $\{\mathbf{w}_0, \mathbf{M}\mathbf{w}_0, \cdots, \mathbf{M}^{s-1}\mathbf{w}_0\}$. The output of the Lanczos algorithm are an $n \times s$ matrix $\mathbf{W}$ and an $s \times s$ tridiagonal matrix $\mathbf{T}$, such that $\mathbf{W}^T \mathbf{M} \mathbf{W} = \mathbf{T}$. $\mathbf{W}$ has the orthonormal columns:

$$w_0 : \text{an initial unit-vector}$$
$$w_k = p_k(\mathbf{M})w_0, \ k = 1, \cdots, s-1$$

where p_k are the Lanczos polynomials which are orthogonal with respect to the measure $\mu(t)$.

Now, the bilinear form $\mathbf{z}^T f(\mathbf{M})\mathbf{z}$ is reformed as

$$\mathbf{z}^T f(\mathbf{M})\mathbf{z} \approx \sum_{k=0}^{s-1} \tau_k^2 f(\theta_k) \tag{2.3}$$

$$\tau_k = \mathbf{Y}_{0,k} = \mathbf{e}_1^T \mathbf{y}_k, \ \theta_k = \Theta_{k,k}, \ \mathbf{T} = \mathbf{Y}\Theta\mathbf{Y}^T \tag{2.4}$$

2.4.3 Stochastic Lanczos Quadrature(SLQ)

There is a method called stochastic Lanczos quadrature(SLQ) [22], for approximate computing of the trace of functions of large matrices. Firstly, the stochastic trace estimator is used for approximating the trace. Next, the estimation method we discussed in the section 2.4.2 is applied.

The stochastic trace estimator has been explained in the previous section.

$$\text{tr}(f(\mathbf{M})) = \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}^T f(\mathbf{M})\mathbf{x}] \approx \frac{n}{n_v} \sum_{i=0}^{n_v-1} \mathbf{z}_i^T f(\mathbf{M})\mathbf{z}_i \tag{2.5}$$

Now, we need to compute $\mathbf{z}_i^T f(\mathbf{M})\mathbf{z}_i$. By applying (2.1) to the stochastic trace estimator (2.3), we get Stochastic Lanczos Quadrature estimator:

$$\text{tr}(f(\mathbf{M})) \approx \frac{n}{n_v} \sum_{i=0}^{n_v-1} \left(\sum_{k=0}^{s-1} (\tau_k^i)^2 f(\theta_k^i) \right) \tag{2.6}$$

2.5 Spectral Distribution

In the field of condensed matter physics, the density of states is defined as the distribution of eigenvalues. Calculating all the eigenvalues takes $O(n^3)$ when the number of vertices is n, so calculating the exact solution is expensive. Therefore, it does not scale well for large-scale networks with millions or billions of vertices. Recently, Kun Dong et al.[8] have developed an approximation algorithm for finding the distribution of eigenvalues in a network . This has made it possible to obtain the distribution of eigenvalues for large graphs in a practical time.

First, we define DOS as the spectrum of the entire network. Next, we define LDOS as the local spectrum, and define what the equation looks like when we consider the spectrum of each vertex as its special case.

Definition 4. *Let* $\mathbf{M} \in \mathbb{R}^{n \times n}$ *be any real symmetric matrix. The spectral density, or density of states (DOS) is*

$$\mu(\lambda) = \frac{1}{n} \sum_{i=0}^{n-1} \delta(\lambda - \lambda_i), \quad \int f(\lambda)\mu(\lambda) = \text{tr}(f(\mathbf{M})) \tag{2.7}$$

where δ *is the Dirac delta function and* f *is any analytic test function.*

Definition 5. *For any vector* $u \in \mathbb{R}^n$, *the local density of states (LDOS) is*

$$\mu(\lambda; u) = \sum_{i=0}^{n-1} |u^T q_i|^2 \delta(\lambda - \lambda_i), \quad \int f(\lambda)\mu(\lambda; u) = u^T f(\mathbf{M})u \tag{2.8}$$

where $Q = [q_0, ..., q_{n-1}]$ *is orthogonal. Especially, when we consider the case* $u = e_k$ *where* e_k *is the k-th standard basis vector, the pointwise density of states (PDOS) is*

$$\mu_k(\lambda) = \sum_{i=0}^{n-1} |e_k^T q_i|^2 \delta(\lambda - \lambda_i) = \sum_{i=0}^{n-1} |q_i(k)|^2 \delta(\lambda - \lambda_i), \quad \int f(\lambda)\mu_k(\lambda) = e_k^T f(\mathbf{M})e_k \tag{2.9}$$

where $|q_i(k)|$ *is the magnitude of the weight for a k-th vertex in the i-th eigenvector.*

2.5.1   Kernel Polynomial Method (KPM)

To approximate DOS and PDOS, the Kernel Polynomial Method (KPM) [23] can be used. This method uses a dual basis of orthogonal polynomial basis, and Kun Dong et al.[8] use Chebyshev polynomials. First, the spectral density is expanded by using the dual Chebyshev basis, and the DOS can be approximated by using the stochastic trace estimator. The PDOS can be obtained by using the diagonal elements of the m-th Chebyshev polynomial of $\mathbf{M}$.

When approximating with Chebyshev polynomials, the eigenvalues must be in the interval $[-1, 1]$, and the eigenvalues of the normalized adjacency matrix of unweighted graphs are in this interval. For arbitrary matrices $\mathbf{M}$, this condition can be satisfied by using shifting and rescaling as follows:

$$\widetilde{\mathbf{M}} = \frac{2\mathbf{M} - (\lambda_{n-1} + \lambda_0)}{\lambda_{n-1} - \lambda_0}.$$

The Chebyshev polynomials are defined recursively as follows:

$$T_0(x) = 1, T_1(x) = x, T_{m+1}(x) = 2xT_m(x) - T_{m-1}(x).$$

Let $w(x)$ be $2/[1 + \delta_{0n}\pi\sqrt{1-x^2}]$. Then,

$$\int_{-1}^{1} w(x)T_m(x)T_n(x)dx = \delta_{mn}$$

holds, and the Chebyshev polynomials are orthogonal with respect to $w(x)$. Therefore, we can expand DOS and PDOS into a series as follows

$$\mu(\lambda) = \sum_{m=1}^{\infty} d_m T_m^*(\lambda) \tag{2.10}$$

$$\mu_k(\lambda) = \sum_{m=1}^{\infty} d_{mk} T_m^*(\lambda) \tag{2.11}$$

$$d_m = \int_{-1}^{1} T_m(\lambda)\mu(\lambda)d\lambda = \frac{1}{n}\sum_{i=0}^{n-1} T_m(\lambda_i) = \frac{1}{N}\text{tr}(T_m(\mathbf{M})) \tag{2.12}$$

$$d_{mk} = \int_{-1}^{1} T_m(\lambda)\mu_k(\lambda)d\lambda = \sum_{i=0}^{n-1} |q_i(k)|^2 T_m(\lambda_i) = T_m(\mathbf{M})_{kk} \tag{2.13}$$

where $T_m^* = w(x)T(x)$. From the above equation, we need to extract the diagonal elements of T(M) fast in order to calculate DOS and PDOS. This can be done by using stochastic trace estimator [11] or stochastic diagonal estimator [3]. The trace can be approximated by Corollary 4.1, and there are various ways to choose random vectors [2]. $diag(T_m(\mathbf{M}))$ can be obtained in a similar way.

In practice, it is impossible to expand DOS and PDOS by an infinite number of polynomials, so we must use a finite number of moments. The convergence speed is fast enough so that the number of moments can be small, but such truncation causes Gibbs oscillations. This can be dealt with by using Jackson's smoothing technique [12, 8].

Chapter 3

the Sum of Powers of Eigenvalues of Network (SPENet)

Our main purpose is computing sum of k-th powers of eigenvalues $\sum_{i=0}^{n-1} \lambda_i^k$ which we call SPENet. We propose two approximation algorithms (STE and SLQ) based on the stochastic trace estimator and the stochastic Lanczos quadrature. For naive algorithms, the computational complexity of SPENet using full eigendecomposition is $O(n^3)$, while for STE algorithm it is $O(mkn_v)$ and for SLQ algorithm it is $O((ms + ns^2)n_v)$.

Since the laplacian matrix $\mathbf{L}$ and the normalized laplacian matrix $\mathcal{L}$ have zero eigenvalues, we are interested in the case that index of power k is positive. When we consider the case that $\mathbf{M} = \mathbf{A}$, k must be positive integer or zero because $\mathbf{A}$ have negative eigenvalues.

3.1 Algorithms

3.1.1 STE algorithm

The first estimation method for SPENet is based on the stochastic trace estimator(STE), which we apply in our setting for the trace of power of matrix $f(\mathbf{M}) = \mathbf{M}^k$. We are interested in the case where $\mathbf{M} = \mathbf{A}, \mathbf{L},$ or $\mathcal{L}$. In this algorithm, the index of power k must be positve integer.

By using theorem 2 and 5, we can apply the stochastic trace estimator for computing sum of powers of k-th eigenvalues.

$$\sum_{i=0}^{n-1} \lambda_i^k = \operatorname{tr}(\mathbf{M}^k)$$

$$= \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}^T \mathbf{M}^k \mathbf{x}] \approx \frac{n}{n_v} \sum_{i=0}^{n_v-1} \mathbf{z}_i^T \mathbf{M}^k \mathbf{z}_i \tag{3.1}$$

$$= \frac{n}{n_v} \sum_{i=0}^{n_v-1} \mathbf{z}_i^T \underbrace{\mathbf{M}\mathbf{M}\cdots\cdots\mathbf{M}}_{k \text{ times}} \mathbf{z}_i \tag{3.2}$$

The product of two matrices costs $O(n^3)$, but the product of a vector and a sparse matrix costs only $O(m)$. Therefore, when calculating $\mathbf{z}_i^T \mathbf{M}^k \mathbf{z}_i$, we just calculate the product of a vector and a matrix k times, and the inner product of a vector once from the left. From the above, the cost of the STE algorithm will be $O(mkn_v)$.

3.1.2 SLQ algorithm

The second estimation method for SPENet is based on the stochastic Lanczos quadrature(SLQ), which we apply in our setting for the trace of power of matrix $f(\mathbf{M}) = \mathbf{M}^k$. We are interested in the case where $\mathbf{M} = \mathbf{A}, \mathbf{L}, \text{or} \mathcal{L}$. In this algorithm, the index of power k must be positve real number.

By using theorem 1 and (2.4), we can estimate SPENet.

$$\sum_{i=0}^{n-1} \lambda_i^k = \text{tr}(\mathbf{M}^k)$$

$$= \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}^T \mathbf{M}^k \mathbf{x}] \approx \frac{n}{n_v} \sum_{i=0}^{n_v-1} \mathbf{z}_i^T \mathbf{M}^k \mathbf{z}_i$$

$$\approx \frac{n}{n_v} \sum_{i=0}^{n_v-1} \left(\sum_{j=0}^{s-1} (\tau_j^i)^2 (\theta_j^i)^k \right) \tag{3.3}$$

Since we apply s-step Lanczos algorithm for τ_k^i and θ_k^i, the orthogonalization cost inside the Lanczos algorithm is $O(ns^2)$ and the cost of the product of a vector and a matrix s times in the Lanczos algorithm is $O(ms)$. From the above, all the computational cost of SLQ algorithm is $O((ms + ns^2)n_v)$.

Algorithm 1 STE algorithm

Input: a real symmetric matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$, a positive integer k, and n_v.
Output: Approximate sum of k-th powers of eigenvalues $\sum_{i=0}^{n-1} \lambda_i^k$
  1: **for** $i = 0$ to $n_v - 1$ **do**
  2:    Generate a random vector $\mathbf{x}_i$ with the standard normal distribution.
  3:    Form unit vector $\mathbf{z}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2}$.
  4:    $\mathbf{v}_i^T \leftarrow \mathbf{z}_i^T$.
  5:    **for** $j = 0$ to $k - 1$ **do**
  6:       $\mathbf{v}_i^T \leftarrow \mathbf{v}_i^T \mathbf{M}$.
  7:    **end for**
  8:    $\Gamma \leftarrow \Gamma + \mathbf{v}_i^T \mathbf{z}_i$
  9: **end for**
 10: $\Gamma \leftarrow \frac{n}{n_v} \Gamma$
 11: **return** Γ

Algorithm 2 SLQ algorithm

Input: a real symmetric matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$, a positive real number k, and n_v.
Output: Approximate sum of k-th powers of eigenvalues $\sum_{i=0}^{n-1} \lambda_i^k$
  1: **for** $i = 0$ to $n_v - 1$ **do**
  2:    Generate a random vector $\mathbf{x}_i$ with the standard normal distribution.
  3:    Form unit vector $\mathbf{z}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2}$.
  4:    $T \leftarrow \text{Lanczos}(\mathbf{M}, \mathbf{z}_i, s)$.
  5:    $[\mathbf{Y}, \mathbf{\Theta}] \leftarrow \text{eig}(T)$
  6:    Compute $\tau_j = [e_1^T y_j]$ for $j = 0, ..., s - 1$
  7:    $\Gamma \leftarrow \Gamma + \sum_{j=0}^{s-1} (\tau_j)^2 (\theta_j)^k$
  8: **end for**
  9: $\Gamma \leftarrow \frac{n}{n_v} \Gamma$
 10: **return** Γ

3.2 Conditions of k

In SLQ algorithm, we can generalize the index of power k. If $\mathbf{M}$ is positive definite, we can use not only a positive integer but also a positive real number as k. This is because that $f(x) = x^k$ is analytic inside a closed interval $[\lambda_0, \lambda_{n-1}]$. However the function $f(x) = x^k$ have a singularity at zero and the laplacian matrix $\mathbf{L}$ and the normalized laplacian matrix $\mathcal{L}$ are positive semi-definite matrix, which have $\lambda_0 = 0$.

To overcome the issue mentioned above, we can use *shifting the spectrum* proposed by Shashanka Ubaru, Jie Chen, and Yousef Saad[22]. The idea is shifting the eigenvalues by replacing $\mathbf{M}$ with $\mathbf{M} + \delta\mathbf{I}$. The shifted matrix have the eigenvalues in the interval $[\lambda_0 + \delta, \lambda_{n-1} + \delta]$. Hence we can obtaing the approximation error bounds of SLQ algorithm and find that SLQ algorithm is practically useful.

If k is a positive real number and $\mathbf{M} = \mathbf{A}$, the SPENet $\sum_{i=0}^{n-1} \lambda_i^k$ is undefined because $\mathbf{A}$ have negative eigenvalues.

	$\mathbf{A}$	$\mathbf{L}$	$\mathcal{L}$
$k \in \mathbb{Z}_{\geq 0}$	STE or SLQ	STE or SLQ	STE or SLQ
$k \in \mathbb{R}_{\geq 0}$	undefined	SLQ	SLQ

Table 3.1: Conditions under which the algorithms can be used

3.3 Error bounds

Suppose k is a positive integer and Lanczos step size is s. When $2s - 1$ is greater or equal to the degree of $f(x)$, the error of Gauss quadrature rule is zero.

3.4 Experiments

We evaluate computational time and variance of SPENet using STE and SLQ algorithm. Additionally, we evaluate its relative error against the exact computation of the eigenvalues where allowed by the graph size. We perform our experiments averating 10 times for all experiments unless stated otherwise.

We use 40 real-world small networks[18].

3.4.1 Parameter Settings

3.4.2 Approximation Accuracy

3.4.3 Parameter Sensitivity

Chapter 4

Applications and Gallery of Spectral Distributions

In this chapter, we discuss applications of the spectral distribution defined in Section 2.5. We then show by experiments how the spectral distributions of various synthesized and real-world graphs look in practice.

4.1 Quantitative Evaluation and Application

The DOS and PDOS of a network can be approximated by using the Kernel polynomial method as described in Section 2.5.1. However, its usage has not been discussed in-depth, and only the fact that it can be visualized has been emphasized. By using this distribution as a discrete probability distribution, it will be possible to evaluate it quantitatively using various concepts such as entropy, divergence, distance.

For each of them, we discuss what advantages and disadvantages they have and compare them with previous studies on similar concepts for networks.

4.1.1 Entropy

It is possible to calculate the entropy of a spectral distribution expressed as a discrete probability distribution: the entropy of the entire network can be obtained using DOS, and the entropy of each vertex can be obtained using PDOS.

However, Von Neumann graph entropy, a concept similar to the entropy of the entire network, has already been studied [4, 6, 21]. It is inspired by the Von Neumann entropy of quantum mechanics and has already been studied not only for applications to divergence and distance but also for applications and properties of the entropy itself. Since there are faster computational methods [6, 21] than those proposed by Kun Dong et al [8], there is little advantage in using DOS for entropy . On the other hand, the entropy per-vertex may be worth using. By considering the relationship between the size of the entropy of each vertex, it may be possible to calculate the "importance" of each vertex.

4.1.2 Divergence and Distance

By using DOS expressed as a discrete probability distribution, it is possible to define the divergence and distance between networks. For example, Kullback-Leibler divergence [14] is famous for its divergence, and since Kullback-Leibler divergence is asymmetric, we can also use the symmetric Jeffreys divergence and Jensen-Shannon divergence [17, 16, 13]. It is also possible to use the Jensen-Shannon distance, which was proved to be a valid distance metric [9].

The Von Neumann graph entropy can also be used to define divergence and distance, and various studies have already been conducted on this. In the field of quantum mechanics, divergence and quantum Jensen-Shannon divergence have already been defined [5, 15], and these can be calculated using the Von Neumann graph entropy. However, unlike the method of calculating the divergence and distance using DOS, the divergence and distance obtained via Von Neumann graph entropy cannot be precomputed.

However, the distance called NetLSD has also been defined using the heat kernel, and an approximation algorithm has been established [20, 21]. NetLSD is capable of precomputation and can perform approximate calculations faster than DOS. Therefore, when considering the divergence and distance between networks, there is little advantage in using DOS.

On the other hand, there is an advantage of using PDOS. Usually, the distance between vertices is the length of the shortest path, but with this definition, we can express how similar the properties of the vertices are. This may make it possible to search for vertices with similar properties.

4.1.3 Applications

4.2 Experiments

4.2.1 Gallery of DOS/PDOS

4.2.2 Propaties of Graphs with Similar DOS/PDOS

Chapter 5

Conclusion

References

[1] Ryan P Adams, Jeffrey Pennington, Matthew J Johnson, Jamie Smith, Yaniv Ovadia, Brian Patton, and James Saunderson. Estimating the spectral density of large implicit matrices. *arXiv preprint arXiv:1802.03451*, 2018.

[2] Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2):1–34, 2011.

[3] Costas Bekas, Effrosyni Kokiopoulou, and Yousef Saad. An estimator for the diagonal of a matrix. *Applied numerical mathematics*, 57(11-12):1214–1229, 2007.

[4] Samuel L Braunstein, Sibasish Ghosh, and Simone Severini. The laplacian of a graph as a density matrix: a basic combinatorial approach to separability of mixed states. *Annals of Combinatorics*, 10(3):291–317, 2006.

[5] Jop Briët and Peter Harremoës. Properties of classical and quantum jensen-shannon divergence. *Physical review A*, 79(5):052311, 2009.

[6] Pin-Yu Chen, Lingfei Wu, Sijia Liu, and Indika Rajapakse. Fast incremental von neumann graph entropy computation: Theory, algorithm, and applications. In *International Conference on Machine Learning*, pages 1091–1101, 2019.

[7] Fan RK Chung and Fan Chung Graham. *Spectral graph theory.* Number 92. American Mathematical Soc., 1997.

[8] Kun Dong, Austin R Benson, and David Bindel. Network density of states. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1152–1161, 2019.

[9] Dominik Maria Endres and Johannes E Schindelin. A new metric for probability distributions. *IEEE Transactions on Information theory*, 49(7):1858–1860, 2003.

[10] Gene H Golub and Gérard Meurant. *Matrices, moments and quadrature with applications*, volume 30. Princeton University Press, 2009.

[11] Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

[12] Dunham Jackson. *Über die Genauigkeit der Annäherung stetiger Funktionen durch ganze rationale Funktionen gegebenen Grades und trigonometrische Summen gegebener Ordnung.* Dieterich, 1911.

[13] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

[14] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[15] PW Lamberti, AP Majtey, A Borras, Montserrat Casas, and A Plastino. Metric character of the quantum jensen-shannon divergence. *Physical Review A*, 77(5):052311, 2008.

[16] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[17] Frank Nielsen. On the jensen–shannon symmetrization of distances relying on abstract means. *Entropy*, 21(5):485, 2019.

[18] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.

[19] Josef Stoer and Roland Bulirsch. *Introduction to numerical analysis*, volume 12. Springer Science & Business Media, 2013.

[20] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alexander Bronstein, and Emmanuel Müller. Netlsd: hearing the shape of a graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2347–2356, 2018.

[21] Anton Tsitsulin, Marina Munkhoeva, and Bryan Perozzi. Just slaq when you approximate: Accurate spectral distances for web-scale graphs. In *Proceedings of The Web Conference 2020*, pages 2697–2703, 2020.

[22] Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of tr(f(a)) via stochastic lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017.

[23] Alexander Weiße, Gerhard Wellein, Andreas Alvermann, and Holger Fehske. The kernel polynomial method. *Reviews of modern physics*, 78(1):275, 2006.