Quantum Random Walk in Complex Networks

K. Avrachenkov, P. Jacquet, J. K. Sreedharan
k.avrachenkov@inria.fr, philippe.jacquet@alcatel-lucent.com, jithin.sreedharan@inria.fr

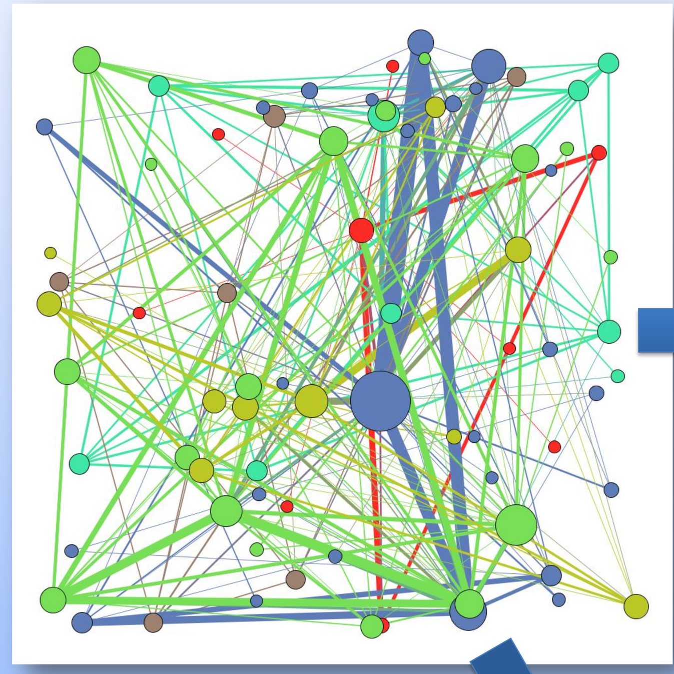
Bell Labs

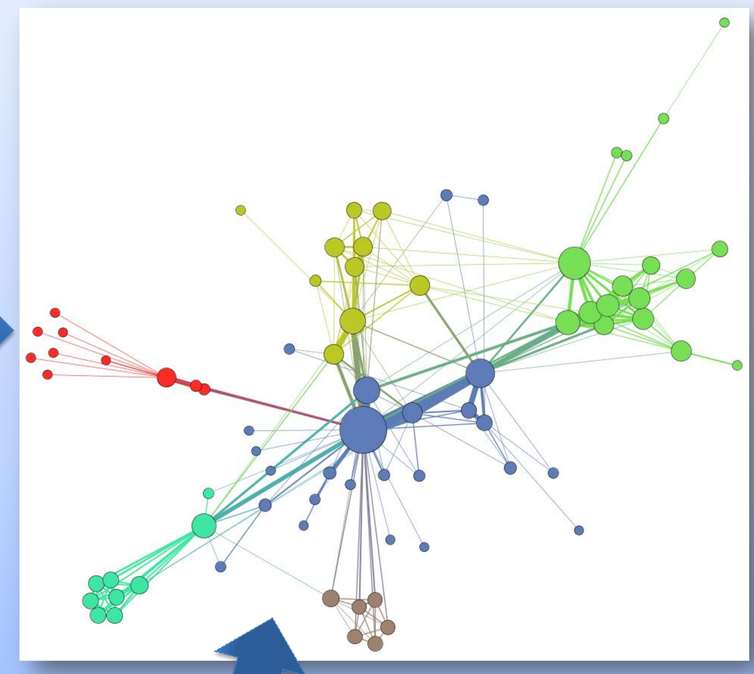
Inria informatics mathematics

Motivation from Community Detection

- A classical problem in Graph theory
- It is more difficult when the graph is not known a priori
- An effective solution is Spectral Clustering:

  For partitioning into k communities, it requires k upper extreme eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$ and corresponding eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ of Adjacency matrix $\mathbf{A}$ or lower extreme eigenvalues of Laplacian matrix.
- Our aim is to find exact values or approximations of $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$ and $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ in a quicker way.





Our Central Idea

A variation of Spectral theorem says that

$$\int_{-\infty}^{+\infty} e^{i\mathbf{A}t} e^{-it\theta} dt = \sum_j \delta_{\lambda_j}(\theta) \mathbf{u}_j \mathbf{u}_j^t$$

But to smooth the harmonic oscillations that will otherwise hide the Dirac peaks, and for ease of calculations we modify it to

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

where $\mathbf{b}_0$ is any initial vector.

At any node i

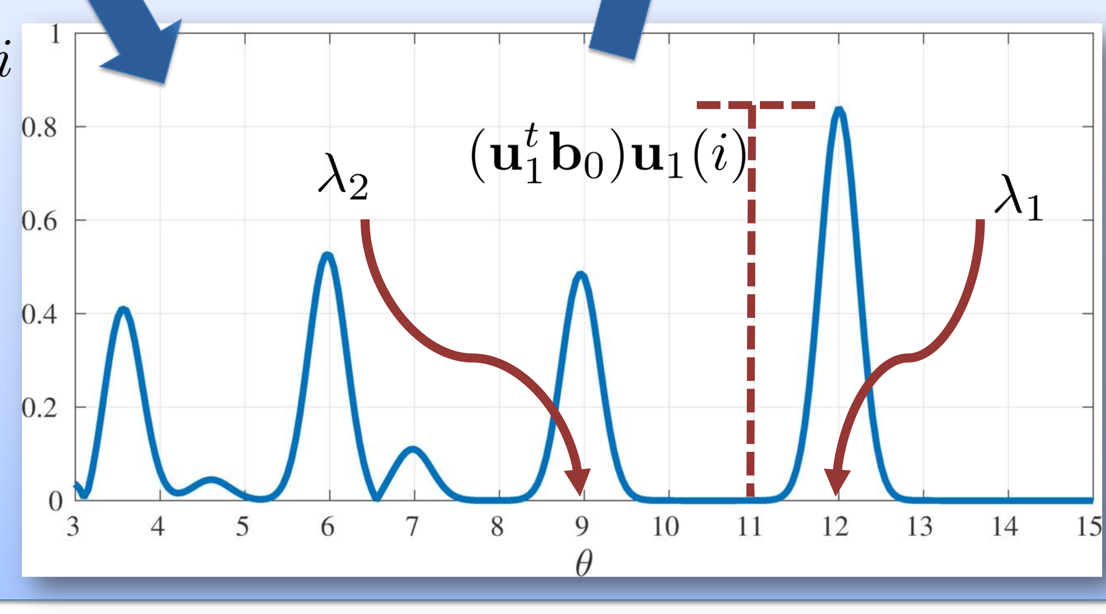


Fig 1 is taken from Wang et al. Physical Implementation of Quantum Walks. Springer Berlin, 2013.

Computation with Quantum Random Walk

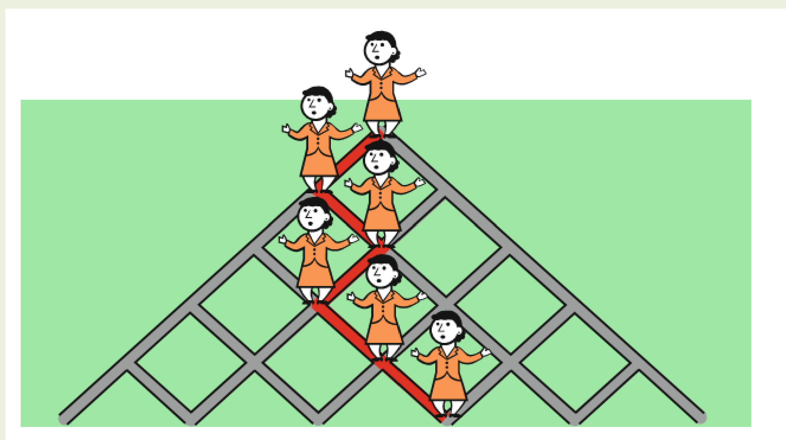
Continuous time Quantum Random Walk (QRW) is defined as $\boldsymbol{\psi}_t = e^{-i\mathbf{A}t}\boldsymbol{\psi}_0$, where $\boldsymbol{\psi}_t$ is a complex amplitude vector $\{\boldsymbol{\psi}_t(i), 1 \leq i \leq n\}$ with the probability of finding the QRW in node i at time t is $|\boldsymbol{\psi}_t(i)|^2$

The central equation (1) can be approximated in discrete time as

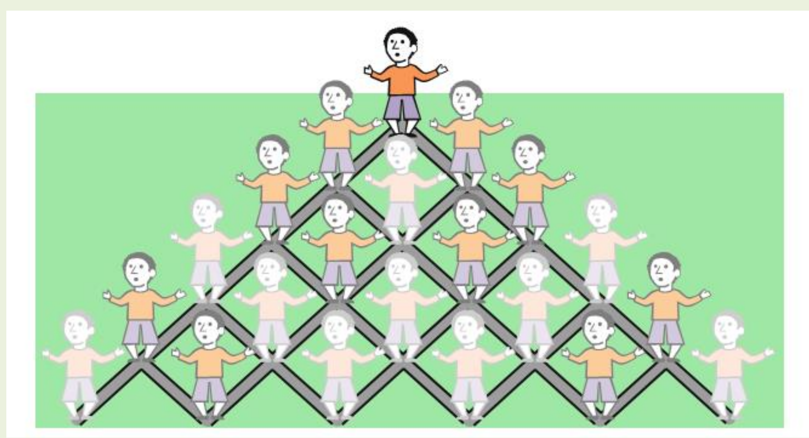
$$\int_{-\infty}^{+\infty} e^{i\mathbf{A}t} \mathbf{b}_0 e^{-t^2 v/2} e^{-it\theta} dt \approx \varepsilon \Re \left(\mathbf{b}_0 + 2 \sum_{\ell=1}^{d_{\max}} e^{i\ell\varepsilon\mathbf{A}} \mathbf{b}_0 e^{-i\ell\varepsilon\theta} e^{-\ell^2\varepsilon^2 v/2} \right) \quad (2)$$

Using QRW for calculating extreme eigenvalues and eigenvectors:
(use of $-\mathbf{A}$ instead of $\mathbf{A}$ will only just reverse the location of eigenvalues)

1. Initialize QRW, $\boldsymbol{\psi}_0$ with a random vector $\mathbf{b}_0$ in $[0,1]^n$
2. Sample the QRW at $(\ell\varepsilon)$ instants $1 \leq \ell \leq d_{\max}$
3. Calculate the approximation (2) at each node
4. Find peaks in (2) and thus obtain eigenvalues and eigenvector components
5. Do not measure the quantum system until the calculations are over



A sample path of classical RW



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

Fig. 1: Comparison of classical RW with QRW

In the mean time, while the quantum computers are not readily available, we propose three approaches using classical numerical techniques.

Computation with Classical Computer

We compute approximation (2) with the following centralized and distributed techniques. The main task is to compute $e^{i\ell\varepsilon\mathbf{A}}\mathbf{b}_0$.

Graph example: character network in Les Misérables novel. Nodes are the characters and edges are formed if two characters appear in the same chapter. Figures show the plot at the node Valjean.

Centralized Approach

Adjacency matrix is known. Approximating $e^{i\ell\varepsilon\mathbf{A}}$ by

Order-1: $(\mathbf{I} + i\varepsilon\mathbf{A})^\ell$

Order-2: $(\mathbf{I} + i\varepsilon\mathbf{A} + \frac{1}{2}(i\varepsilon\mathbf{A})^2)^\ell$

Order-4: $(\mathbf{I} + i\varepsilon\mathbf{A} + \frac{1}{2}(i\varepsilon\mathbf{A})^2 + \frac{1}{6}(i\varepsilon\mathbf{A})^3 + +\frac{1}{24}(i\varepsilon\mathbf{A})^4)^\ell$

Distributed Diffusion Approach

Calculate $e^{i\ell\varepsilon\mathbf{A}}\mathbf{b}_0$ via Order-1, 2 and 4 approximations in distributed and asynchronous way.

Distributed: Each node knows only its neighbors

Asynchronous: Each node does not have to keep track the diffusion timings of its neighbors

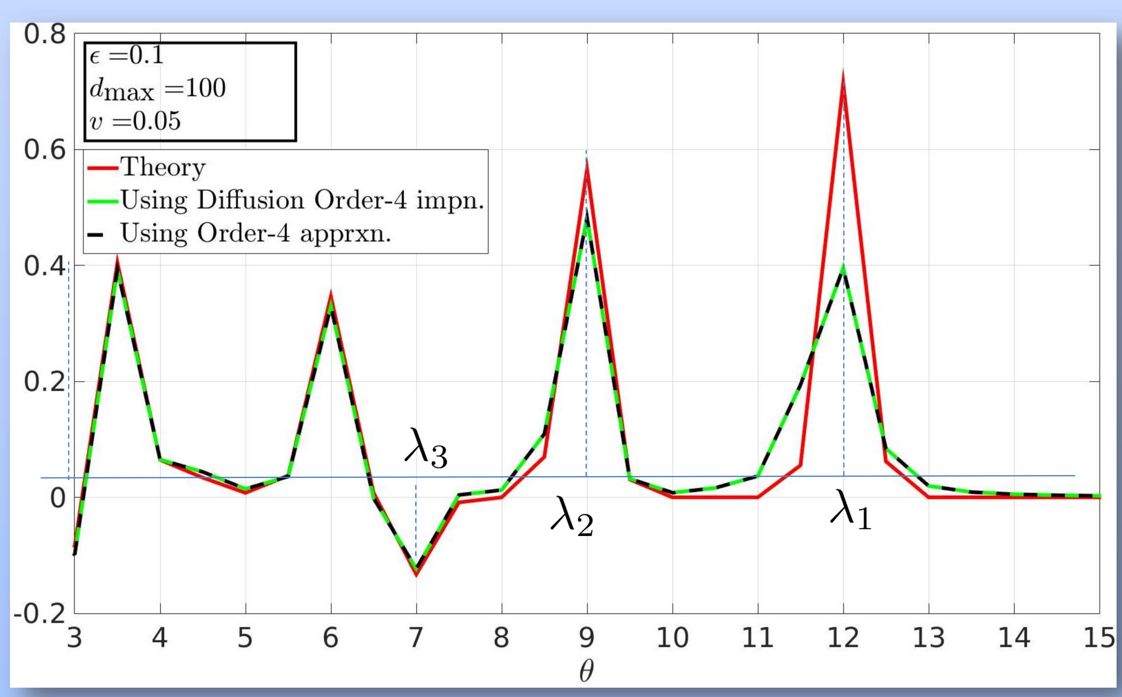
Fluid diffusion approach: The idea is to compute the coefficients of the polynomial in z, $\sum_{\ell=1}^{d_{\max}} (\mathbf{I} + i\varepsilon\mathbf{A})^\ell z^\ell$ using fluid diffusion with initial fluid at node i as $\mathbf{b}_0(i)$. For order-2 and 4, polynomial will change accordingly.

	Memory (vector size)	Computing time
Order-1	$2d_{\max}$	$\tau d_{\max}$
Order-2	$4d_{\max}$	$2\tau d_{\max}$
Order-4	$8d_{\max}$	$4\tau d_{\max}$

$\tau =$ average diffusion-fusion cycle time

Sample values for satisfactory performance in Les Misérables graph.

	$d_{\max}$	ε	v
Order-1	5000	0.001	0.05
Order-2	500	0.01	0.05
Order-4	100	0.1	0.05



Distributed Monte Carlo Approach for Order-1

$$\begin{aligned}
\mathbf{R}_k &\triangleq (\mathbf{I} + i\varepsilon\mathbf{A})^k \mathbf{b}_0, \; 0 \leq k \leq d_{\max} \\
\mathbf{R}_{k+1} &= (\mathbf{I} + i\varepsilon\mathbf{A})\mathbf{R}_k, \mathbf{R}_0 = \mathbf{b}_0 \\
&= \mathbf{R}_k + i\varepsilon(\mathbf{D}\mathbf{D}^{-1})\mathbf{A}\mathbf{R}_k \\
&= \mathbf{R}_k + i\varepsilon\mathbf{D}\mathbf{P}\mathbf{R}_k,
\end{aligned}$$

$\mathbf{D}$: the diagonal matrix with entries as the degrees of the nodes $(D_1, \ldots, D_n)$

$\mathbf{P}$: transition probability matrix of classical Random Walk on graph

The ith component of $\mathbf{R}_{k+1}$, $\mathbf{R}_{k+1}(i) = \mathbf{R}_k(i) + i\varepsilon D_i \mathbb{E}(\mathbf{R}_k(\xi_i))$,

ξ_i : a randomly picked neighbor of node i.

Implementation via
a) Monte Carlo gossip
b) Parallel Random Walk