Distributed Spectral Decomposition in Networks by Complex Diffusion and Quantum Random Walk

Jithin K. Sreedharan*
joint work with Konstantin Avrachenkov* and Philippe Jacquet†

INFOCOM, 12 April 2016

* **INRIA**, France † **Bell Labs**, France

Introduction

Question we address here

- Symmetric graph matrices like adjacency matrix, Laplacian matrix (undirected graph)

Question we address here

- Symmetric graph matrices like adjacency matrix, Laplacian matrix (undirected graph)
- Eigenvalues: $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$
  Corresponding eigenvectors: $\mathbf{u}_1, \ldots, \mathbf{u}_n$.

Question we address here

- Symmetric graph matrices like adjacency matrix, Laplacian matrix (undirected graph)
- Eigenvalues: $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$
  Corresponding eigenvectors: $\mathbf{u}_1, \ldots, \mathbf{u}_n$.

Problem

A scalable way to find largest k eigenvalues $\lambda_1, \ldots, \lambda_k$ and the eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$.

Why is it relevant? Some uses of graph spectrum and graph eigenvectors

- Number of triangles:

Why is it relevant? Some uses of graph spectrum and graph eigenvectors

- Number of triangles:
  - Total number of triangles in a graph: $\frac{1}{6}\sum_{i=1}^{n} |\lambda_i|^3$.

Why is it relevant? Some uses of graph spectrum and graph eigenvectors

- Number of triangles:
  - Total number of triangles in a graph: $\frac{1}{6}\sum_{i=1}^{n}|\lambda_i|^3$.
  - Number of triangles that a node m participated in: $\frac{1}{2}\sum_{i=1}|\lambda_i^3|\,\mathbf{u}_i(m)$

Why is it relevant? Some uses of graph spectrum and graph eigenvectors

- Number of triangles:
  - Total number of triangles in a graph: $\frac{1}{6}\sum_{i=1}^{n} |\lambda_i|^3$.
  - Number of triangles that a node m participated in: $\frac{1}{2}\sum_{i=1} |\lambda_i^3| \, \mathbf{u}_i(m)$
- Dimensionality reduction, link prediction and Weak and strong ties: Each node is mapped into a point in $\mathbb{R}^k$ space.

Why is it relevant? Some uses of graph spectrum and graph eigenvectors

- Number of triangles:
  - Total number of triangles in a graph: $\frac{1}{6}\sum_{i=1}^{n}|\lambda_i|^3$.
  - Number of triangles that a node m participated in: $\frac{1}{2}\sum_{i=1}|\lambda_i^3|\,\mathbf{u}_i(m)$
- Dimensionality reduction, link prediction and Weak and strong ties: Each node is mapped into a point in $\mathbb{R}^k$ space.
- Finding near-cliques: phenomenon of Eigenspokes in eigenvector-eigenvector scatter plot of adjacency matrix.

Challenges in classical techniques

- Power iteration:

$$\mathbf{b}_{\ell+1} = \frac{1}{\|\mathbf{b}_\ell\|}\mathbf{A}\mathbf{b}_\ell \qquad \Longrightarrow$$

Challenges in classical techniques

- Power iteration:

$$\mathbf{b}_{\ell+1} = \frac{1}{\|\mathbf{b}_\ell\|}\mathbf{A}\mathbf{b}_\ell \qquad \Longrightarrow \qquad \begin{aligned} \lambda_1 &= \lim_{k\to\infty} \frac{\mathbf{b}_{k+1}\mathbf{b}_k^\intercal}{\|\mathbf{b}_k\|} \\ \mathbf{u}_1 &= \lim_{k\to\infty} \frac{\mathbf{b}_k}{\|\mathbf{b}_k\|} \end{aligned}$$

Drawback: Only principal components, orthonormalization

Challenges in classical techniques

- Power iteration:

$$\mathbf{b}_{\ell+1} = \frac{1}{\|\mathbf{b}_\ell\|}\mathbf{A}\mathbf{b}_\ell \qquad \Longrightarrow \qquad \begin{aligned} \lambda_1 &= \lim_{k\to\infty} \frac{\mathbf{b}_{k+1}\mathbf{b}_k^\intercal}{\|\mathbf{b}_k\|} \\ \mathbf{u}_1 &= \lim_{k\to\infty} \frac{\mathbf{b}_k}{\|\mathbf{b}_k\|} \end{aligned}$$

  Drawback: Only principal components, orthonormalization
- Inverse iteration method:

$$\mathbf{b}_{\ell+1} = \frac{1}{\|\mathbf{b}_\ell\|}(\mathbf{A}-\mu\mathbf{I})^{-1}\mathbf{b}_\ell \Longrightarrow$$

Challenges in classical techniques

- Power iteration:

$$\mathbf{b}_{\ell+1} = \frac{1}{\|\mathbf{b}_\ell\|}\mathbf{A}\mathbf{b}_\ell \qquad \Longrightarrow \qquad \begin{aligned} \lambda_1 &= \lim_{k\to\infty} \frac{\mathbf{b}_{k+1}\mathbf{b}_k^\intercal}{\|\mathbf{b}_k\|} \\ \mathbf{u}_1 &= \lim_{k\to\infty} \frac{\mathbf{b}_k}{\|\mathbf{b}_k\|} \end{aligned}$$

Drawback: Only principal components, orthonormalization

- Inverse iteration method:

$$\mathbf{b}_{\ell+1} = \frac{1}{\|\mathbf{b}_\ell\|}(\mathbf{A}-\mu\mathbf{I})^{-1}\mathbf{b}_\ell \Longrightarrow \begin{aligned} &\text{Closest eigenvalue to } \mu\text{: } \lim_{k\to\infty} \mu + \frac{\|\mathbf{b}_k\|}{\mathbf{b}_{k+1}\mathbf{b}_k^\intercal} \\ &\text{Eigenvector: } \lim_{k\to\infty} \frac{\mathbf{b}_k}{\|\mathbf{b}_k\|} \end{aligned}$$

Drawback: Inverse calculation, orthonormalization

Our contribution

Our contribution

- Distributed way to find the spectrum
- At each node, eigenvalues correspond to the frequencies of spectral peaks and respective eigenvector components are the amplitudes at those points.

Our contribution

- Distributed way to find the spectrum
- At each node, eigenvalues correspond to the frequencies of spectral peaks and respective eigenvector components are the amplitudes at those points.
- Idea of Complex Power Iterations

Our contribution

- Distributed way to find the spectrum
- At each node, eigenvalues correspond to the frequencies of spectral peaks and respective eigenvector components are the amplitudes at those points.
- Idea of **Complex Power Iterations**
- Diffusion algorithms, Monte Carlo techniques and Random Walk implementation

Our contribution

- Distributed way to find the spectrum
- At each node, eigenvalues correspond to the frequencies of spectral peaks and respective eigenvector components are the amplitudes at those points.
- Idea of **Complex Power Iterations**
- Diffusion algorithms, Monte Carlo techniques and Random Walk implementation
- Connection with Quantum random walks

Our contribution

- Distributed way to find the spectrum
- At each node, eigenvalues correspond to the frequencies of spectral peaks and respective eigenvector components are the amplitudes at those points.
- Idea of **Complex Power Iterations**
- Diffusion algorithms, Monte Carlo techniques and Random Walk implementation
- Connection with Quantum random walks
- Simulation on real-world networks of varying sizes

Complex Power Iterations

Central idea

- Approach based on complex numbers.
- Let $\mathbf{b}_t = e^{i\mathbf{A}t}\mathbf{b}_0$, solution of $\frac{\partial}{\partial t}\mathbf{b}_t = i\mathbf{A}\mathbf{b}_t$.

Central idea

- Approach based on complex numbers.
- Let $\mathbf{b}_t = e^{i\mathbf{A}t}\mathbf{b}_0$, solution of $\frac{\partial}{\partial t}\mathbf{b}_t = i\mathbf{A}\mathbf{b}_t$.
  Harmonics of $\mathbf{b}_t$ corresponds to eigenvalues.

Central idea

- Approach based on complex numbers.
- Let $\mathbf{b}_t = e^{i\mathbf{A}t}\mathbf{b}_0$, solution of $\frac{\partial}{\partial t}\mathbf{b}_t = i\mathbf{A}\mathbf{b}_t$.
  Harmonics of $\mathbf{b}_t$ corresponds to eigenvalues.
- Details: from spectral theorem,

$$\frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{i\mathbf{A}t}e^{-it\theta}dt \quad = \quad \sum_{j=1}^{n}\delta_{\lambda_j}(\theta)\mathbf{u}_j\mathbf{u}_j^\intercal$$

Smoothing and a sample plot

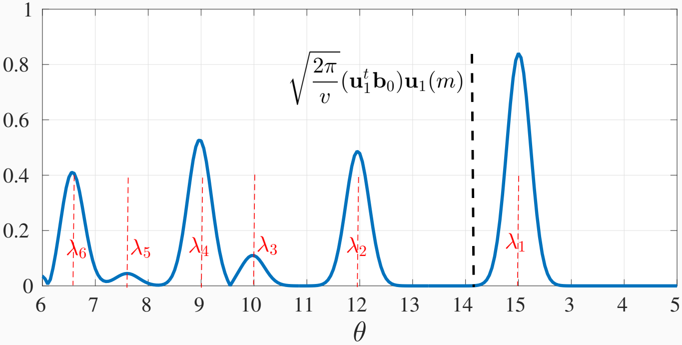
Idea of Gaussian smoothing:

$$\frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{i\mathbf{A}t}\mathbf{b}_0 e^{-t^2 v/2} e^{-it\theta} dt \quad = \quad \sum_{j=1}^{n} \frac{1}{\sqrt{2\pi v}} \exp\left(-\frac{(\lambda_j-\theta)^2}{2v}\right)\mathbf{u}_j(\mathbf{u}_j^\intercal \mathbf{b}_0)$$

Smoothing and a sample plot

Idea of Gaussian smoothing:

$$\frac{1}{2\pi}\int_{-\infty}^{+\infty} e^{i\mathbf{A}t}\mathbf{b}_0 e^{-t^2 v/2} e^{-it\theta} dt \quad = \quad \sum_{j=1}^{n} \frac{1}{\sqrt{2\pi v}} \exp(-\frac{(\lambda_j-\theta)^2}{2v})\mathbf{u}_j(\mathbf{u}_j^\intercal \mathbf{b}_0)$$



Sample plot at an arbitrary node m

Computing the integral

Discretization:

$$\mathbf{f}_\theta = \varepsilon \Re\Big(\mathbf{b}_0 + 2 \sum_{\ell=1}^{d_{\max}} e^{-i\ell\varepsilon\theta} e^{-\ell^2\varepsilon^2 v/2} \mathbf{x}_\ell\Big),$$

where $\mathbf{x}_\ell$ is approximation of $e^{i\varepsilon\ell\mathbf{A}}\mathbf{b}_0$.

Computing the integral

Discretization:

$$\mathbf{f}_\theta = \varepsilon \Re\Big(\mathbf{b}_0 + 2\sum_{\ell=1}^{d_{\max}} e^{-i\ell\varepsilon\theta} e^{-\ell^2\varepsilon^2 v/2}\mathbf{x}_\ell\Big),$$

where $\mathbf{x}_\ell$ is approximation of $e^{i\varepsilon\ell\mathbf{A}}\mathbf{b}_0$.
Approximations:

- First order: $e^{i\mathbf{A}\ell\varepsilon} = (\mathbf{I} + i\varepsilon\mathbf{A})^\ell(1 + O(\varepsilon^2\ell))$

Computing the integral

Discretization:

$$\mathbf{f}_\theta = \varepsilon \Re\Big(\mathbf{b}_0 + 2\sum_{\ell=1}^{d_{\max}} e^{-i\ell\varepsilon\theta} e^{-\ell^2\varepsilon^2 v/2}\mathbf{x}_\ell\Big),$$

where $\mathbf{x}_\ell$ is approximation of $e^{i\varepsilon\ell\mathbf{A}}\mathbf{b}_0$.

Approximations:

- First order: $e^{i\mathbf{A}\ell\varepsilon} = (\mathbf{I} + i\varepsilon\mathbf{A})^\ell(1 + O(\varepsilon^2\ell))$
- Higher order: Numerical solution to $\frac{\partial}{\partial t}\mathbf{b}_t = i\mathbf{A}\mathbf{b}_t$ with $\mathbf{b}_0$ as the initial value. Use Runge-Kutta methods.

Computing the integral

Discretization:

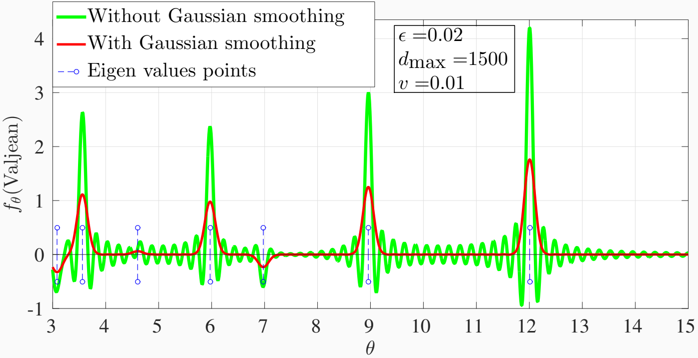
$$\mathbf{f}_\theta = \varepsilon \Re\Big(\mathbf{b}_0 + 2\sum_{\ell=1}^{d_{\max}} e^{-i\ell\varepsilon\theta} e^{-\ell^2\varepsilon^2 v/2}\mathbf{x}_\ell\Big),$$

where $\mathbf{x}_\ell$ is approximation of $e^{i\varepsilon\ell\mathbf{A}}\mathbf{b}_0$.
Approximations:

- First order: $e^{i\mathbf{A}\ell\varepsilon} = (\mathbf{I} + i\varepsilon\mathbf{A})^\ell(1 + O(\varepsilon^2\ell))$
- Higher order: Numerical solution to $\frac{\partial}{\partial t}\mathbf{b}_t = i\mathbf{A}\mathbf{b}_t$ with $\mathbf{b}_0$ as the initial value. Use Runge-Kutta methods. Order-r RK method is **equivalent** to

$$\mathbf{x}_\ell = \left(\sum_{j=0}^{r} \frac{(i\varepsilon\mathbf{A})^j}{j!}\right)^\ell \mathbf{b}_0$$

Gaussian smoothing



Effect of Gaussian smoothing

Complex Diffusion

Different approaches

1. **Centralized approach**: Adjacency matrix $\mathbf{A}$ is fully known

Different approaches

1. **Centralized approach**: Adjacency matrix $\mathbf{A}$ is fully known
2. **Complex diffusion**: Distributed and asynchronous. Only local information available, communicates with all the neighbors

Different approaches

1. **Centralized approach**: Adjacency matrix $\mathbf{A}$ is fully known
2. **Complex diffusion**: Distributed and asynchronous. Only local information available, communicates with all the neighbors
3. **Monte Carlo techniques**: Only local information, but communicates with only one neighbor.

Complex diffusion Order-1

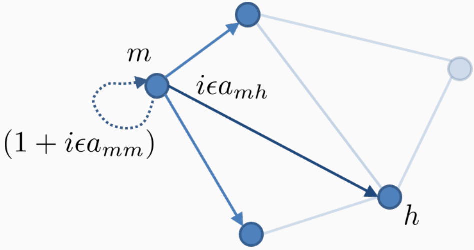
1. Initialize node m with $\mathbf{b}_0(m)$
2. Move weighted copy of fluid to all neighbors and to itself:

$$i\varepsilon a_{m,h}\mathbf{b}_k(m) \text{ to } h \in \mathcal{N}(m)$$
$$(1 + i\varepsilon a_{mm})\mathbf{b}_k(m) \text{ to } m$$

Complex diffusion Order-1

1. Initialize node m with $\mathbf{b}_0(m)$
2. Move weighted copy of fluid to all neighbors and to itself:

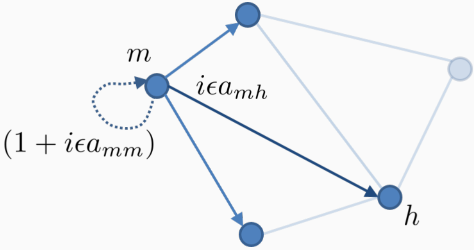
$$i\varepsilon a_{m,h}\mathbf{b}_k(m) \text{ to } h \in \mathcal{N}(m)$$
$$(1 + i\varepsilon a_{mm})\mathbf{b}_k(m) \text{ to } m$$



Complex diffusion Order-1

1. Initialize node m with $\mathbf{b}_0(m)$
2. Move weighted copy of fluid to all neighbors and to itself:

$$i\varepsilon a_{m,h}\mathbf{b}_k(m) \text{ to } h \in \mathcal{N}(m)$$
$$(1 + i\varepsilon a_{mm})\mathbf{b}_k(m) \text{ to } m$$



- Fluid is a complex value

Complex diffusion Order-1

1. Initialize node m with $\mathbf{b}_0(m)$
2. Move weighted copy of fluid to all neighbors and to itself:

$$i\varepsilon a_{m,h}\mathbf{b}_k(m) \text{ to } h \in \mathcal{N}(m)$$
$$(1 + i\varepsilon a_{mm})\mathbf{b}_k(m) \text{ to } m$$

- Fluid is a complex value
- Computations can be distributed

Complexity

Inverse power iteration: For each λ
delay = $D + 2Dd_{\max}$
no. of packets = $|E|n^2 + (n|E| + |E|)d_{\max}$
Complex power iteration: For all λ
delay = $D + d_{\max}$
no. of packets = $|E|d_{\max} + n|E|$.

Complex diffusion Order-1

1. Initialize node m with $\mathbf{b}_0(m)$
2. Move weighted copy of fluid to all neighbors and to itself:

$$i\varepsilon a_{m,h}\mathbf{b}_k(m) \text{ to } h \in \mathcal{N}(m)$$
$$(1 + i\varepsilon a_{mm})\mathbf{b}_k(m) \text{ to } m$$

- Fluid is a complex value
- Computations can be distributed
- Higher order approximations

Complexity

Inverse power iteration: For each λ
delay = $D + 2Dd_{\max}$
no. of packets = $|E|n^2 + (n|E| + |E|)d_{\max}$
Complex power iteration: For all λ
delay = $D + d_{\max}$
no. of packets = $|E|d_{\max} + n|E|$.

Order-r

Compute $\mathbf{A}^r\mathbf{b}_0$ distributedly

Complex diffusion Order-1

1. Initialize node m with $\mathbf{b}_0(m)$
2. Move weighted copy of fluid to all neighbors and to itself:

$$i\varepsilon a_{m,h}\mathbf{b}_k(m) \text{ to } h \in \mathcal{N}(m)$$
$$(1 + i\varepsilon a_{mm})\mathbf{b}_k(m) \text{ to } m$$

- Fluid is a complex value
- Computations can be distributed
- Higher order approximations
- Asynchronous implementation

Complexity

Inverse power iteration: For each λ
delay = $D + 2Dd_{\max}$
no. of packets = $|E|n^2 + (n|E| + |E|)d_{\max}$
Complex power iteration: For all λ
delay = $D + d_{\max}$
no. of packets = $|E|d_{\max} + n|E|$.

Asynchronous

via maintaining a polynomial

$$\mathbf{x}(z) = \sum_{\ell=0}^{d_{\max}} z^{\ell}\mathbf{x}_{\ell}$$

with $\mathbf{x}_{\ell} = (\mathbf{I} + i\varepsilon\mathbf{A})^{\ell}\mathbf{b}_0$

Order-1 Complex gossiping (Monte Carlo algorithm)

- Let

  $$\mathbf{x}_{k+1} = (\mathbf{I} + i\varepsilon\mathbf{A})\mathbf{x}_k, \quad \mathbf{x}_0 = \mathbf{b}_0.$$

Order-1 Complex gossiping (Monte Carlo algorithm)

- Let
  $\mathbf{x}_{k+1} = (\mathbf{I}+i\varepsilon\mathbf{A})\mathbf{x}_k, \quad \mathbf{x}_0 = \mathbf{b}_0.$
  Then

  $$\mathbf{x}_{k+1} = \mathbf{x}_k + i\varepsilon\mathbf{D}\mathbf{P}\mathbf{x}_k,$$

  with $\mathbf{D} := \text{diag}(D_1, \ldots, D_n)$ &
  $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$ is t.p.m. of a RW.

Order-1 Complex gossiping (Monte Carlo algorithm)

- Let
  $\mathbf{x}_{k+1} = (\mathbf{I}+i\varepsilon\mathbf{A})\mathbf{x}_k, \quad \mathbf{x}_0 = \mathbf{b}_0.$
  Then

  $$\mathbf{x}_{k+1} = \mathbf{x}_k + i\varepsilon\mathbf{D}\mathbf{P}\mathbf{x}_k,$$

  with $\mathbf{D} := \text{diag}(D_1, \ldots, D_n)$ &
  $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$ is t.p.m. of a RW.

  $$\mathbf{x}_{k+1}(m) = \mathbf{x}_k(m) + i\varepsilon D_m \mathbb{E}[\mathbf{x}_k(\xi_m)],$$

  where ξ_m as a randomly
  picked neighbour of node-m

Order-1 Complex gossiping (Monte Carlo algorithm)

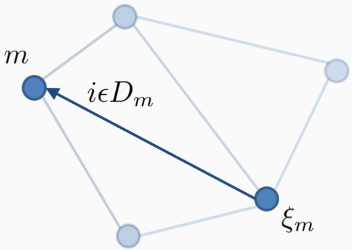
- Let
  $\mathbf{x}_{k+1} = (\mathbf{I}+i\varepsilon\mathbf{A})\mathbf{x}_k, \quad \mathbf{x}_0 = \mathbf{b}_0.$
  Then

  $$\mathbf{x}_{k+1} = \mathbf{x}_k + i\varepsilon\mathbf{D}\mathbf{P}\mathbf{x}_k,$$

  with $\mathbf{D} := \mathrm{diag}(D_1, \ldots, D_n)$ &
  $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$ is t.p.m. of a RW.

  $$\mathbf{x}_{k+1}(m) = \mathbf{x}_k(m) + i\varepsilon D_m \mathbb{E}[\mathbf{x}_k(\xi_m)],$$

  where ξ_m as a randomly
  picked neighbour of node-m



- This can be implemented via
  parallel random walks.

Implementation with Quantum Random Walk (QRW)

Preliminaries

We tried to solve a discretization of $\frac{\partial}{\partial t}\mathbf{b}_t = i\mathbf{A}\mathbf{b}_t$.

Preliminaries

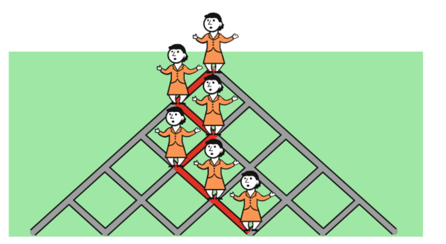
We tried to solve a discretization of $\frac{\partial}{\partial t}\mathbf{b}_t = i\mathbf{A}\mathbf{b}_t$.
Very similar to classic Schrödinger equation:

$$i\hbar\frac{\partial}{\partial t}\boldsymbol{\psi}_t = \mathbf{H}\boldsymbol{\psi}_t \qquad \text{where} \qquad \begin{aligned}\boldsymbol{\psi}_t &= \text{wave function}\\ \hbar &= \text{Planck constant}\\ \mathbf{H} &= \text{Hamilitonian operator}\end{aligned}$$

Continuous time QRW

Continuous time QRW on a graph: $\boldsymbol{\psi}_t = e^{-i\mathbf{A}t}\boldsymbol{\psi}_0$: $\boldsymbol{\psi}_t$ is a complex amplitude vector $\{\boldsymbol{\psi}_t(i), 1 \leq i \leq n\}$ with the probability of finding QRW in node i at time t is $|\boldsymbol{\psi}_t(i)|^2$.

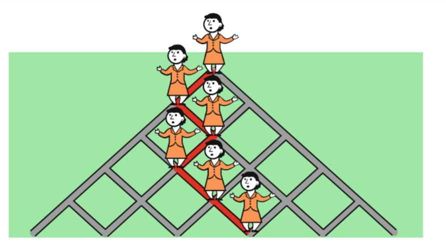
Sample path example



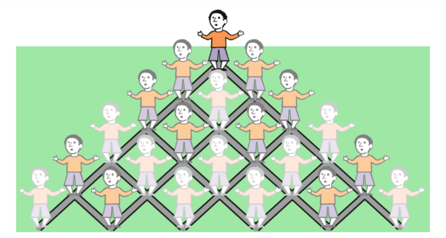
A sample path of classical RW

Figures are taken from Wang et al. Physical Implementation of Quantum Walks. Springer Berlin, 2013.

Sample path example



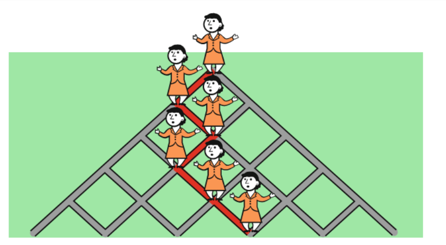
A sample path of classical RW



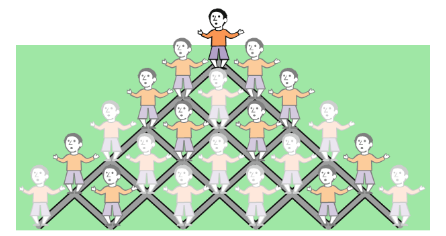
A sample quantum wave function $\boldsymbol{\psi}_t$ of QRW

Figures are taken from Wang et al. Physical Implementation of Quantum Walks. Springer Berlin, 2013.

Sample path example



A sample path of classical RW



A sample quantum wave function ψ_t of QRW

Figures are taken from Wang et al. Physical Implementation of Quantum Walks. Springer Berlin, 2013.

Technique

Technique

1. Walker represented by a qubit with $d_{\max}$ atoms: Initialized as $(1/\sqrt{d_{\max}}) \sum_{k=0}^{d_{\max}-1} |k\rangle$.

Technique

1. Walker represented by a qubit with $d_{\max}$ atoms: Initialized as $(1/\sqrt{d_{\max}}) \sum_{k=0}^{d_{\max}-1} |k\rangle$.
2. State $|k\rangle$ gets delyed by $k\varepsilon$ time units via *splitting chain*

Technique

1. Walker represented by a qubit with $d_{\max}$ atoms: Initialized as $(1/\sqrt{d_{\max}}) \sum_{k=0}^{d_{\max}-1} |k\rangle$.
2. State $|k\rangle$ gets delyed by $k\varepsilon$ time units via *splitting chain*
3. Walker moves as CT-QRW with wave function

$$\mathbf{\Psi}_t^{d_{\max}} = \frac{1}{\sqrt{d_{\max}}} \sum_{k=0}^{d_{\max}-1} e^{i(t-k\varepsilon)\mathcal{H}} \mathbf{\Psi}_0 |k\rangle.$$

Technique

1. Walker represented by a qubit with $d_{\max}$ atoms: Initialized as $(1/\sqrt{d_{\max}}) \sum_{k=0}^{d_{\max}-1} |k\rangle$.
2. State $|k\rangle$ gets delyed by $k\varepsilon$ time units via *splitting chain*
3. Walker moves as CT-QRW with wave function

$$\mathbf{\Psi}_t^{d_{\max}} = \frac{1}{\sqrt{d_{\max}}} \sum_{k=0}^{d_{\max}-1} e^{i(t-k\varepsilon)\mathcal{H}} \mathbf{\Psi}_0 |k\rangle.$$

4. At $t \geq \varepsilon d_{\max}$, on node m apply QFT on $\mathbf{\Psi}_t^{d_{\max}}(m) \Longrightarrow \sum_{k=0}^{d_{\max}-1} y_k |k\rangle$
5. When we measure, we see k with probability $|y_k|^2$, an eigenvalue point shifted by Δ.

Parameter analysis and tuning

Convergence rate and trace technique

- Order of convergence:

$$
\begin{aligned}
&\varepsilon \Re\left(\mathbf{I} + 2 \sum_{\ell=1}^{d_{\max}} e^{i\ell\varepsilon \mathbf{A}} \mathbf{b}_0 e^{-i\ell\varepsilon\theta} e^{-\ell^2\varepsilon^2 v/2} \right) \\
&\quad = \int_{-\varepsilon d_{\max}}^{+\varepsilon d_{\max}} e^{i\mathbf{A}t} \mathbf{b}_0 e^{-t^2 v/2} e^{-it\theta} dt + O\left(\lambda_1 \varepsilon^2 d_{\max} \|\mathbf{b}_0\|\right)
\end{aligned}
$$

Convergence rate and trace technique

- Order of convergence:

$$\varepsilon \Re \left(\mathbf{I} + 2 \sum_{\ell=1}^{d_{\max}} e^{i\ell\varepsilon\mathbf{A}} \mathbf{b}_0 e^{-i\ell\varepsilon\theta} e^{-\ell^2\varepsilon^2 v/2} \right)$$
$$= \int_{-\varepsilon d_{\max}}^{+\varepsilon d_{\max}} e^{i\mathbf{A}t} \mathbf{b}_0 e^{-t^2 v/2} e^{-it\theta} dt + O\left(\lambda_1 \varepsilon^2 d_{\max} \|\mathbf{b}_0\|\right)$$

- Getting equal peaks for all eigenvalues

  Take $\mathbf{b}_0$ as a vector of i.i.d. Gaussian$(0, w)$:

$$\mathbb{E}[\mathbf{b}_0^\intercal \mathbf{f}(\theta)] = w \sum_{j=1}^{n} \sqrt{\frac{2\pi}{v}} \exp\left(-\frac{(\lambda_j - \theta)^2}{2v}\right)$$

Convergence rate and trace technique

- Order of convergence:

$$\varepsilon\Re\left(\mathbf{I}+2\sum_{\ell=1}^{d_{\max}} e^{i\ell\varepsilon\mathbf{A}}\mathbf{b}_0 e^{-i\ell\varepsilon\theta}e^{-\ell^2\varepsilon^2 v/2}\right)$$
$$= \int_{-\varepsilon d_{\max}}^{+\varepsilon d_{\max}} e^{i\mathbf{A}t}\mathbf{b}_0 e^{-t^2 v/2}e^{-it\theta}dt + O\left(\lambda_1\varepsilon^2 d_{\max}\|\mathbf{b}_0\|\right)$$

- Getting equal peaks for all eigenvalues

  Take $\mathbf{b}_0$ as a vector of i.i.d. Gaussian$(0, w)$:

$$\mathbb{E}[\mathbf{b}_0^\intercal\mathbf{f}(\theta)] = w\sum_{j=1}^{n}\sqrt{\frac{2\pi}{v}}\exp(-\frac{(\lambda_j-\theta)^2}{2v})$$

  - Detecting algebraic multiplicity

Choice of parameters

Let Δ be the maximum degree.

Choice of parameters

Let Δ be the maximum degree.

1. Parameter v: With 99.7% of the Gaussian areas not overlapping,

$$6v < \min_{1 \leq i \leq k-1} |\lambda_i - \lambda_{i+1}| < 2\lambda_1 < 2\Delta$$

Choice of parameters

Let Δ be the maximum degree.

1. Parameter v: With 99.7% of the Gaussian areas not overlapping,

$$6v < \min_{1 \le i \le k-1} |\lambda_i - \lambda_{i+1}| < 2\lambda_1 < 2\Delta$$

2. Parameter ε: From sampling theorem, to avoid aliasing,

$$\varepsilon < \frac{1}{2(|\lambda_1 - \lambda_n| + 6v)}$$

Choosing $\varepsilon < \frac{1}{4\Delta + 12v}$ will ensure this.

Choice of parameters

Let Δ be the maximum degree.

1. Parameter v: With 99.7% of the Gaussian areas not overlapping,

$$6v < \min_{1 \leq i \leq k-1} |\lambda_i - \lambda_{i+1}| < 2\lambda_1 < 2\Delta$$

2. Parameter ε: From sampling theorem, to avoid aliasing,

$$\varepsilon < \frac{1}{2(|\lambda_1 - \lambda_n| + 6v)}$$

   Choosing $\varepsilon < \frac{1}{4\Delta + 12v}$ will ensure this.

3. Parameter $d_{\max}$: $1/d_{\max} < \varepsilon < 1/\sqrt{d_{\max}}$

Choice of parameters

Let Δ be the maximum degree.

1. Parameter v: With 99.7% of the Gaussian areas not overlapping,

$$6v < \min_{1 \leq i \leq k-1} |\lambda_i - \lambda_{i+1}| < 2\lambda_1 < 2\Delta$$

2. Parameter ε: From sampling theorem, to avoid aliasing,

$$\varepsilon < \frac{1}{2(|\lambda_1 - \lambda_n| + 6v)}$$

   Choosing $\varepsilon < \frac{1}{4\Delta + 12v}$ will ensure this.

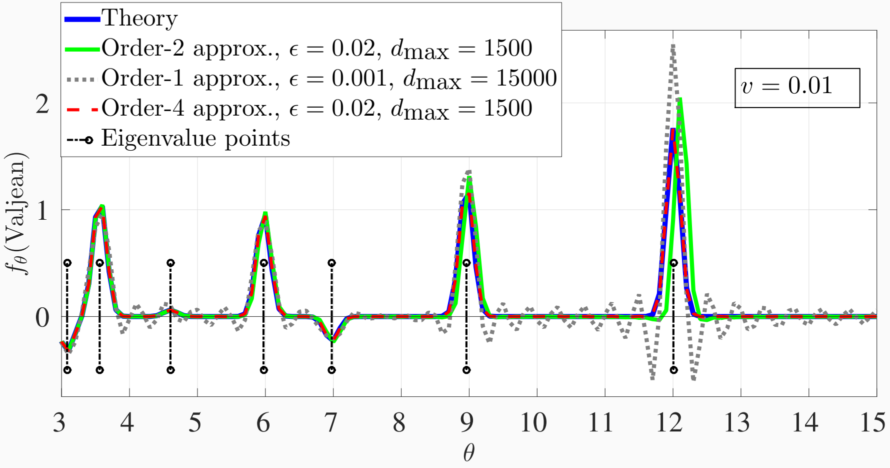
3. Parameter $d_{\max}$: $1/d_{\max} < \varepsilon < 1/\sqrt{d_{\max}}$

Scalability

No. of iterations $d_{\max}$ depends on maximum degree Δ.

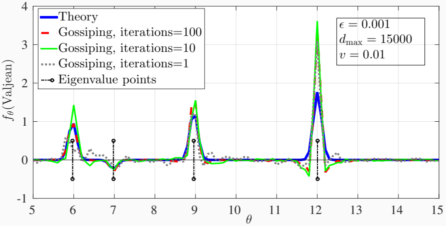
Numerical studies on real-world networks

Number of nodes: 77, number of edges: 254.



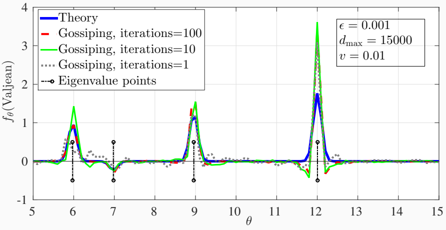
Complex diffusion

Les Misérables network (contd.)

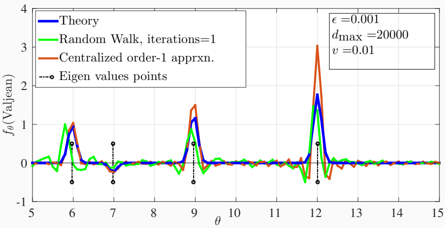


Monte Carlo gossiping

Les Misérables network (contd.)



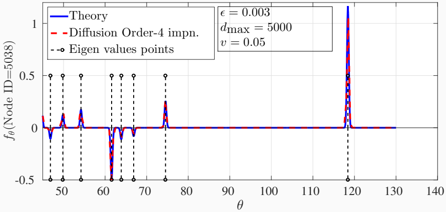
Monte Carlo gossiping



Parallel random walk

Enron email network

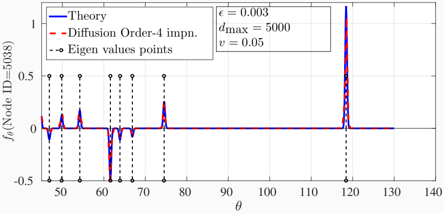
Number of nodes: 33K, number of edges: 180K.



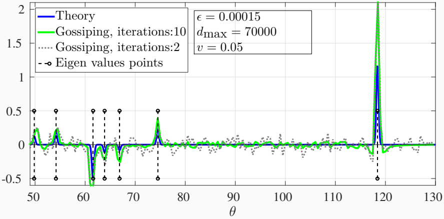
Complex diffusion order-4

Enron email network

Number of nodes: 33K, number of edges: 180K.



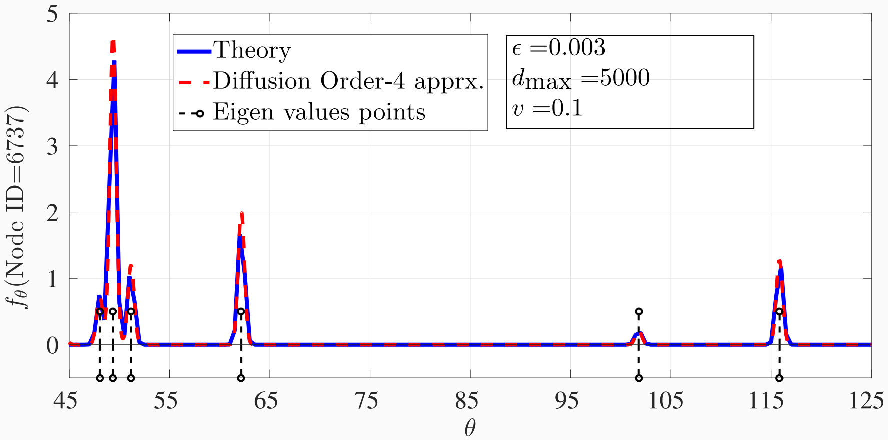
Complex diffusion order-4



Monte Carlo gossiping

DBLP network

Number of nodes: 317K, number of edges: 1M.



Complex diffusion order-4

Conclusions

- A simple interpretation of spectrum in terms of peaks of eigenvalue points.

Conclusions

- A simple interpretation of spectrum in terms of peaks of eigenvalue points.
- Developed distributed algorithms at node level based on complex power iterations

Conclusions

- A simple interpretation of spectrum in terms of peaks of eigenvalue points.
- Developed distributed algorithms at node level based on complex power iterations
  - Complex diffusion: each node collect fluid from all the neighbors
  - Complex gossiping: each node collect fluid from one random neighbor
  - Parallel random walk implementation

Conclusions

- A simple interpretation of spectrum in terms of peaks of eigenvalue points.
- Developed distributed algorithms at node level based on complex power iterations
  - Complex diffusion: each node collect fluid from all the neighbors
  - Complex gossiping: each node collect fluid from one random neighbor
  - Parallel random walk implementation
- Connection with quantum random walk techniques

Conclusions

- A simple interpretation of spectrum in terms of peaks of eigenvalue points.
- Developed distributed algorithms at node level based on complex power iterations
  - Complex diffusion: each node collect fluid from all the neighbors
  - Complex gossiping: each node collect fluid from one random neighbor
  - Parallel random walk implementation
- Connection with quantum random walk techniques
- Derived order of convergence and algorithms are scalable with the maximum degree of the graph.

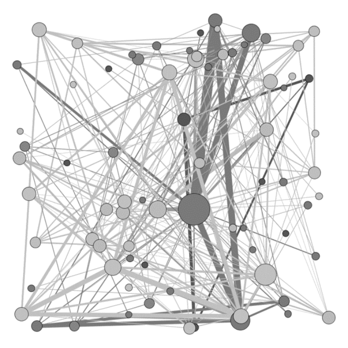
Conclusions

- A simple interpretation of spectrum in terms of peaks of eigenvalue points.
- Developed distributed algorithms at node level based on complex power iterations
  - Complex diffusion: each node collect fluid from all the neighbors
  - Complex gossiping: each node collect fluid from one random neighbor
  - Parallel random walk implementation
- Connection with quantum random walk techniques
- Derived order of convergence and algorithms are scalable with the maximum degree of the graph.
- Numerical simulations on various real-world networks.

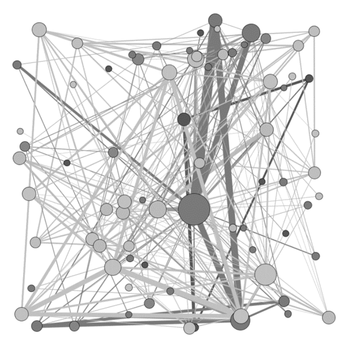
Thank you!
Questions?

More information available at http://bit.do/Jithin

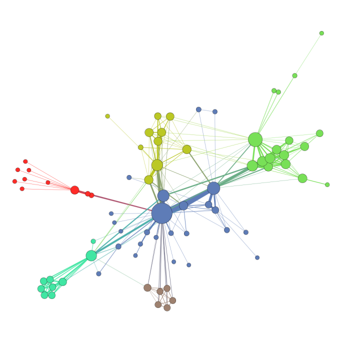
Motivation from graph clustering



Motivation from graph clustering



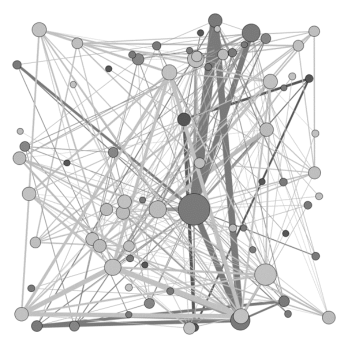
$\Longrightarrow$



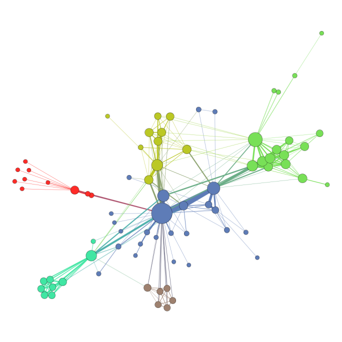
Les Misérables network

- A classical problem in graph theory

Motivation from graph clustering



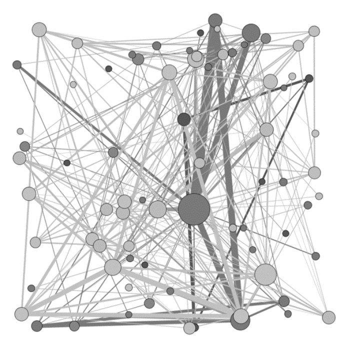
$\Longrightarrow$



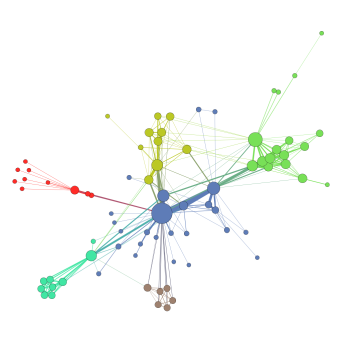
Les Misérables network

- A classical problem in graph theory
- More difficult when graph is not known a priori

Motivation from graph clustering



$\Longrightarrow$



Les Misérables network

- A classical problem in graph theory
- More difficult when graph is not known a priori
- An efficient solution is **Spectral clustering:**
  Requires knowledge of eigenvalues and eigenvectors of graph matrices.