Characterisation of Networks
and their applications

Richard C. Wilson
Dept. of Computer Science
University of York
Graphs and Networks

- A map of a city
- A fish illustration
- Molecular structures

THE UNIVERSITY OF YORK

COMPUTER SCIENCE
Graphs and networks

PIN

Social Network
Some of the questions we are trying to answer:

What is the structure of a network?
Are there parts?
How are they connected?
Do the parts look the same?

Are two networks the same?
How similar are they?
Can we tell two types of network apart?
How can we model a set of networks?
Network characterisations

- Characterise the network without knowing the labelling
- Do not depend on labelling or order we take the vertices

• This is very easy to do
  - Number of vertices (|V|)
  - Number of edges (|E|)
  - Number of triangles

• Similar networks should have similar characterisations
  - Tell us something about structure

• We want characterisations which are
  - Informative
  - Easy to compute
  - Invariant to labelling
The *spectrum* of a graph provides a set of graph features (eigenvalues of a matrix representation)

\[ \text{Sp}(L) \]

Can we use the spectrum to determine isomorphism?

\[ \text{Sp}(L) = \text{Sp}(PLP^T) \]

These two graphs have the same spectrum using the adjacency representation

\[
\begin{bmatrix}
5.24 \\
3\right)^2 \\
2 \\
0.76
\end{bmatrix}
\]

\[
\begin{bmatrix}
5.24 \\
3\right)^2 \\
2 \\
0.76
\end{bmatrix}
\]

Necessary but not sufficient...
Cospectral graphs

- How many such graphs are there and does it depend on representation? (Zhu & Wilson 2008)

*50 trillion graphs of size 13
Spectral Features

• The eigendecomposition of a matrix representation is

$$X = U \Lambda U^T$$

• We used the eigenvalues in the spectrum, but there is valuable information in the eigenvectors.
  – Unfortunately the eigenvectors are not invariant $$U \rightarrow PU$$
  – The components are permuted

• Observation:

$$S_2(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3$$

is a polynomial which does not change when the variables are permuted
  – Part of a family of elementary symmetric polynomials invariant to permutation [Wilson & Hancock 2003]

$$S_r(x_{1,\omega} \ldots x_{n,\omega}) = \sum_{i_1 < i_2 < \ldots < i_r} x_{i_1,\omega} x_{i_2,\omega} \ldots x_{i_r,\omega}$$
• Shape graphs distributed by polynomial features [Wilson, Hancock, Luo 2005]
Spectral Features

• These polynomials completely characterise the eigendecomposition
  – Solves the isomorphism problem for graphs with simple spectra (no repeated eigenvalues)
  – There are also other solutions for these graphs
  – Not for other graphs as the eigendecomposition is not unique

• Can reconstruct a graph from a similar set of polynomials
  – In principle, a distribution on the polynomials is generative model of graphs

• Problem: representation not stable enough, similar graphs don’t always have similar representations.
Random Walks

- Spectral features are not tightly coupled to structure
- Can we explore the structure of the network?
- A random walker travels between vertices by choosing an edge at random

- At each time step, a step is taken down an edge
Continuous time random walk

- A continuous-time random walk is the limit of such a walk as the time-steps and transition probabilities become small.
- This is diffusion or heat flow process:

\[
\frac{\partial p}{\partial t} = \nabla^2 p
\]

- On a network, we identify the Laplacian operator \(\nabla^2\) with the Laplacian of the network \(L\):

\[
\frac{\partial p}{\partial t} = -Lp
\]
Heat Kernel

• Solution

\[ p(t) = \mathbf{H}(t) p(0) \]

\[ \mathbf{H}(t) = \exp(-\mathbf{L}t) \]

• Heat kernel \( \mathbf{H}(t) \)

• Trace of \( \mathbf{H} \) is a network characterisation [Xiao, Wilson, Hancock 09]

\[ \text{Tr}[\mathbf{H}(t)] = \sum_i \exp[-\lambda_i t] \]

• Describes a graph based on the shape of heat as it flows across network
  - How much heat is retained at a vertex at time \( t \)

• Shape over time encoded with moments

\[ \mu(s) = \int_0^\infty t^{s-1} \text{Tr}[\mathbf{H}(t)] dt \]
Heat Kernel Moments

COIL database
4 objects x 72 views
Heat Kernel Moments
• Trace only encodes a limited amount of information about the kernel

• Other work on Heat Kernel Signature [Sun, Ovsjanikov, Guibas 2009]
  – Diagonal of heat kernel evaluated at discrete time steps
  – Histogram of values

• Also Wave Kernel Signature [Aubry, Schlickewei, Cremers 2011]
  – Kernel of

\[
\frac{\partial^2 p}{\partial t^2} = -Lp
\]
Walks and Paths

• A random walk on a network of time $t$ generates a path of length $t$
• We can characterise a graph by the numbers of these paths
• For example the total number of paths of length $k$ is $n_k = \sum_{i,j} (A^k)_{ij}$
• Characterise the network by a sequence of numbers $[a_1n_1, a_2n_2, a_3n_3, \ldots]$
• $a$ gets smaller to down-weight the importance of long paths
• Problem with these paths: tottering
• A network with minimum degree 2 can be decomposed into *prime cycles* [Bass 1992, Kotani and Sunada (2000)]

• Prime cycle of a graph:
  - A cycle which has *no backtracking* and is not a multiple of another cycle

![Diagram](Ihara Zeta Function)

Prime

Not Prime (backtracking)

Not Prime (twice round a single cycle)
Ihara zeta function

- Ihara zeta function:

\[ \zeta_G(s) = \prod_{p \text{ prime cycle}} (1 - s^{l(p)})^{-1} \]

- Similar idea to Riemann zeta function

- Depends purely on prime cycle lengths
  - So characterizes graph with prime cycles

- To evaluate, we would need to find all prime cycles
  - A hard problem

- Can we (efficiently) characterize graphs using prime cycles? [Ren, Wilson, Hancock 2011 TNN]
Ihara Zeta Function

- There is another expression for the Ihara zeta function
  \[ \zeta_G(s) = \frac{1}{\det(I - uT)} \]
- \( T \) is a matrix directly derived from the network \( G \)
  - Adjacency matrix of oriented line graph of \( G \) (Hashimoto matrix)
- The determinant is a polynomial in \( u \)
  \[ \det(I - uT) = c_0 + c_1u + c_2u^2 + \ldots + c_mu^m \]
- Can compute these coefficients which are graph characteristics
Ihara coefficients

• These coefficients have a topological basis
  – $-c_3/2$ number of triangles
  – $-c_4/2$ number of squares
  – $-c_5/2$ number of pentagons
  – $-c_6/2$ the number of hexagons plus twice the number of pairs of edge disjoint triangles plus the number of pairs of triangles with a common edge

• OLG is large $O(|V|^2)$
  – Need efficiency gains for large graphs

• Showed how to evaluate efficiently using Bell polynomials [Aziz, Wilson, Hancock 2011]
Backtrackless Walks

- A backtrackless walk is a random walk which does not allow backwards steps (no tottering)
- Can formulate in a similar way to normal walks, but using the matrix $T$ (from Ihara zeta functions)
- More costly to compute but are some tricks to speed up
  - [Aziz, Wilson, Hancock 2013]
- Characterise graph in same way as walk

$$n_k = \sum_{i,j} (T^{k-1})_{ij}$$

$$[a_1n_1, a_2n_2, a_3n_3, \ldots]$$
MUTAG
Collection of 188 labelled chemical compounds.
Task is to predict whether each compound has mutagenicity or not.

\[
\text{\begin{tabular}{c}
\includegraphics[width=0.5\textwidth]{chemical_structure.png}
\end{tabular}}
\]
### Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random walk kernel</td>
<td>Mutag(labelled)</td>
<td>90.0%</td>
</tr>
<tr>
<td>Backtrackless walk kernel</td>
<td>Mutag(labelled)</td>
<td>91.1%</td>
</tr>
<tr>
<td>Feature vector from Random walk</td>
<td>COIL(unlabeled)</td>
<td>94.4%</td>
</tr>
<tr>
<td>Feature vector from backtrackless random walk</td>
<td>COIL(unlabeled)</td>
<td>95.5%</td>
</tr>
<tr>
<td>Feature vector from Ihara coefficients</td>
<td>COIL(unlabeled)</td>
<td>94.4%</td>
</tr>
<tr>
<td>Shortest Path Kernel</td>
<td>COIL(unlabeled)</td>
<td>86.7%</td>
</tr>
<tr>
<td>Feature vector from Random walk</td>
<td>Mutag(unlabeled)</td>
<td>89.4%</td>
</tr>
<tr>
<td>Feature vector from backtrackless random walk</td>
<td>Mutag(unlabeled)</td>
<td>90.5%</td>
</tr>
<tr>
<td>Feature vector from Ihara coefficients</td>
<td>Mutag(unlabeled)</td>
<td>80.5%</td>
</tr>
</tbody>
</table>
Kernel Signatures Revisited

- Kernel signatures are based on the discrete Laplacian $L$

$$\frac{dH(t)}{dt} = -LH(t) \quad \frac{\partial^2 p}{\partial t^2} = -Lp$$

- Discrete Laplacian has connectivity but no length
  - No transmission time for signals
  - All signal present on some vertex
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
• The heat kernel signature of a point on the mesh is found by sampling the retained heat at a number of time points

\[ \text{HKS} = [H_{t_0}(x,x), H_{t_1}(x,x), H_{t_2}(x,x), \ldots] \]

• Need to calculate a representative adjacency matrix

\[
M_{ij} = \begin{cases} 
\cot \alpha_{ij} + \cot \beta_{ij} & (i, j) \in E \\
2 & \text{otherwise}
\end{cases}
\]

\[
S_{ii} = \text{Area}(i)
\]

\[
A = S^{1/2}MS^{1/2}
\]

• Captures geometric information about the shape
EHKS

• Can compute the eigenfunctions of edge-based Laplacian to obtain spectral form for heat kernel

\[ H_t(x, y) = \sum_{\omega} e^{-\omega^2 t} \phi_{\omega}(x)\phi_{\omega}(y) \]

• Edge-based HKS uses this kernel to compute the signature
• EHKS defined at every point on mesh edges as well as vertices
SHREC 2010 database
10 shapes with 20 nonrigid deformations in each shape class

Robust to deformations and noise
Clustering of different body parts of a human body using EHKS.
Global HKS is a histogram of the local signatures
Classification using GEHKS

Good separation of classes
Hard Networks

• Some networks are very hard to tell apart

• Strongly regular
  – Every vertex has 6 edges
  – Every adjacent pair of vertices has 2 common neighbours
  – Every non-adjacent pair of vertices has 2 common neighbours
Discrete-Time Random Walks

- By taking \( n \) steps of a random walk, we can explore more of the graph structure
  - Characterised by \( T^n \)
- The support of a random walk is the graph created by joining the vertices with a path between them
  - Replace the non-zero entries of \( T^n \) with an edge \( S(T^n) \)
- We can use the spectrum of the support to characterise the graph
- Unfortunately, \( \text{Sp}[S(T^n)] \) cannot distinguish strongly regular graphs for any value of \( n \) [Emms, Severini, Wilson, Hancock 2009]
Quantum Walk

- A quantum walker is different from a random walker in a number of ways
  - Unitary transition matrix $U$
  - Can be at a superposition of different vertices at the same time
  - Different paths can interfere and cancel out

- Quantum evolution is reversible, so a step must depend on the source and target vertex $(i,j)$
  - Walk on edges

- $U$ constructed from coin operator and transition
  - Grover coin gives real $U
• Since the walk can have positive or negative amplitude, we can look at the positive support of the walk
  – Graph representing the paths which have positive amplitude in the walk
  – Structure dependent on interference patterns on the graph $S^+(U^n)$

• Spectra of the support on some strongly regular graphs [Emms, Severini, Wilson, Hancock 2009]
  – $S^+(U)$ cannot distinguish SRGs
  – $S^+(U^2)$ cannot distinguish SRGs
  – $S^+(U^3)$ distinguishes all available SRGs (>33,000 graphs)
  – The third step is the first time the walk can return to the same location and create interference
Finally...

- There are many ways to characterise graphs
- Spectral theory and random walks provide some very useful methods
- Still challenging to use these methods on very large real-world networks