Walks on Networks

Prof. Richard C. Wilson

Dept. of Computer Science
University of York

THE UNIVERSITY of York

DEPARTMENT OF COMPUTER SCIENCE

Random Walks

- Random walks have found multiple uses in the analysis of graphs and networks
- Closely related to
  - Spectral methods
  - Heat kernel and diffusion processes
- Used for
  - Matching
  - Embedding
  - Characterisation
  - Graph-based processing

Random walks algorithms

Algorithms

- Image smoothing: heat-kernel can be used as an anisotropic filter for structure preserving image smoothing.
- Graph simplification: can be used to reduce graphs to simpler structures such as trees and strings that are more easily manipulated.
- Embedding: Can embed graphs in space so as to preserve diffusion distance or commute time between nodes.
- Permutation invariants: can be derived from structure of random walks.
- Consistent labelling: Relaxation labelling can be realised as a process of running a continuous time random walk on graph where nodes are object-label assignments and edges represent label compatibility.

Graph Analysis

- Gori, Maggini & Sarti: Graph Matching using Random Walks.
- Robles-Kelly & Hancock: String Edit Distance, Random Walks & Graph Matching.
- Meila & Shi: A Random Walks View of Spectral Segmentation.

Random walk embeddings

Embeddings

- Borgwardt: Random walk kernels on graphs.
- Lafon et al: Diffusion map. Commute time is average of diffusion time over all paths connecting a pair of nodes.
- Qiu and Hancock: Commute time for image segmentation and multi-body tracking.

Random Walk Evolution

- Weighted adjacency matrix $$A_{uv} = \begin{cases} w(u,v) & (u,v) \in E \\ 0 & otherwise \end{cases}$$
- Degree matrix $$D_{uu} = \sum_{v \in V} A_{uv}$$
- Transition matrix $$\mathbf{T} = \mathbf{D}^{-1}\mathbf{A}$$
- Time evolution of vertex prob. $$\mathbf{p}_t = \mathbf{T}\mathbf{p}_{t-1} = \mathbf{T}^t \mathbf{p}_0$$
- Steady state $$\mathbf{p}_s = \mathbf{T}\mathbf{p}_s$$

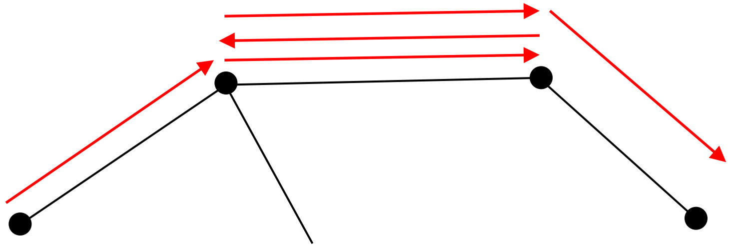
Steady state is determined by the leading eigenvector of **T** (λ=1)

Graph Kernels

- Random Walk Kernel (Gartner et al 2003)
  - Count the number of matching walks between two graphs

$$K(G_1,G_2)=\sum_{(i,j)\in V_\times}\sum_{k=0}^{\infty}\varepsilon_k\left[A_\times^k\right]_{ij}$$

  - $A_\times$ is the product graph of G_1 and G_2
  - k is the walk length
  - The number of walks becomes very large
  - The random walk graph kernel suffers from the problem of tottering



  - Reduces expressive power and masks structural differences

Backtrackless Random Walk

- A random walk of length *k* is a sequence of vertices

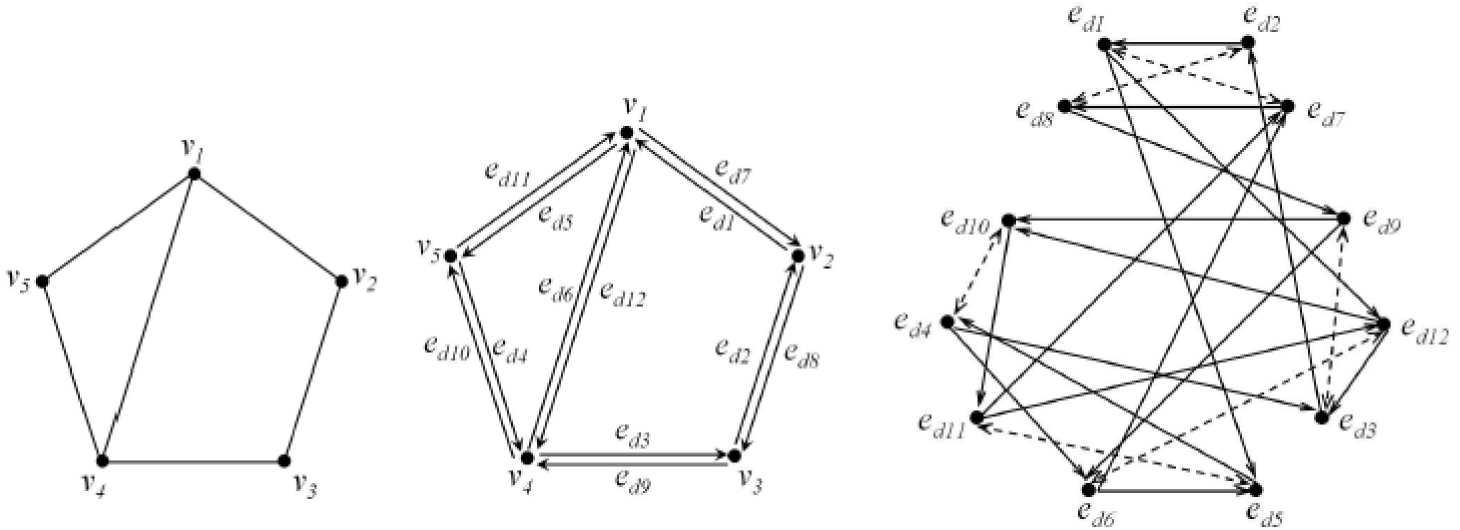
$$u_1, u_2, \ldots, u_{k+1}$$

  – Such that $e_i = (u_i, u_{i+1}) \in E$

- A backtrackless random walk has the additional condition

$$e_i \neq e_{i+1}$$

  – A sequence of oriented edges, excluding backtracking step



(a) Original Graph (b) Digraph (c) Oriented Line Graph

Backtrackless Random Walk Kernel

- The backtrackless random walk kernel is

$$K(G_1, G_2) = \sum_{(i,j)\in V_\times} \sum_{k=0}^{\infty} \varepsilon_k \left[A_\times^k\right]_{ij}$$

- Defined on the product graph of the OLGs

$$V_\times\left(G_1 \times G_2\right) = \{(v_1, v_2) \in V_1 \times V_2\}$$
$$E_\times\left(G_1 \times G_2\right) = \{((u_1, u_2), (v_1, v_2))\}$$
$$(u_1, u_2) \in E_1 \wedge (v_1, v_2) \in E_2$$

- By eliminating the reverse edges in the OLG, we eliminate backtracking
- Complexity is a problem
  - Efficient method to compute given in (Aziz, Wilson Hancock SIMBAD'11)

Commute Time

- Commute-time embedding [Qui, Hancock, 2007]
- Hitting time $Q(u,v)$: Expected time to arrive at v, starting from u
- Commute time $\mathrm{CT}(u,v)$: Round trip time

$$\mathrm{CT}(u,v) = Q(u,v) + Q(v,u)$$

- Green's function for graph (spectral representation)
  - Type of pseudo-inverse of Laplacian

$$G(u,v) = \sum_{i=2}^{|V|} \lambda_i^{-1} \phi_i(u) \phi_i(v)$$

- Relationship

$$Q(u,v) = \frac{vol}{d_v} G(v,v) - \frac{vol}{d_u} G(u,u)$$

Commute Time

- Commute time

$$CT(u,v) = Q(u,v) + Q(v,u)$$

$$= \frac{vol}{d_u} G(u,u) + \frac{vol}{d_v} G(v,v) - \frac{vol}{d_u} G(u,v) - \frac{vol}{d_u} G(v,u)$$

$$= vol \sum_{i=2}^{|V|} \frac{1}{\lambda_i} \left(\phi_i(u) - \phi_i(v) \right)^2$$

- Commute-time embedding

$$\mathbf{Y} = \sqrt{\frac{vol}{\Lambda}} \Phi^T$$

Preserves CT as distances between vertices

Quantum commute time [Emms, Hancock, Wilson 2008]

Quantum Walk

- A *Quantum Walker* obeys the laws of quantum mechanics
- Described by a complex wave function ψ
  - Amplitude may be negative, state probability is ψ ψ*
  - Evolution must be reversible
  - Observation collapses wave function

- Richer structure due to interference
- [Emms et al; QIC 2009,PR 2009,IVC 2009]

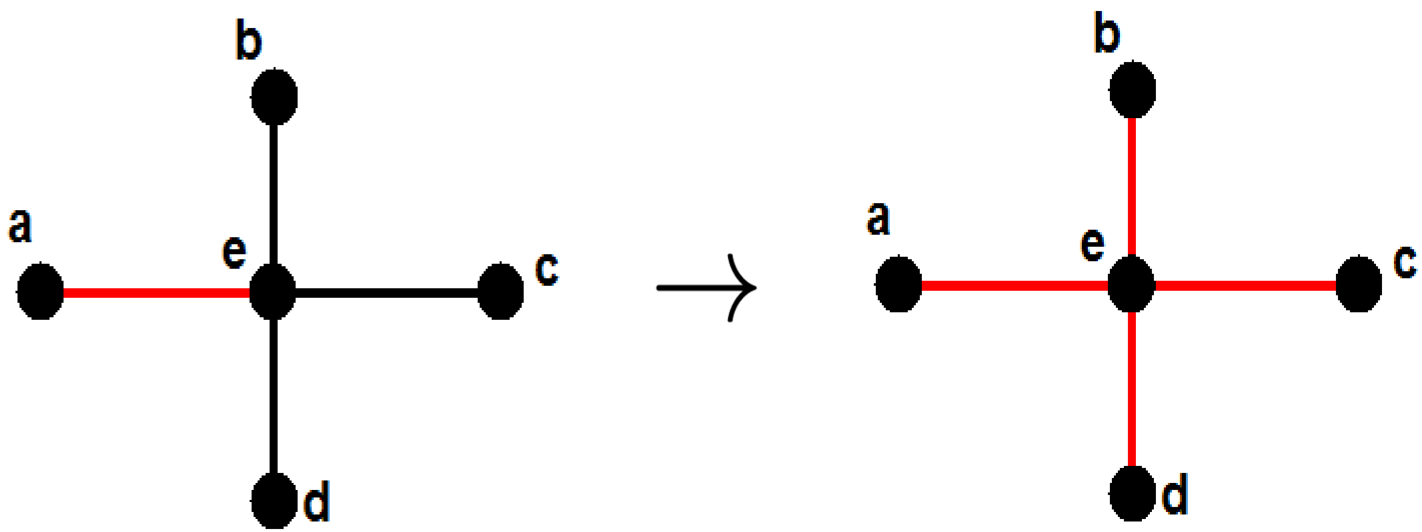
Quantum Walk Evolution

- Evolution matrix $\mathbf{U} = \mathbf{VC}$
  - Unitary (rather than stochastic) matrix $\mathbf{U}^{\dagger}\mathbf{U}=\mathbf{I}$
- Coin matrix (Grover coin) $\mathbf{C}$
- Transition matrix $\mathbf{V}$
- Time evolution of wavefunction. $\boldsymbol{\psi}_t = \mathbf{U}\boldsymbol{\psi}_{t-1} = \mathbf{U}^t \boldsymbol{\psi}_0$
- No steady state
- As walk is reversible, walks must be labelled by current and previous vertex (u,v)
  - Walk is on *edges* of graph

- Evolution matrix

$$U_{(a,b),(b,c)} = \frac{2}{d_b} - \delta_{ac} \qquad (a,b),(b,c) \in E$$

- Example



Initial state

$$\psi_0 = |(a,e)\rangle$$

Next state

$$\psi_1 = -\frac{1}{2}|(e,a)\rangle + \frac{1}{2}\left(|(e,b)\rangle + |(e,c)\rangle + |(e,d)\rangle\right)$$

Structure of Quantum Walk

- Spectrum of **U** is $\left\{\pm 1^{2|E|-2|V|}, \pm\sqrt{1-\lambda_i}\right\}$ [Emms et al 2009]
  - λ_i are the eigenvalues of the random walk
  - No difference to random walk
  - Spectrally, powers of **U** not interesting
- The positive support of **U** is

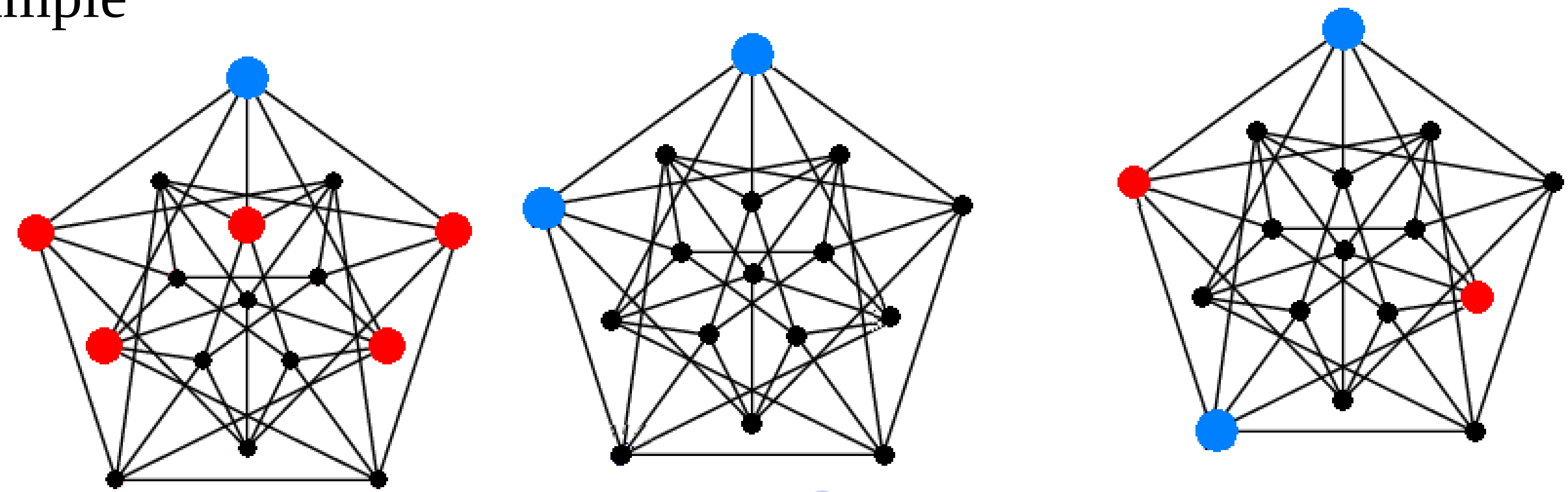
$$\mathrm{Sp}^+(\mathbf{U})_{ij} = \begin{cases} 1 & \text{if } U_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

- A graph connecting vertices which have positive amplitude for the quantum walk
- Encodes interference effects
- $\mathrm{Sp}^+(\mathbf{U})$ is the oriented line graph of G
  - Backtrackless random walk

Strongly Regular Graphs (SRGs)

- A strongly regular graph with parameters (n,k,l,m) is a k-regular graph on n vertices for which each pair of adjacent vertices share l common neighbours and each pair of non-adjacent vertices share m common neighbours.

Example



- k=5
- m=0
- l=2

- There is no proven poly-time algorithm for SRG isomorphism.

Strongly Regular Graphs

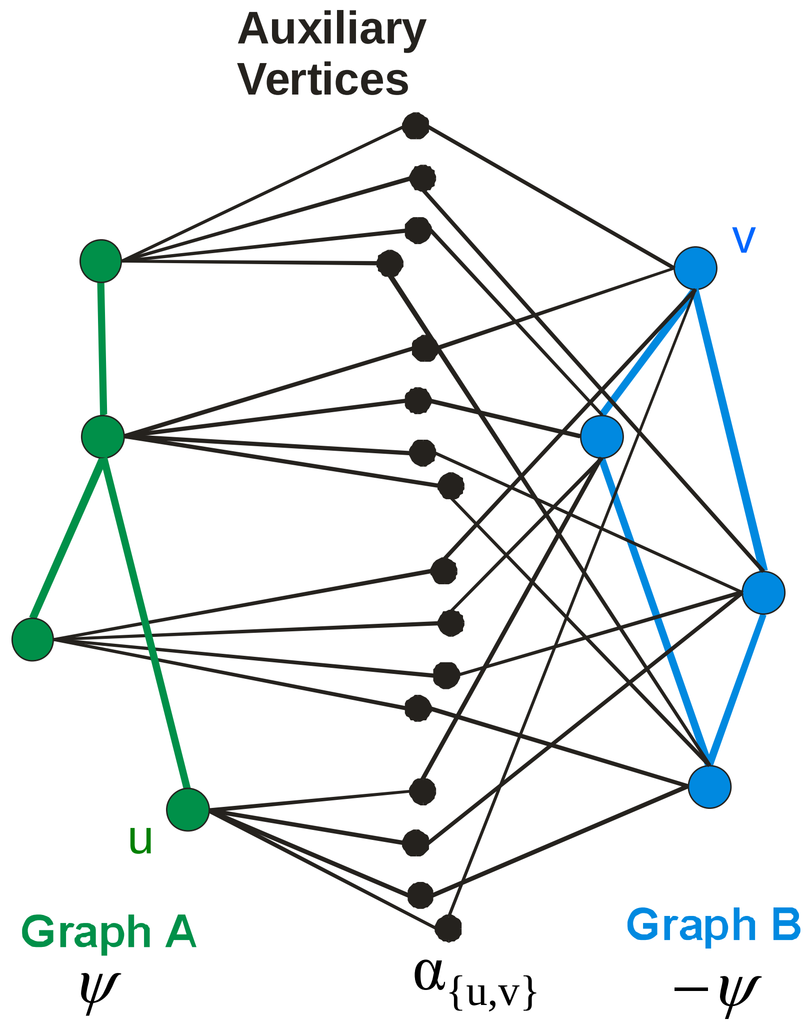
- The spectrum of $Sp^+(\mathbf{U})$ does not distinguish SRGs of the same family [Emms et al 2009]
- Nor does $Sp^+(\mathbf{U}^2)$ (the two-step paths with positive amplitude)
- But $Sp^+(\mathbf{U}^3)$ gives different spectra for all tested pairs of SRGs
  - Eg SRG(36,15,6,6) has 32548 members all of which are spectrally unique
- Why $\mathbf{U}^3$
  - First order in which positive and negative amplitude walks can interfere (triangles in graph)

Cospectral trees

$\lvert V \rvert$	Number of Trees	Number with a Cospectral partner using L	Number with a Cospectral partner using $S^+(U^3)$
≤ 10	198	0	0
11	235	6	0
12	551	6	0
13	1301	18	0
14	3159	30	0
15	7741	48	0
16	19320	68	0
17	48629	221	0
18	123867	230	0
19	317955	440	2
20	823065	648	2
21	2144505	1056	24
22	5623756	1563	32
23	14828074	2858	68
24	39299897	3623	290

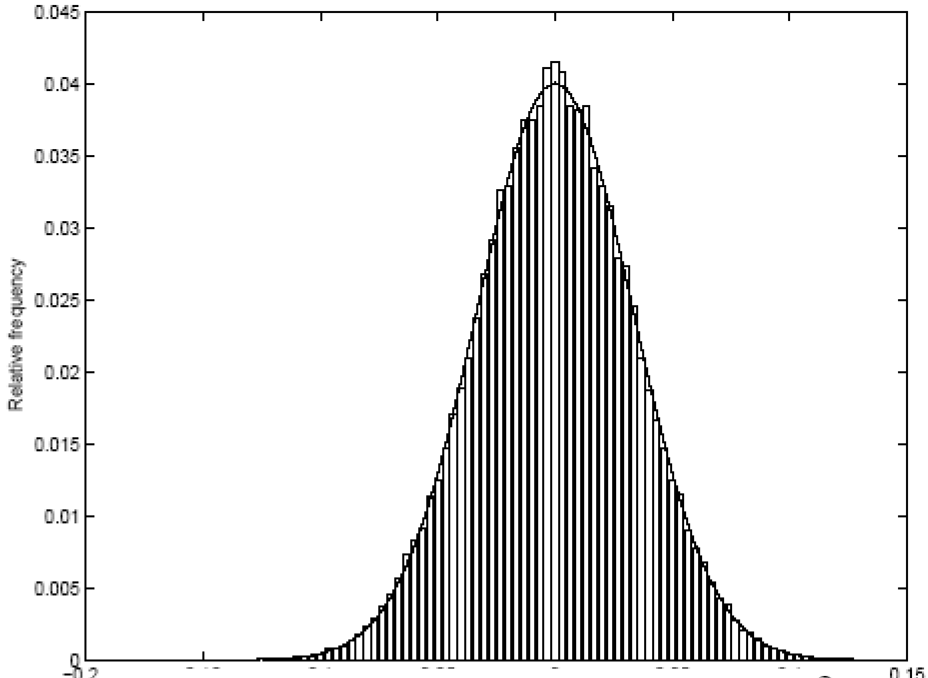
Quantum Walk Graph Matching

- Take two graphs to be matched.
  [Emms et al 2009]
- Auxiliary vertices join vertices from each graph
- Begin QRW on **A** in configuration ψ and on **B** in configuration -ψ
- If graphs are isomorphic, walks contribute equal and opposite amplitude for matching vertices at the auxiliary vertices
  - Zero amplitude indicates match
- Similar graphs should have similar walks and large destructive interference



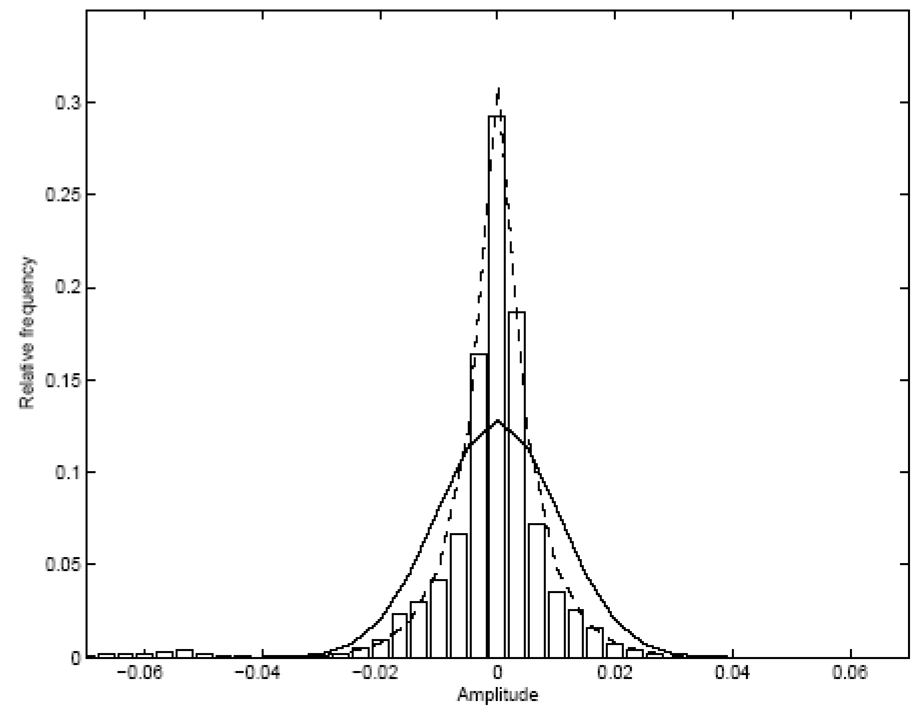
Matching found for non-isomorphic graphs using distribution of amplitudes

False matches modelled by a Gaussian distribution, σ_f:



$$p(x|\text{false}) = \frac{1}{\sqrt{2\pi\sigma_f}} e^{-\frac{x^2}{2\sigma_f^2}},$$

True matches modelled by a double exponential distribution, $\sigma_t << \sigma_f$:



$$p(x|\text{true}) = \frac{1}{\sigma_t\sqrt{2}} e^{-\frac{\sqrt{2}|x|}{\sigma_t}}.$$

Graph Calculus

- Discrete Laplacian **L** used in many applications on graphs
- Diffusion processes
  - Continuous time random walk

$$\frac{d\mathbf{H}(t)}{dt} = -\mathbf{L}\mathbf{H}(t)$$

- Wave solutions of Schrodingers equation
  - Wave kernels signatures

$$i\frac{\partial \boldsymbol{\Psi}(t)}{\partial t} = \mathbf{L}\boldsymbol{\Psi}$$

- Discrete Laplacian has connectivity but no length

- Geometric realization of graph $\mathcal{G}$ [Friedman & Tillich 2004]
  - Interval (length) associated with each edge
  - Metric graph with certain boundary conditions
- Leads to the idea of a two-part Laplacian

$$\Delta = \Delta_V d\mathcal{V} + \Delta_E d\mathcal{E}$$

- Vertex-based part Δ_V coincides with concept of discrete Laplacian
- Edge-based Laplacian Δ_E has interesting properties
  - Solutions exist on edges

Edge-based Differential Equations

1. Heat equation
   - Edge-based heat kernel

$$\frac{\partial f(t)}{\partial t} = -\Delta_E f(t)$$

2. Wave equation

$$\frac{\partial^2 f(t)}{\partial t^2} = -\Delta_E f(t)$$

3. Relativistic heat equation

$$\frac{\partial^2 f(t)}{\partial t^2} + \alpha \frac{\partial f(t)}{\partial t} = -\Delta_E f(t)$$

In contrast to discrete Laplacian, last two exhibit finite propagation speed

- Models transmission times in networks

Eigensystem

- The eigensystem of Δ_E comes in two parts
- Eigenfunctions supported on the vertices
  - Eigenvalues and eigenfunctions determined by random walk matrix T
- Eigenfunctions which are zero on the vertices
  - Eigenvalues and eigenfunctions determined by backtrackless random walk via adjacency matrix of OLG
- Contains structure from both classical RW and backtrackless RW

Conclusions

- Random walks are a powerful tool for analysing network structure
- We have explored the use of a number of different types of walk
  - Random walk (heat kernel)
  - Backtrackless walks (graph kernels)
  - Quantum walks (spectra and matching)
- Edge-based Laplacian is an interesting future direction
  - Contains structure from RW and BRW
  - Finite speed of signal propagation for networks where transmission time matters