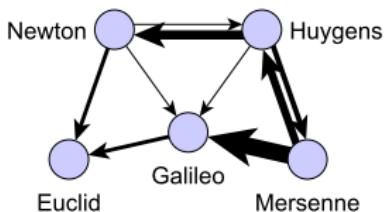




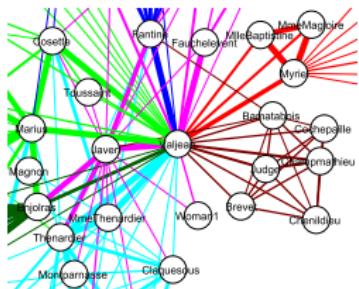
Complexity & Networks

Part II: Networks

(PHYS96008)



Tim EVANS
 Theory group, and
 Centre for Complexity Science
 Imperial College London



Spring 2021

Table of Contents

1. What is a NETWORK?
2. Basic Network Analysis
3. Fat-Tailed Distributions
4. Network Models
5. Phase Transitions in Random Graphs
6. Network Measures without Eigenvalue Calculations
7. Processes on Networks

1. What is a NETWORK?

- 1.1 Mathematical Definition
- 1.2 Why use a network?
- 1.3 From Data to Network
- 1.4 Classification of Networks by Edge Type
- 1.5 Classification of Networks by Edge Density
- 1.6 Representation of Networks
- 1.7 Visualisation

C & N Part II: Networks Lecture 1

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

Why Networks?

Intended Learning Outcomes

- Students can define a graph/network mathematically
- Students what is captured in a network representation.

PS= See problem sheet, **EFS**= Exercise For Students

1. What is a NETWORK?

1.1 Mathematical Definition

1.2 Why use a network?

1.3 From Data to Network

1.4 Classification of Networks by Edge Type

1.5 Classification of Networks by Edge Density

1.6 Representation of Networks

1.7 Visualisation

What is a NETWORK?

A network is known as a GRAPH in maths.

Formal definition of a graph \mathcal{G} is

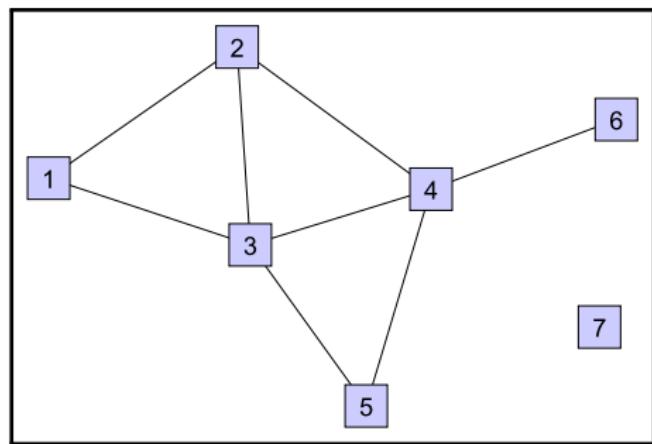
- A set of VERTICES \mathcal{V} .
 $N = |\mathcal{V}|$ is the number of vertices.
- A set of EDGES \mathcal{E} which are *pairs* of vertices taken from \mathcal{V} .
 $E = |\mathcal{E}|$ is the number of edges.

Notation

- The graph is often written as the pair of vertex and edge sets
 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
- Edges are written as a pair of vertices (i, j)
where i and j are vertices in the graph, i.e. $i, j \in \mathcal{V}$

Simple example

- Vertices may have no edges
e.g. vertex 7
- Graphs may have several disconnected parts
called COMPONENTS.
These are special cases of
SUBGRAPHS.
e.g. vertices 1 to 6 and their
edges form one component,
vertex 7 is a component by
itself.
- Natural visualisation shown



Vertex Set $\mathcal{V} = \{1, 2, 3, 4, 5, 6, 7\}$

Edge Set

$$\mathcal{E} = \{(1, 2), (1, 3), (2, 3), (2, 4), (3, 4), (3, 5), (4, 5), (4, 6)\}$$

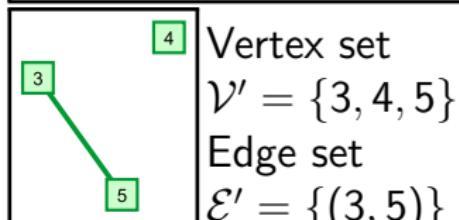
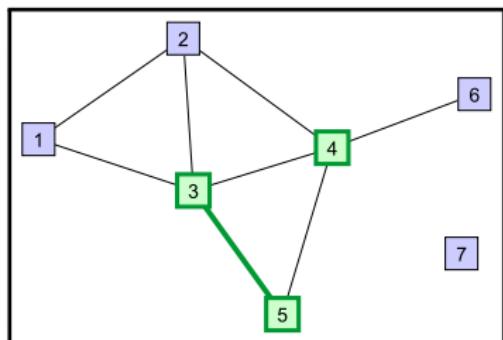
Subgraphs

Subgraphs

A SUBGRAPH $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a **graph** made from a subsets of the vertex and edge sets of original graph \mathcal{G} .

So $\mathcal{V}' \subseteq \mathcal{V}$, $\mathcal{E}' \subseteq \mathcal{E}$ is necessary but *not sufficient*.

\mathcal{V}' and \mathcal{E}' must also form a graph in their own right.



This is a valid subgraph

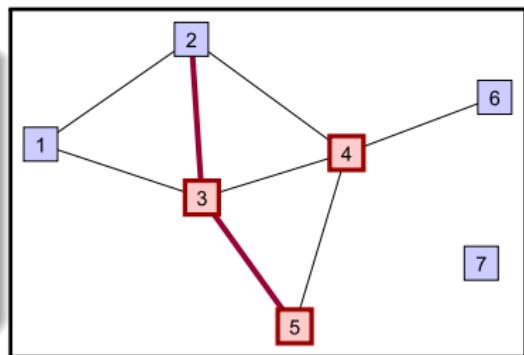
Not a subgraph

Subgraphs

A SUBGRAPH $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a *graph* made from a subsets of the vertex and edge sets of original graph \mathcal{G} .

So $\mathcal{V}' \subseteq \mathcal{V}$, $\mathcal{E}' \subseteq \mathcal{E}$ is necessary but *not sufficient*.

\mathcal{V}' and \mathcal{E}' must also form a graph in their own right.



Vertex set $\mathcal{V}' = \{3, 4, 5\}$

Edge set $\mathcal{E}' = \{(2, 3), (3, 5)\}$

Not a subgraph as $2 \notin \mathcal{V}'$.

Terminology

Several names are used for vertices and edges in different fields.
Even the name for a network is not universal!

“point”	“line”	“combination”	typical field
vertex	edge	graph	Mathematics
node	link	network	Engineering, Computer Science
site	bond	network	Physics
actor	tie	network	Social Science

Each ‘end’ of an edge is sometimes called a STUB.



1. What is a NETWORK?

1.1 Mathematical Definition

1.2 Why use a network?

1.3 From Data to Network

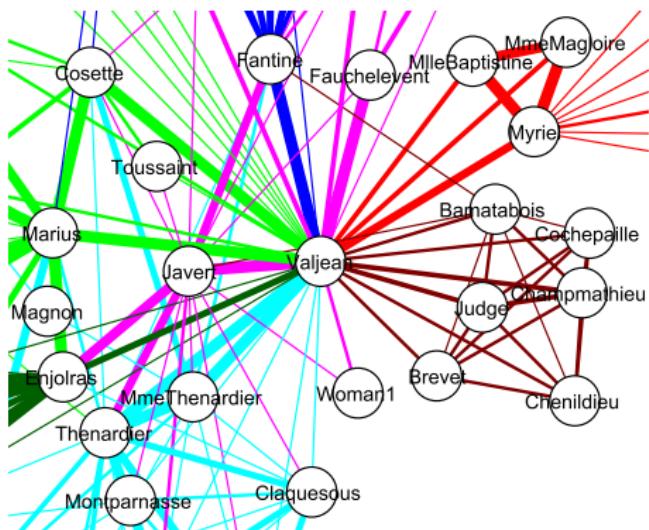
1.4 Classification of Networks by Edge Type

1.5 Classification of Networks by Edge Density

1.6 Representation of Networks

1.7 Visualisation

Why Use a Network?



WHY?

Characters in *Les Misérables* linked if appearing in same scene.

[Evans and Lambiotte, 2009]

Bilateral relationships

A network captures **bilateral relationships** between objects.

A Vertex can be *anything* you like.

The edges record *some relationship* between *pairs* of objects

Network	Vertex	Edge
social	person	friendship (enmity)
genetic	person	at least 50% shared DNA
organisation	person	line of command
world wide web	web pages	hyperlink
internet	network router	communication link
citation	paper	entry in bibliography
protein network	proteins	participate in same reaction
food web	species	predator-prey
travel	location	trips made between locations
street map	intersections	part of a street

Bilateral relationships

A network captures **bilateral relationships** between objects.

A Vertex can be *anything* you like.

The edges record *some relationship* between *pairs* of objects

Examples?

Network	Vertex	Edge
?	?	?
:	:	:

Network	Vertex	Edge
social	person	friendship (enmity)
genetic	person	at least 50% shared DNA
organisation	person	line of command
world wide web	web pages	hyperlink
internet	network router	communication link
citation	paper	entry in bibliography

Bilateral relationships

A network captures **bilateral relationships** between objects.

A Vertex can be *anything* you like.

The edges record *some relationship* between *pairs* of objects

Network	Vertex	Edge
social	person	friendship (enmity)
genetic	person	at least 50% shared DNA
organisation	person	line of command
world wide web	web pages	hyperlink
internet	network router	communication link
citation	paper	entry in bibliography
protein network	proteins	participate in same reaction
food web	species	predator-prey
travel	location	trips made between locations
street map	intersections	part of a street

A Network is not just Bilateral relationships

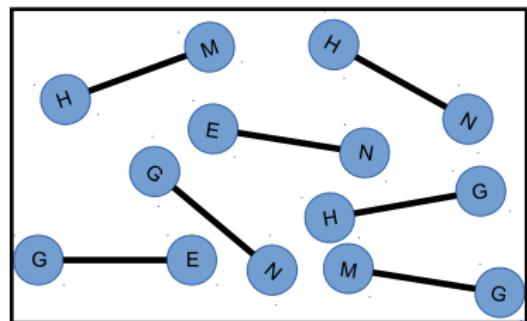
Not a network

A network is more than a collection of edges (bilateral relationships)

You could do statistics on pairs but you do not need a network for that.

e.g.

relationships between colours of trousers and tops worn by people.



Link indicates one mathematician mentioned in other's biography:

E = Euclid, G = Galileo,

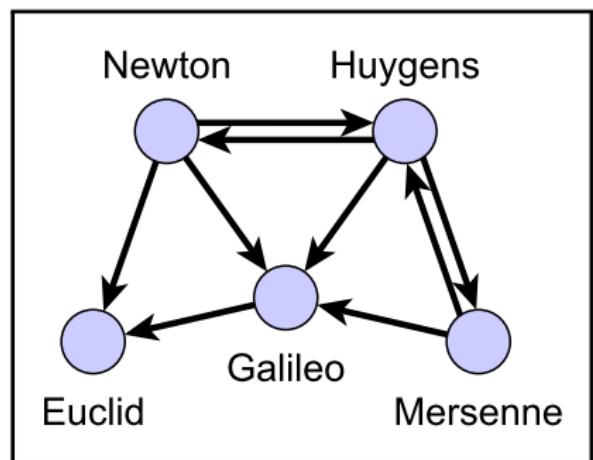
H = Huygens, M = Mersenne,

N = Newton.

A Network is about friends of friends of friends ...

A Network of Relationships

A network is about the interconnections of the whole set of vertices



From MacTutor website
[http://www-history.mcs.
st-and.ac.uk/](http://www-history.mcs.st-and.ac.uk/)

Social Network Analysis

Social science has used network analysis for decades, a topic known as SOCIAL NETWORK ANALYSIS.

(see Wasserman & Faust 1994 sec.1.2, or Freeman 2004 for history)

A Survey at Imperial

Let people at Imperial be the vertices.

- The official hierarchy: edges define chain of command.
[Rector] - [Head of Faculty] - [Head of Department]
- [Head of Group] - [Staff member] - [Student]
- Actual working relationships
- Friendships (and perhaps enmities) as the edges.

Aside: Hypergraphs

Social science stresses often stresses role of triads (triples)

e.g

- BALANCE THEORY (Heider 1958).
- STRUCTURAL HOLES (Burt 1995)

Suggests we use HYPERGRAPHS

These record general relationships between two or more nodes.

Hyperedges are subsets of two or more vertices from the vertex set.

C & N Part II: End of Networks Lecture 1

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “Why Networks”

Intended Learning Outcomes

- Students can define a graph/network mathematically
- Students what is captured in a network representation.

C & N Part II: Networks Lecture 2

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

The Zoo of Network Types

Intended Learning Outcomes

- Students will know the most important types of edge
- Students will know the most important types of network

PS= See problem sheet, **EFS**= Exercise For Students

1. What is a NETWORK?

1.1 Mathematical Definition

1.2 Why use a network?

1.3 From Data to Network

1.4 Classification of Networks by Edge Type

1.5 Classification of Networks by Edge Density

1.6 Representation of Networks

1.7 Visualisation

From Data to Network

- ★ Vertices can represent many different objects.
- ★ Edges capture many different bilateral relationships.

One data set, many networks:-

- There is no one network representation of a data set.
- The 'obvious' network representation not always the best.
- A network may not capture all information available.
- A network may not be the best way of tackling a problem.

Reasonable Aim(?)

A network is a simple representation capturing the essential information about objects and their relationships in some system.

From Data to Network

- ★ Vertices can represent many different objects.
- ★ Edges capture many different bilateral relationships.

One data set, many networks:-

- There is no one network representation of a data set.
- The 'obvious' network representation not always the best.
- A network may not capture all information available.
- A network may not be the best way of tackling a problem.

Reasonable Aim(?)

A network is a simple representation capturing the essential information about objects and their relationships in some system.

From Data to Network

- ★ Vertices can represent many different objects.
- ★ Edges capture many different bilateral relationships.

One data set, many networks:-

- There is no one network representation of a data set.
- The 'obvious' network representation not always the best.
- A network may not capture all information available.
- A network may not be the best way of tackling a problem.

Reasonable Aim(?)

A network is a simple representation capturing the essential information about objects and their relationships in some system.

Examples of Different Vertices

Consider a set of web pages

Vertices

Web Pages the obvious network

Semantic all pages on one topic are one vertex

Host group pages by location of server

Text words are vertices

Newton's Web Page

Newton and **Huygens** exchanged letters.

Newton was very interested in friction.

Huygens' Web Page

Huygens investigated friction.

Huygens often wrote to **Newton**.

Examples of Different Relationships

Take individual web pages as the vertices, and then edges could be
Edges

Hyperlinks the obvious network

Co-citation edge indicates two pages point to the same third page

Semantic edge between pages sharing similar common words

Newton's Web Page

Newton and **Huygens**
exchanged letters.

Newton was very
interested in friction.

Huygens' Web Page

Huygens investigated
friction.

Huygens often wrote to
Newton.

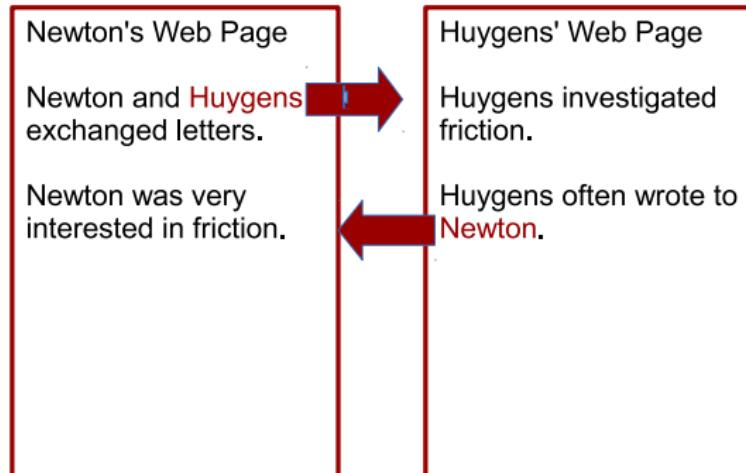
Examples of Different Relationships

Take individual web pages as the vertices, and then edges could be
Edges

Hyperlinks the obvious network

Co-citation edge indicates two pages point to the same third page

Semantic edge between pages sharing similar common words



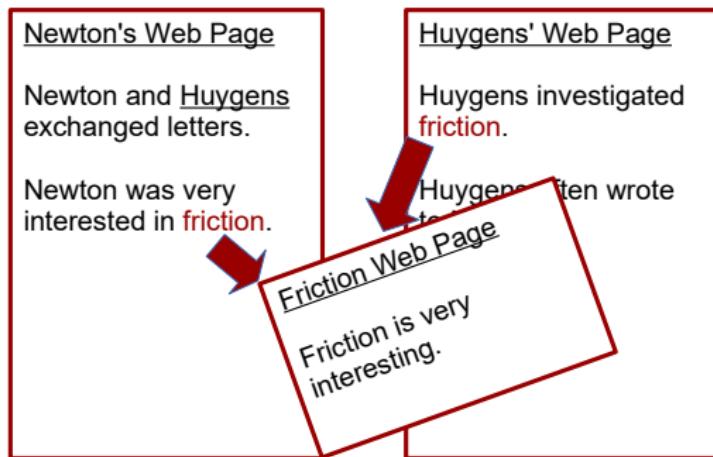
Examples of Different Relationships

Take individual web pages as the vertices, and then edges could be
Edges

Hyperlinks the obvious network

Co-citation edge indicates two pages point to the same third page

Semantic edge between pages sharing similar common words



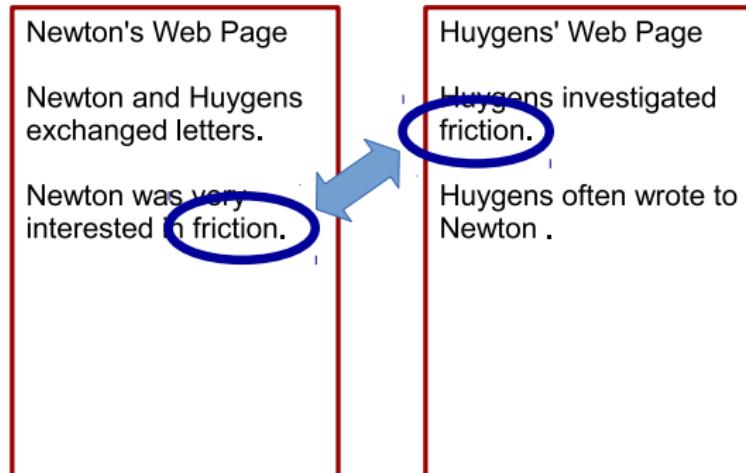
Examples of Different Relationships

Take individual web pages as the vertices, and then edges could be
Edges

Hyperlinks the obvious network

Co-citation edge indicates two pages point to the same third page

Semantic edge between pages sharing similar common words



One Data Set, Many Networks

Different types of vertex carry different information.

Different types of edge carry different information.

1. What is a NETWORK?

1.1 Mathematical Definition

1.2 Why use a network?

1.3 From Data to Network

1.4 Classification of Networks by Edge Type

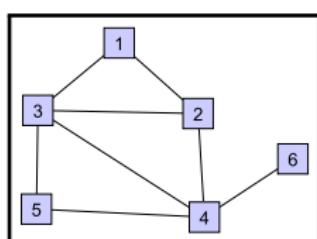
1.5 Classification of Networks by Edge Density

1.6 Representation of Networks

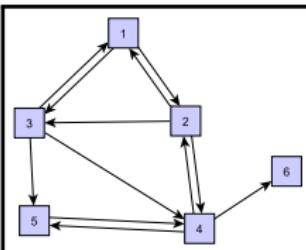
1.7 Visualisation

Types of network by Edge Type

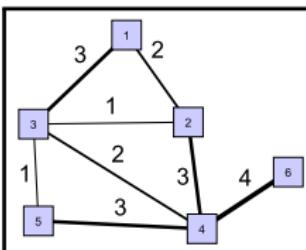
We can classify networks by their edge type:-



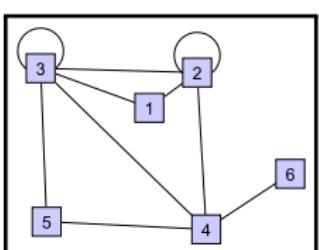
SIMPLE



DIRECTED

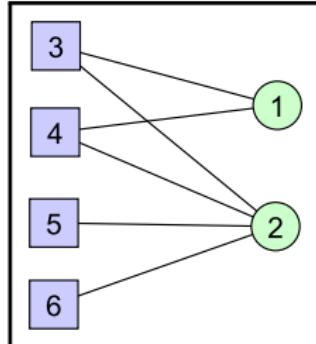


WEIGHTED

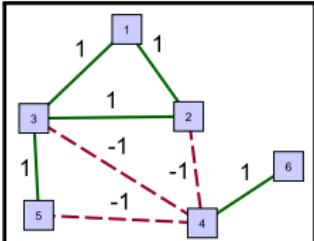


with SELF-LOOPS

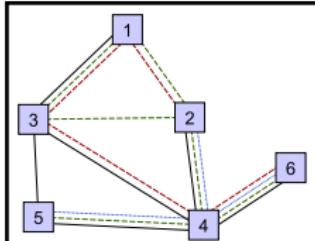
BIPARTITE



SIGNED



MULTIPLEX

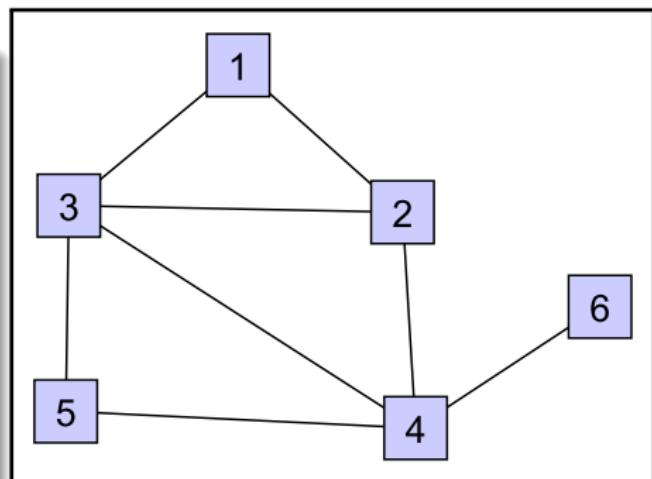


Simple Graphs

Simple Graph

A simple graph has

- at most one edge between any two pairs of vertices
- no values (WEIGHTS) on edges
- no SELF-LOOPS
 - an edge where both ends are connected to the same vertex.



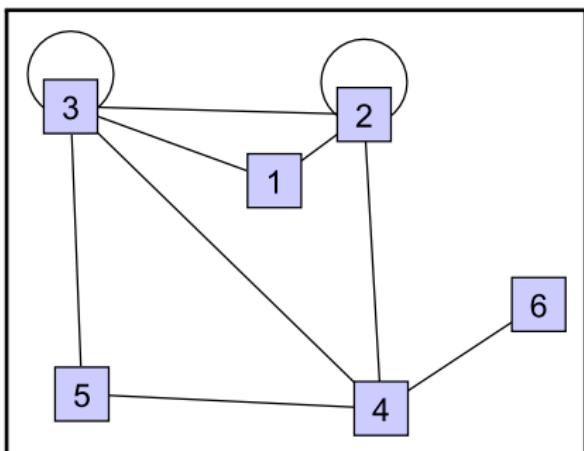
Self-Loops

Self-Loops

A SELF-LOOP is an edge where both ends are connected to the same vertex.

Self-loops sometimes have no meaning, are irrelevant to analysis or don't appear at all.

e.g. *cannot be a friend with yourself.*



Sometimes self-loops are very useful

e.g. vertex = town,
edge = commuting trip,
self-loop = commuters who live and work
in same town.

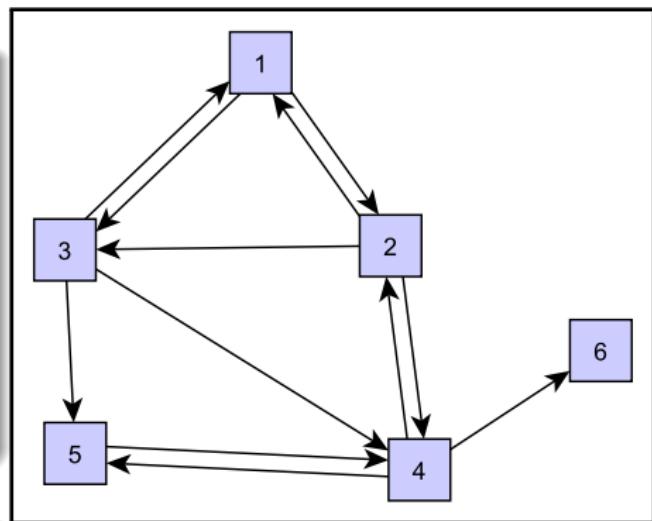
Directed graph

Directed graph

- Each edge has a direction.
- Arrows point away from SOURCE vertices towards TARGET vertices.
- Edges are *ordered pairs* (s, t) so $(s, t) \neq (t, s)$.

Web pages

The hyperlinks between websites (the vertices) show this type of structure.



- RECIPROCATED edges
e.g. $(1, 3)$ and $(3, 1)$
- UNRECIPROCATED
e.g. $(3, 5)$.

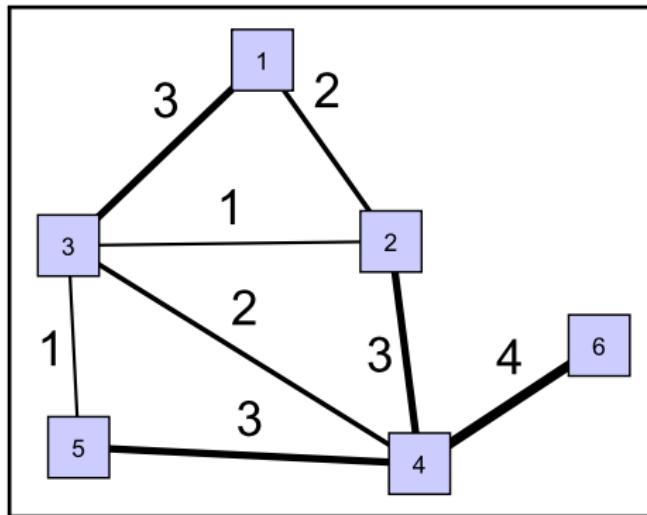
Weighted graph

Weighted graph

Each edge has a real value, the EDGE WEIGHT.

Phone call network

Value of edge records number or length of calls between people.



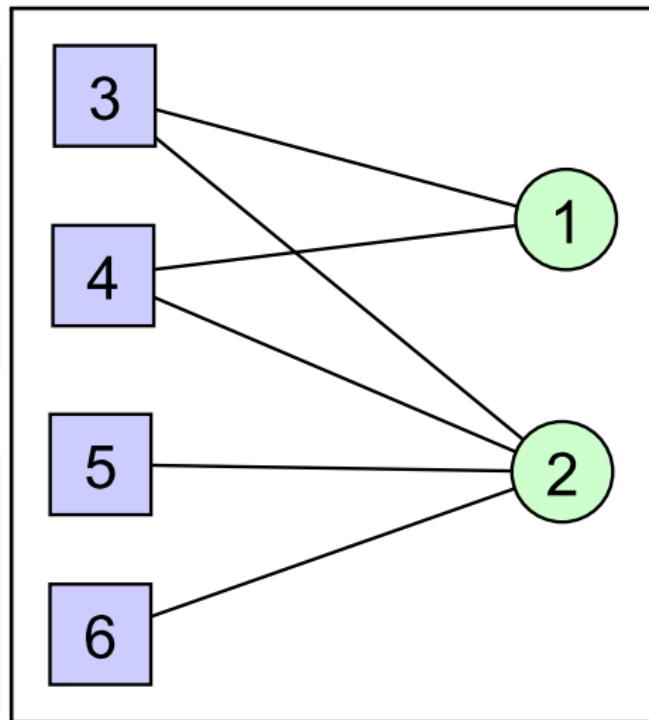
Bipartite network

Bipartite network

A BIPARTITE NETWORK (a TWO-MODE NETWORK in social sciences) has two types of vertex *and* edges **always** run between vertices of opposite type.

Academic Papers

The square vertices represent authors and the circles are the papers. Links represent authorship of publications.

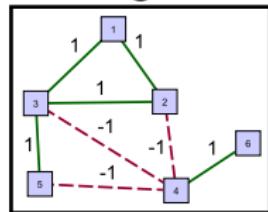


Many More Types of Network (#)

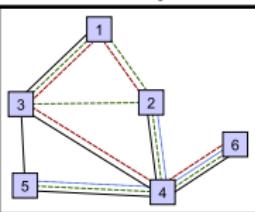
Many other types of networks exist

Classified by their edge type, with more complicated types of vertex, etc etc.

- Signed



- Multiplex



- Boolean Network
- Multilayer Network
- ...

We do not need other networks in this course but some are mentioned in the notes.

1. What is a NETWORK?

- 1.1 Mathematical Definition
- 1.2 Why use a network?
- 1.3 From Data to Network
- 1.4 Classification of Networks by Edge Type
- 1.5 Classification of Networks by Edge Density**
- 1.6 Representation of Networks
- 1.7 Visualisation

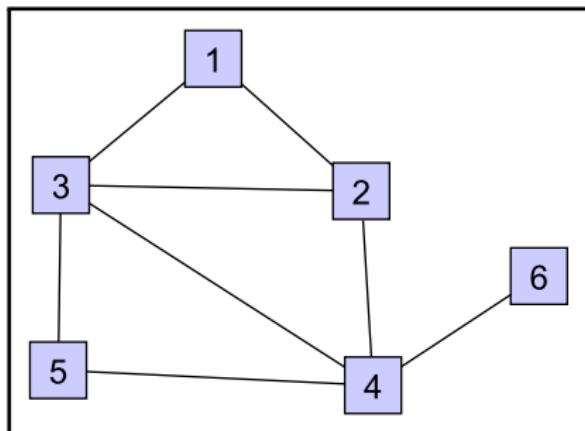
??

ADD TREE?

??

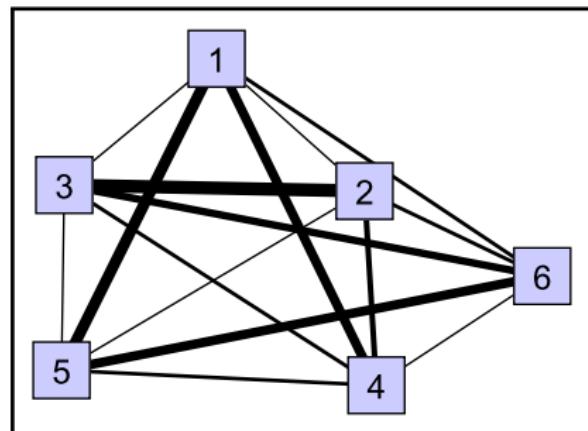
Sparse and Dense Networks

Networks classified by DENSITY of edges in the network,
i.e. fraction of possible edges present.



SPARSE

- Most nodes connected to only a few other nodes.
- Many nodes pairs are not connected.



DENSE

- All nodes connected to almost all other nodes.
- Few nodes pairs unconnected.

Sparse and Dense Networks

Sparse Networks

A SPARSE NETWORK is one with a

- number of edges similar to number of nodes
- finite number of edges attached to each vertex
(finite average degree, $\langle k \rangle$)
- many zero entries in its *adjacency matrix* (see later F1.40)

All three definitions are equivalent.

Numerically a SPARSE MATRIX REPRESENTATION can be exploited.
(e.g. edge list, adjacency lists)

Dense Networks

A DENSE NETWORK has many more edges than a sparse network, it is close to a COMPLETE NETWORK

These are very woolly definitions!

Sparse and Dense Networks

Sparse Networks

A SPARSE NETWORK is one with a

- number of edges similar to number of nodes
- finite number of edges attached to each node (finite average degree, $\langle k \rangle$)
- many zero entries in its *adjacency matrix* (see later)



All three definitions are equivalent.

Numerically a SPARSE MATRIX REPRESENTATION can be exploited.
(e.g. edge list, adjacency lists)

Dense Networks

A DENSE NETWORK has many more edges than a sparse network, it is close to a COMPLETE NETWORK

These are very woolly definitions!

Sparse and Dense Networks precisely

Sparse Networks

As $N \rightarrow \infty$ a SPARSE NETWORK has

- $E \sim O(N^1)$
- $\langle k \rangle \sim O(1)$

Maximum number of edges is $O(N^2)$
e.g. simple graph has

Dense Networks

As $N \rightarrow \infty$ a DENSE NETWORK has

- $E \sim O(N^2)$
- $\langle k \rangle \sim O(N)$

$$\frac{N(N - 1)}{2}$$

★ These definitions only useful in mathematical models ★
where $N \rightarrow \infty$ is defined.

Complex Networks

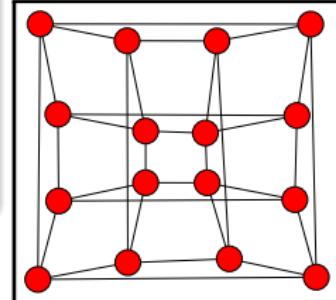
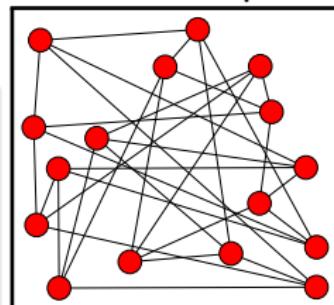
No precise definition, here is my attempt:-

Complex Networks

- A COMPLEX NETWORK is a network formed through local interactions
It *emerges* from activity which has no detailed knowledge of the whole shape
e.g. authors of web sites, c.f. flocks of birds
- A COMPLEX NETWORK has little or no symmetry

Contrast this with crystal structures

Random Graph

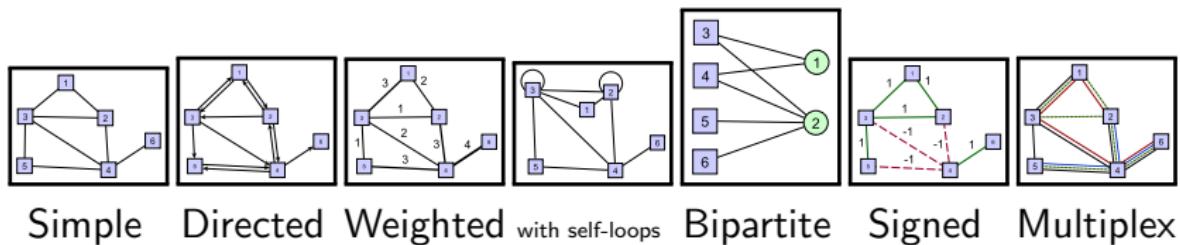


Regular Lattice

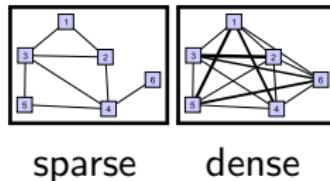
Summary of Network Classification

Networks classified by the type of edge

— the information recorded in the bilateral relationships



Networks classified by edge density



Complex Networks

— large scale structures emerging from individual bilateral interactions

C & N Part II: End of Networks Lecture 2

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “The Zoo of Network Types”

Intended Learning Outcomes

- Students will know the most important types of edge
- Students will know the most important types of network

C & N Part II: Networks Lecture 3

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

Network Representation

Intended Learning Outcomes

- Students will be able to choose an appropriate representation of a network for numerical and mathematical work.
- Students will be aware of the issues around network visualisation.

PS= See problem sheet, **EFS**= Exercise For Students

1. What is a NETWORK?

1.1 Mathematical Definition

1.2 Why use a network?

1.3 From Data to Network

1.4 Classification of Networks by Edge Type

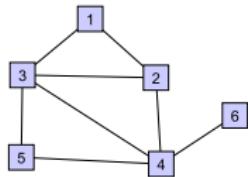
1.5 Classification of Networks by Edge Density

1.6 Representation of Networks

1.7 Visualisation

Different Network Representations

Several different ways to represent one network for numerical or algebraic analysis, including:-



Edge List

source	target
1	2
1	3
2	3
2	4
3	4
3	5
4	5
4	6

Adjacency List

source	neighbours
1	2,3
2	1,3,4
3	1,2,4,5
4	2,3,5,6
5	3,4
6	4

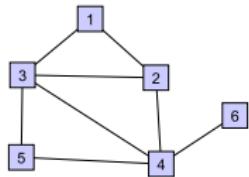
Adjacency Matrix

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

There are other less common methods useful in specialised circumstances e.g. the INCIDENCE MATRIX.

Different Network Representations

Several different ways to represent one network for numerical or algebraic analysis, including:-



Edge List

source	target
1	2
1	3
2	3
2	4
3	4
3	5
4	5
4	6

Adjacency List

source	neighbours
1	2,3
2	1,3,4
3	1,2,4,5
4	2,3,5,6
5	3,4
6	4

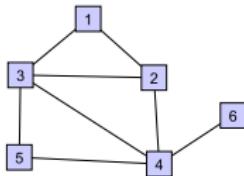
Adjacency Matrix

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

There are other less common methods useful in specialised circumstances e.g. the INCIDENCE MATRIX.

Different Network Representations

Several different ways to represent one network for numerical or algebraic analysis, including:-



Edge List

source	target
1	2
1	3
2	3
2	4
3	4
3	5
4	5
4	6

Adjacency List

source	neighbours
1	2,3
2	1,3,4
3	1,2,4,5
4	2,3,5,6
5	3,4
6	4

Adjacency Matrix

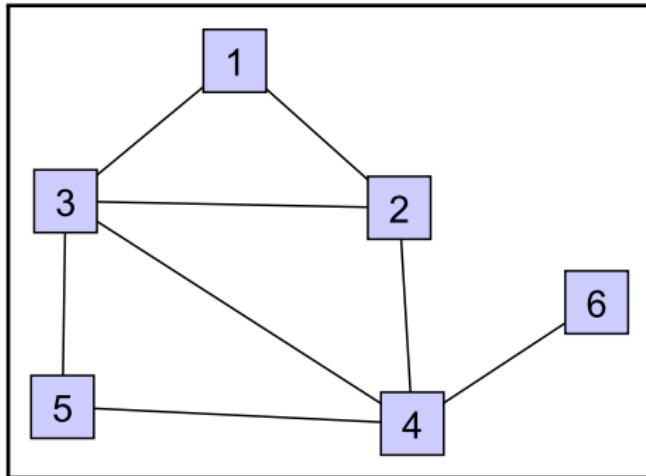
$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

There are other less common methods useful in specialised circumstances e.g. the INCIDENCE MATRIX.

Edge List Representation

Edge List

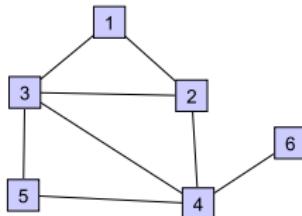
A simple list of the source and target vertices of each edge, with weight of edge if needed.



source	target
1	2
1	3
2	3
2	4
3	4
3	5
4	5
4	6

Edge List Representation Features

1	2
1	3
2	3
2	4
3	4
3	5
4	5
4	6

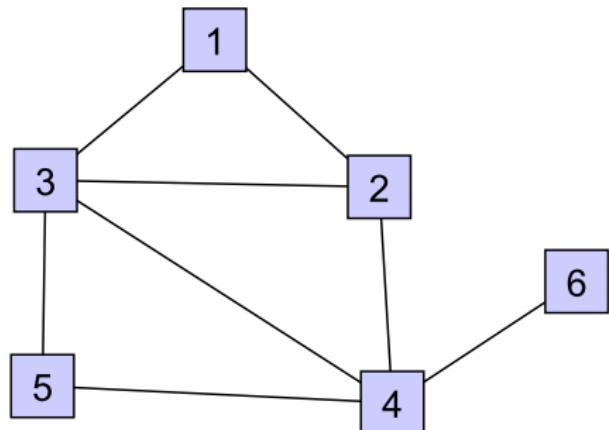


- Good for storing as a data file
 - Simple
 - Sparse, only records edges which are present.
- Poor for programmes
 - Slow to find the edges of one vertex.
- Misses vertices with no edges (fixable)
- Not good for algebraic manipulations

Adjacency List

Adjacency List

A list of the neighbours of each vertex.

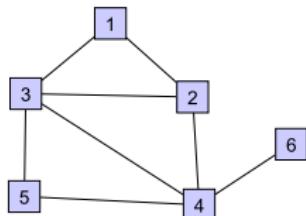


source	neighbours
1	2,3
2	1,3,4
3	1,2,4,5
4	2,3,5,6
5	3,4
6	4

So this is a list where each entry is another list, a list of all the neighbours of the source vertex.

Adjacency List Representation Features

source	neighbours
1	2,3
2	1,3,4
3	1,2,4,5
4	2,3,5,6
5	3,4
6	4



Good data structure for a network
in a computer programme
— a list of lists

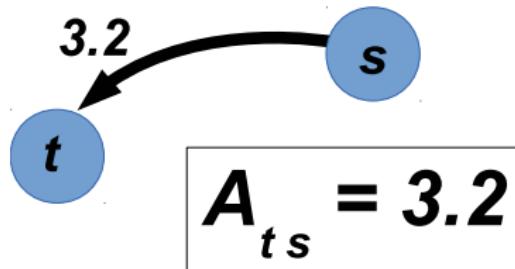
- Sparse, only records edges which are present.
- Easy to find properties of each vertex.
- Perfect representation of network

Not useful for algebraic manipulations

Adjacency Matrix Representation

Adjacency matrix

The ADJACENCY MATRIX A is defined such that A_{ts} is the weight (value) of the edge running from source vertex s to target vertex t .



- We will assume that there is at most one edge for every pair of vertices if the network is undirected.
- If edges have a direction, we will allow at most one edge in each direction for every pair of vertices.
- True or sufficient for many actual problems
⇒ adjacency matrix is a complete representation.

★ Order of indices not fixed, first index may be source or target ★

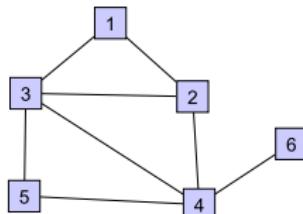
Adjacency Matrix Representation

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Good data structure for
mathematics
— linear algebra

Poor for computers

- Dense, records presence *and* absence of edges
- Slow to find properties of each vertex.



(#) Aside:

There are some very fast numerical routines for matrices and some special matrix classes for sparse matrices so sometimes a numerical approach to sparse networks using adjacency matrices may be possible.

Properties of the Adjacency matrix

SIMPLE NETWORK: Adjacency matrix is symmetric $A = A^T$, and 0 or 1 entries and zero diagonal.

DIRECTED EDGES: Adjacency matrix is not symmetric $A \neq A^T$.

WEIGHTED EDGES: Entries of adjacency matrix are not restricted to 0 or 1. A_{ts} is the weight of edge (s, t) .

SELF-LOOPS: These are the diagonal entries of the adjacency matrix $A_{ii} \neq 0$
(no sum over repeated index i).

BIPARTITE NETWORK: Adjacency matrix can be arranged to be in an off-diagonal block form.

What are the properties of these networks?

Properties of the Adjacency matrix

SIMPLE NETWORK: Adjacency matrix is symmetric $A = A^T$, and 0 or 1 entries and zero diagonal.

DIRECTED EDGES: Adjacency matrix is not symmetric $A \neq A^T$.

WEIGHTED EDGES: Entries of adjacency matrix are not restricted to 0 or 1. A_{ts} is the weight of edge (s, t) .

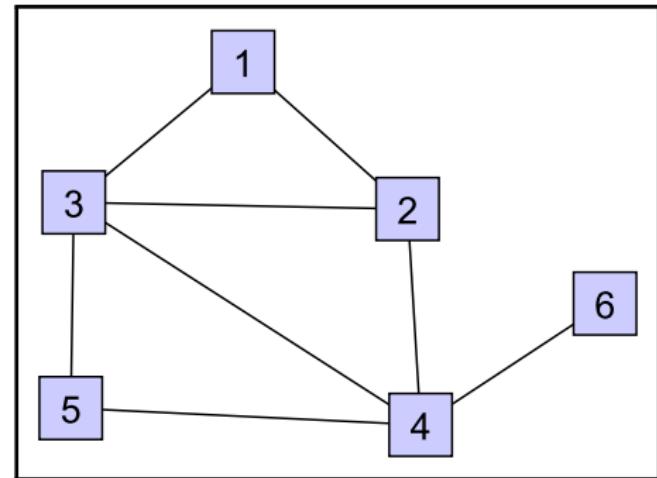
SELF-LOOPS: These are the diagonal entries of the adjacency matrix $A_{ii} \neq 0$
(no sum over repeated index i).

BIPARTITE NETWORK: Adjacency matrix can be arranged to be in an off-diagonal block form.

What are the properties of these networks?

Simple Network Adjacency matrix

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & \textcolor{red}{1} \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & \textcolor{red}{1} & 0 & 0 \end{pmatrix}$$

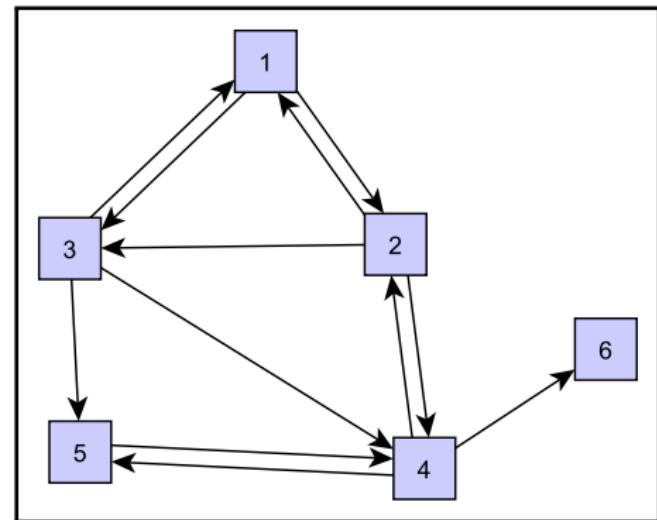


- symmetric
- entries 0 or 1
- 0 on diagonal (no self-loops)

Entries for Edge (4,6) in red/bold

Directed Network Adjacency matrix

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & \textcolor{red}{0} \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & \textcolor{red}{1} & 0 & 0 \end{pmatrix}$$



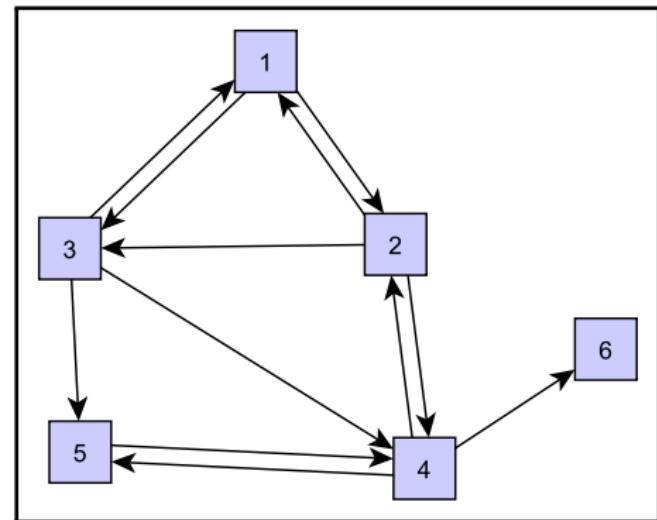
- Adjacency matrix asymmetric
 $A \neq A^T$.
- Here edge A_{ts} runs
from source s
to target t

Entries for Edge (4,6) in red/bold

★ Order of indices varies, always check conventions

Directed Network Adjacency matrix

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & \textcolor{red}{0} \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & \textcolor{red}{1} & 0 & 0 \end{pmatrix}$$



- Adjacency matrix asymmetric
 $A \neq A^T$.
- Here edge A_{ts} runs
from source s
to target t

Entries for Edge (4,6) in red/bold

★ Order of indices varies, always check conventions



Weighted Network Adjacency matrix

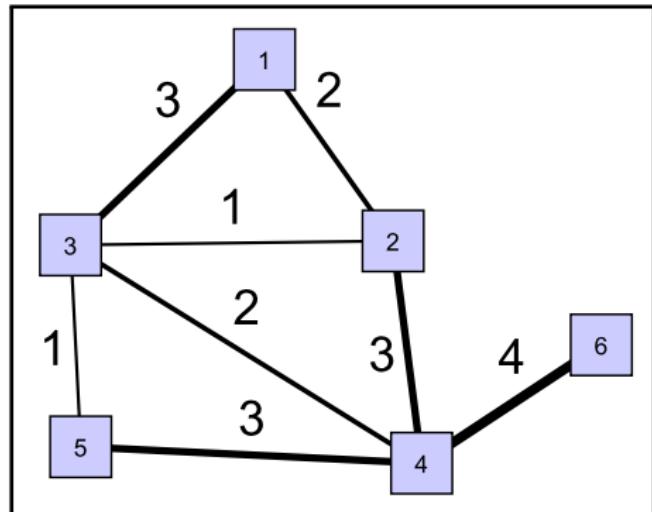
$$\begin{pmatrix} 0 & 2 & 3 & 0 & 0 & 0 \\ 2 & 0 & 1 & 3 & 0 & 0 \\ 3 & 1 & 0 & 2 & 1 & 0 \\ 0 & 3 & 2 & 0 & 3 & \textcolor{red}{4} \\ 0 & 0 & 1 & 3 & 0 & 0 \\ 0 & 0 & 0 & \textcolor{red}{4} & 0 & 0 \end{pmatrix}$$

In this course

- Entries real $A_{ij} \in \mathbb{R}$.
- Entries semi-positive $A_{ij} \geq 0$.

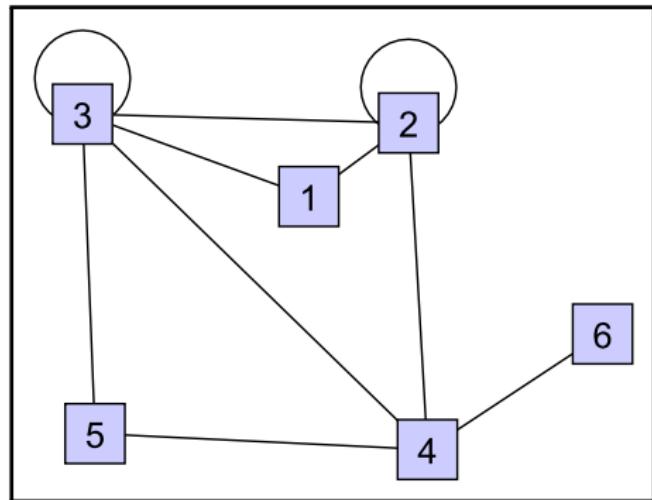
Other choices possible but rarer.

Entries for Edge (4,6) in red/bold



Network with self-loops Adjacency matrix

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & \textbf{2} & 1 & 1 & 0 & 0 \\ 1 & 1 & \textbf{2} & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$



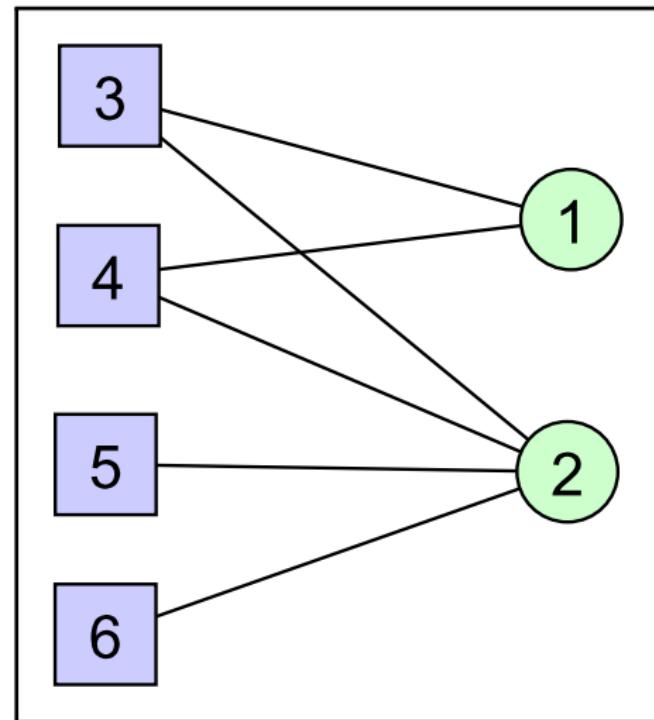
- Non-zero entries on diagonal
real $A_{ii} \neq 0$
- Here 2 used as 2 stubs per self-edge.
- Other representations possible

Entries for self-loops in red/bold.

Bipartite Network Adjacency matrix

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- Block off-diagonal form.



1. What is a NETWORK?

1.1 Mathematical Definition

1.2 Why use a network?

1.3 From Data to Network

1.4 Classification of Networks by Edge Type

1.5 Classification of Networks by Edge Density

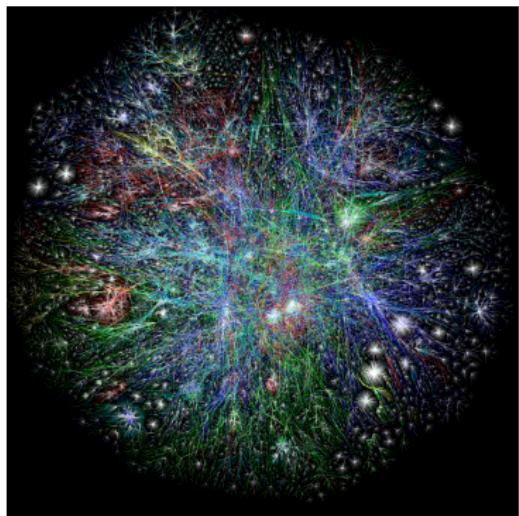
1.6 Representation of Networks

1.7 Visualisation

Visual Representation of Networks

- Vertices usually have no spatial coordinates.
- Vertices may be placed anywhere in a figure.
- Visualisation of networks is both an art and a science.
- The presentation of any data is critical.
- Graph Drawing is a serious subfield of visualisation in computer science.

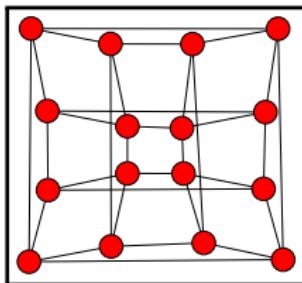
Spaghetti Monsters



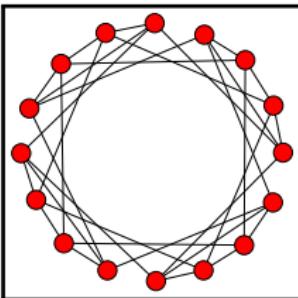
The router level connectivity of
the Internet 24rd November 2003
from opte.org/maps.

Isometry Quiz: menti.com, code 58 24 41 2

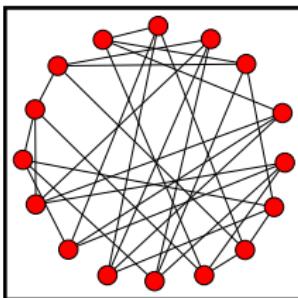
(updated 2/2/21) Which two are **not** identical to (A)? **PS** (Link to quiz)



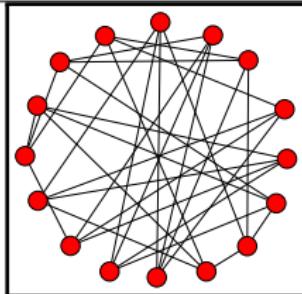
A



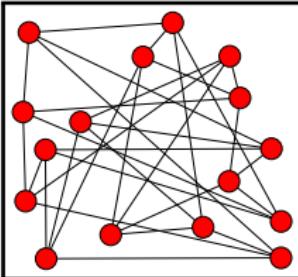
B



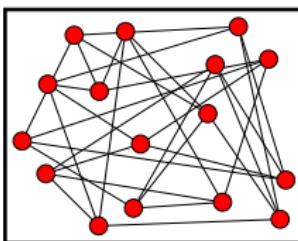
C



D



E



F

C & N Part II: End of Networks Lecture 3

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “Network Representation”

Intended Learning Outcomes

- Students will be able to choose an appropriate representation of a network for numerical and mathematical work.
- Students will be aware of the issues around network visualisation.

2. Basic Network Analysis

2.1 Degree and Degree Distribution

2.2 Making Connections in a Network

C & N Part II: Networks Lecture 4

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

Degree

Intended Learning Outcomes

- Students can define the DEGREE and STRENGTH of a vertex.
- Students can define the DEGREE DISTRIBUTION of a network.

PS= See problem sheet, **EFS**= Exercise For Students

2. Basic Network Analysis

2.1 Degree and Degree Distribution

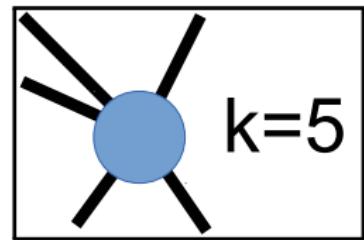
2.2 Making Connections in a Network

Degree and Degree Distribution

First key network measure

Degree of a Vertex

- An “ultra-local” property that has the least amount of information about the network.
- Yet the degree is the most intrinsic property of a vertex,
- The starting point for most analyses

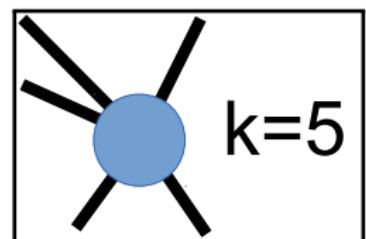


Degree in undirected graphs

Degree

The DEGREE k_i of a vertex i is the number of edges attached to that vertex.

For simple networks



$$k_i = \sum_j A_{ij} = \sum_j A_{ji} .$$

Note: If self-edges are present you may want to be careful about how you count them.

Degree in directed graphs

In and Out Degree

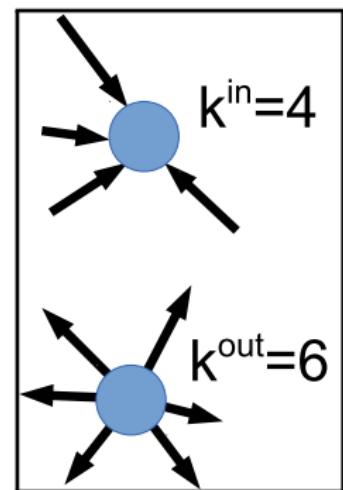
The IN-DEGREE $k_t^{(\text{in})}$ is the number of edges pointing into target vertex t .

$$k_t^{(\text{in})} = \sum_s A_{ts}$$

(sum over rows) The OUT-DEGREE $k_s^{(\text{out})}$ is the number of edges leaving source vertex s

$$k_s^{(\text{out})} = \sum_t A_{ts} \quad (\text{sum over columns})$$

The TOTAL DEGREE is $k_i^{(\text{total})} = k_i^{(\text{in})} + k_i^{(\text{out})}$.



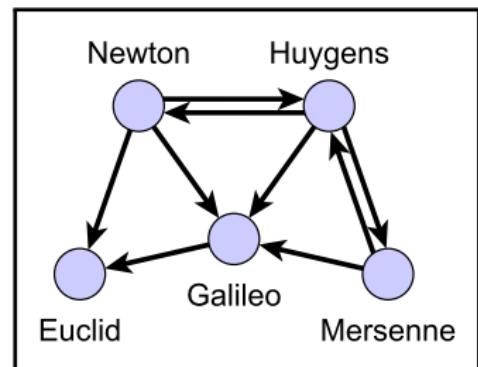
Index Order

WARNING: here we assume A_{ts} is edge from s to t .

Directed graph example: Web pages

Degree of a web page network

- A vertex i represents a web page
- A hyperlink on page i pointing to page j is directed edge (i, j) .
- Number of Web pages linked from your web page is your OUT-DEGREE $k_{\text{me}}^{(\text{out})}$
- Number of pages pointing to your page is your IN-DEGREE $k_{\text{me}}^{(\text{in})}$.
- The $k_{\text{me}}^{(\text{in})}$ is measure of the popularity of your webpage.



From MacTutor website

$$k_{\text{Galileo}}^{(\text{in})} = 3$$

$$k_{\text{Galileo}}^{(\text{out})} = 1$$

Degree Distribution $n(k)$

The Degree Distribution

The DEGREE DISTRIBUTION, $n(k)$, is the number of nodes with degree k .

As a probability distribution ...

$$p(k) = \frac{1}{N} n(k)$$

where

$$N = \sum_k n(k) = \sum_{i \in \mathcal{V}} 1$$

is the number of nodes in the network.

The value of $p(k)$ is the probability of finding a node with degree k when we choose a vertex at random.

Degree Distribution $n(k)$

The Degree Distribution

The DEGREE DISTRIBUTION, $n(k)$, is the number of nodes with degree k .

As a probability distribution

$$p(k) = \frac{1}{N} n(k)$$

where

$$N = \sum_k n(k) = \sum_{i \in \mathcal{V}} 1$$

is the number of nodes in the network.

The value of $p(k)$ is the probability of finding a node with degree k when we choose a vertex at random.

Average Degree $\langle k \rangle$

The Average Degree $\langle k \rangle$

The average number of edges attached to a vertex

$$\langle k \rangle = \frac{1}{N} \sum_{i \in \mathcal{V}} k_i = \sum_k p(k)k$$

for directed network (similar for directed).

Alternative way to specify number of edges E as

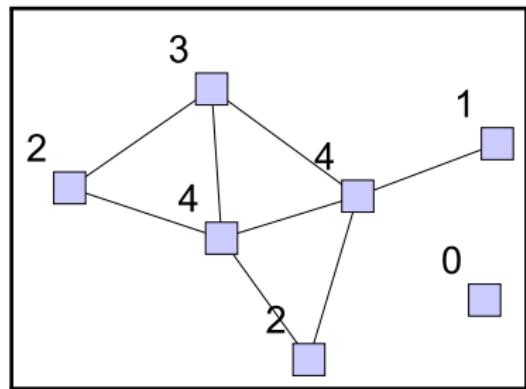
$$\langle k \rangle = \frac{2E}{N} \text{ if undirected}$$

Note: two ends or STUBS per edge available for degree counts

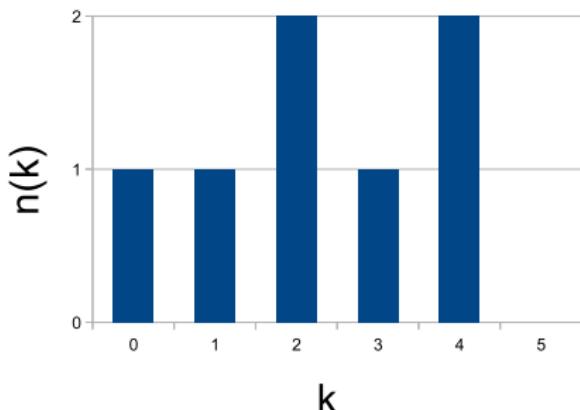
Degree Distribution example

The Degree Distribution

The DEGREE DISTRIBUTION, $n(k)$, is the number of nodes with degree k .



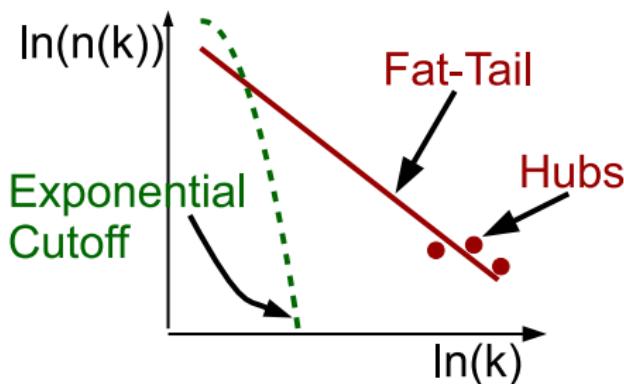
$$N = 7, E = 8$$
$$\Rightarrow \langle k \rangle = \frac{2 \times 8}{7} = 2\frac{2}{7}$$



Why is the degree distribution useful?

Big differences in degree distribution between

- traditional graphs in maths e.g. *Erdős-Reyní random graphs*
- physics networks e.g. *crystal lattices*
- general COMPLEX NETWORKS e.g. *social networks*



HUBS are vertices attached to a large fraction of the edges.

The rich get richer,
Pareto 80:20 rule, ...

Strength of a vertex

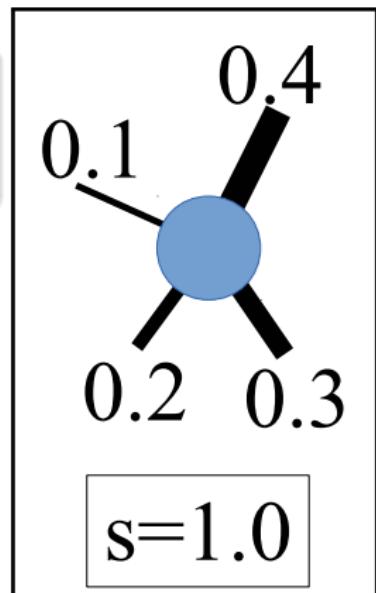
With weighted networks we can generalise the degree to take account of edge weight

The Strength of a vertex

The STRENGTH, s , is the total weight of edges attached to the vertex

For directed networks can count strength of in-coming edges, $s^{(\text{in})}$, and strength of outgoing edges $s^{(\text{out})}$. Can define probability distribution for strength.

$$s_i^{(\text{out})} = \sum_j A_{ji}, \quad s_j^{(\text{in})} = \sum_i A_{ji}.$$



C & N Part II: End of Networks Lecture 4

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “Degree”

Intended Learning Outcomes

- Students can define the DEGREE and STRENGTH of a vertex.
- Students can define the DEGREE DISTRIBUTION of a network.

C & N Part II: Networks Lecture 5

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

Making Connections

Intended Learning Outcomes

- Students can define different types of path in a network.
- Students can define a distance between two nodes in a network.
- Students can interpret the concept of “six degrees of separation” in terms of network distances.

PS= See problem sheet, **EFS**= Exercise For Students

2. Basic Network Analysis

2.1 Degree and Degree Distribution

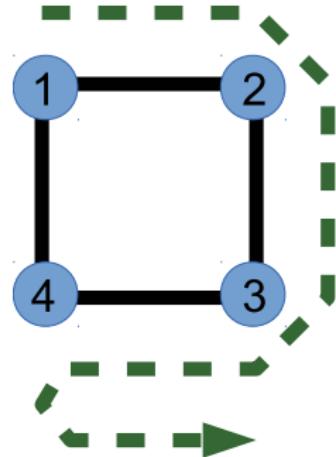
2.2 Making Connections in a Network

Making Connections in a Network

Second important type of network measure

Making Connections in a Network

- Path lengths are important when understanding how information flows around the system.
- Probe global properties of network, — often first such global measure tried.
- Can sometimes see emergent properties.



$$\mathcal{W} = \{1, 2, 3, 4, 3\}, \\ L = 4.$$

Walks on a graph

A Walk

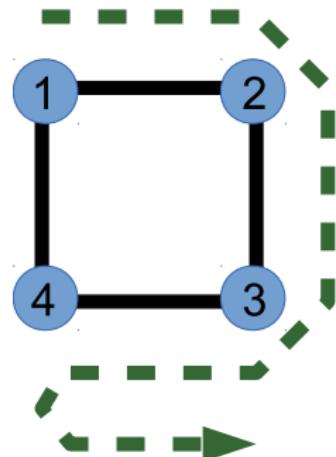
A WALK \mathcal{W} in a graph is a sequence of vertices, v_n with $n \in \{0, 1, \dots, L\}$, where there is an edge from each vertex v_n to the *next* vertex in the sequence, $(v_n, v_{n+1}) \in \mathcal{E}$.

The Length of a Walk

The LENGTH of a walk is L , the number of edges traversed in the walk (the number of vertices in the sequence minus one).

★ If $A_{ts} \in \{0, 1\}$, then number of walks ★

length L from s to t is $[A^L]_{ts}$ PS



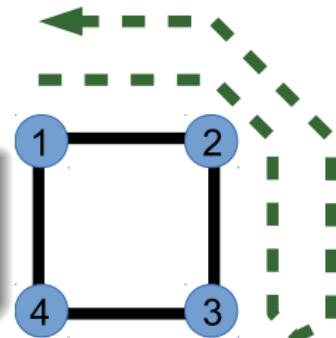
$$\mathcal{W} = \{1, 2, 3, 4, 3\}, \\ L = 4.$$

Cycles and Paths

As a vertex may appear more than once in walk there are some special types of walk:-

Cycles

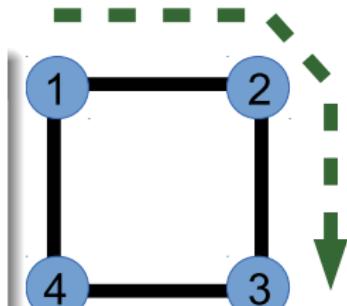
A CYCLE is a walk which starts and ends at the same vertex $v_0 = v_L$.



Paths

A PATH is a walk with *no* cycles, so each vertex in a path is unique

$$v_n = v_m \text{ iff } n = m.$$



Shortest Path

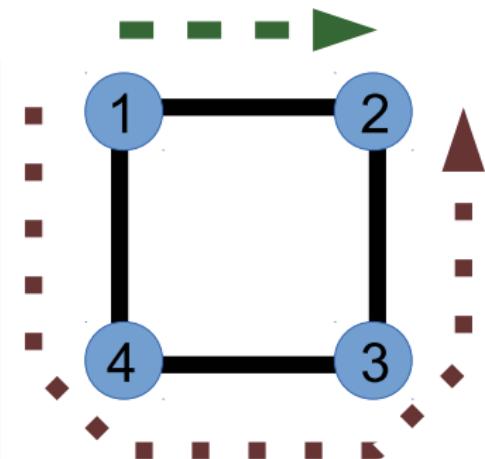
Shortest Paths

Let the length of the SHORTEST PATH from vertex i to j be ℓ_{ij} .

The AVERAGE SHORTEST PATH LENGTH ℓ is

$$\ell = \frac{1}{|\mathcal{P}|} \sum_{i,j \in \mathcal{P}} \ell_{ij}, \quad (2.1)$$

where \mathcal{P} is the set of all connected vertex pairs (i,j) .



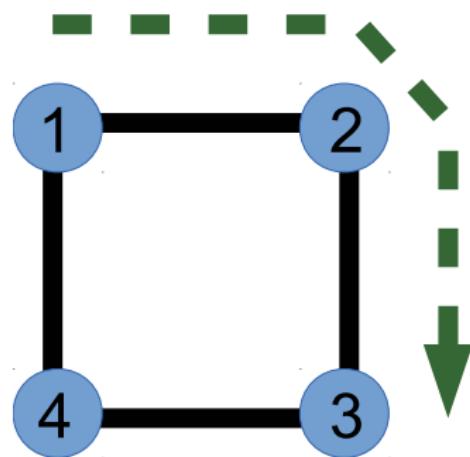
Diameter

Diameter

The DIAMETER D of a network is the largest of all shortest path lengths ℓ_{ij} between connected vertex pairs

$$D = \max\{\ell_{ij} | (i, j) \in \mathcal{P}\}$$

\mathcal{P} is the set of connected vertex pairs



$$D = 2$$

Why are path lengths useful?

- Path lengths are important when understanding how information flows around the system.
- Big differences between traditional physics networks (lattices) and other types of COMPLEX NETWORK

Typical length scales

- On a regular d -dimensional lattice — $\langle \ell \rangle, D \sim N^{1/d}$
- On a typical complex network — $\langle \ell \rangle, D \sim \ln(N)$
- Complex networks have much smaller \leftarrow SMALL WORLD distances than lattices
- Complex networks look like an infinite dimensional lattice **PS**

Why are path lengths useful?

- Path lengths are important when understanding how information flows around the system.
- Big differences between traditional physics networks (lattices) and other types of COMPLEX NETWORK

Typical length scales

- On a regular d -dimensional lattice — $\langle \ell \rangle, D \sim N^{1/d}$
- On a typical complex network — $\langle \ell \rangle, D \sim \ln(N)$
- Complex networks have much smaller distances than lattices \Leftarrow SMALL WORLD
- Complex networks look like an infinite dimensional lattice PS

Why are path lengths useful?

- Path lengths are important when understanding how information flows around the system.
- Big differences between traditional physics networks (lattices) and other types of COMPLEX NETWORK

Typical length scales

- On a regular d -dimensional lattice — $\langle \ell \rangle, D \sim N^{1/d}$
- On a typical complex network — $\langle \ell \rangle, D \sim \ln(N)$
- Complex networks have much smaller distances than lattices \Leftarrow **SMALL WORLD**
- Complex networks look like an infinite dimensional lattice **PS**

Why are path lengths useful?

- Path lengths are important when understanding how information flows around the system.
- Big differences between traditional physics networks (lattices) and other types of COMPLEX NETWORK

Typical length scales

- On a regular d -dimensional lattice — $\langle \ell \rangle, D \sim N^{1/d}$
- On a typical complex network — $\langle \ell \rangle, D \sim \ln(N)$
- Complex networks have much smaller distances than lattices \Leftarrow **SMALL WORLD**
- Complex networks look like an infinite dimensional lattice **PS**

Six Degrees of Separation

Many networks have surprisingly low average path lengths and diameters

Milgram's experiment (1967)

People in Omaha (Nebraska) and Wichita (Kansas) sent packets to people in Cambridge MA specified by name, profession and rough location only. Packets were swapped only between people who knew each other by first name.

Packets which arrived had been through about **five** intermediaries.



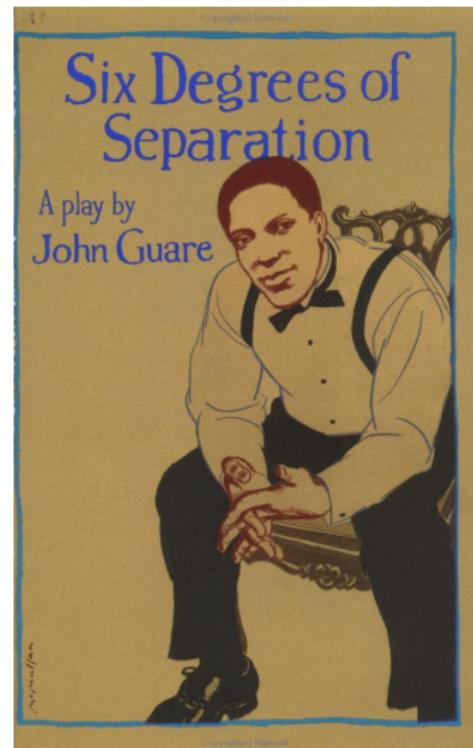
Six Degrees of Separation

Milgram's experiment inspired a play by John Guare (1938–)

I read somewhere that everybody on this planet is separated by only six other people. Six degrees of separation. Between us and everybody else on this planet. The president of the United States. A gondolier in Venice.

Film version:

[Six Degrees of Separation \(1993\)](#)

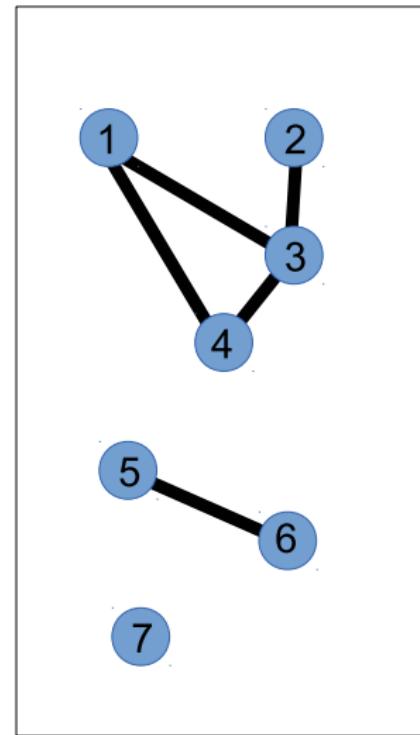


Components of Undirected Graphs

Components

A COMPONENT of an undirected graph is a subgraph in which there is a path between all vertices in the component and *no* path between the vertices of the component and the remaining vertices of the graph.

- Visually components are separate.
- Components PARTITION the set of vertices
i.e. every vertex is in one component and only one component.

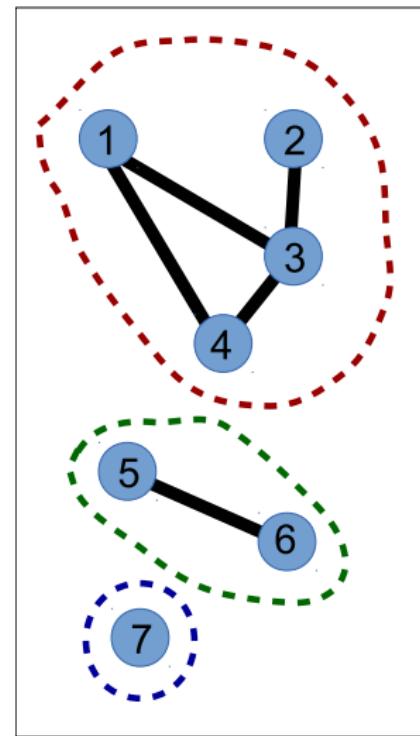


Components of Undirected Graphs

Components

A COMPONENT of an undirected graph is a subgraph in which there is a path between all vertices in the component and *no* path between the vertices of the component and the remaining vertices of the graph.

- Visually components are separate.
- Components PARTITION the set of vertices
i.e. every vertex is in one component and only one component.

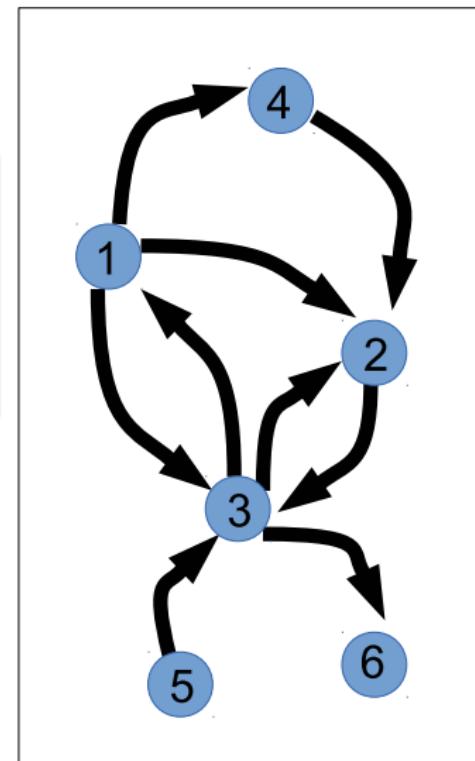


Weak Components of Directed Graphs

Key issue: Two vertices can be linked in just one direction

Weakly Connected Components

A WEAKLY CONNECTED COMPONENT of a directed graph are components of the equivalent undirected graph formed by ignoring the directions.

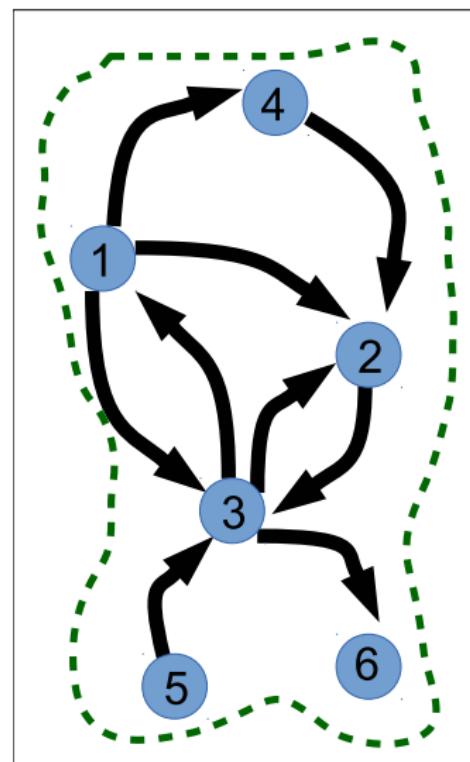


Weak Components of Directed Graphs

Key issue: Two vertices can be linked in just one direction

Weakly Connected Components

A WEAKLY CONNECTED COMPONENT of a directed graph are components of the equivalent undirected graph formed by ignoring the directions.

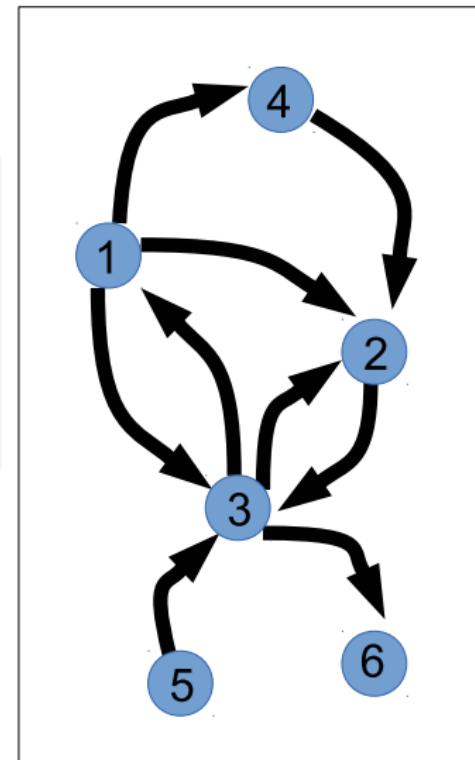


Strong Components of Directed Graphs

Key issue: Two vertices can be linked in just one direction

Strongly Connected Components

All vertices in a STRONGLY CONNECTED COMPONENT are connected by paths in **both** directions but there are never paths in both directions between a vertex in the component and one outside.

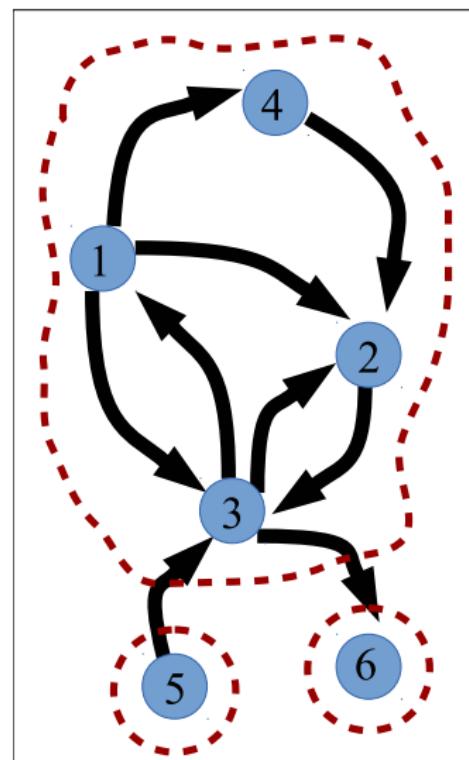


Strong Components of Directed Graphs

Key issue: Two vertices can be linked in just one direction

Strongly Connected Components

All vertices in a STRONGLY CONNECTED COMPONENT are connected by paths in **both** directions but there are never paths in both directions between a vertex in the component and one outside.



C & N Part II: End of Networks Lecture 5

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “Making Connections”

Intended Learning Outcomes

- Students can define different types of path in a network.
- Students can define a distance between two nodes in a network.
- Students can interpret the concept of “six degrees of separation” in terms of network distances.

3. Fat-Tailed Distributions

- 3.1 What is a Fat-Tailed Distribution?
- 3.2 Dealing with a Fat Tail in Practice
- 3.3 Origin of Fat Tails

C & N Part II: Networks Lecture 6

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

Fat-Tailed Distributions

Intended Learning Outcomes

- Students can define a fat-tailed distribution.
- Students can analyse a fat-tailed distribution
- Students can give some mathematical examples of fat-tailed distributions

PS= See problem sheet, **EFS**= Exercise For Students

Networks and Fat Tailed Distributions



3. Fat-Tailed Distributions

3.1 What is a Fat-Tailed Distribution?

3.2 Dealing with a Fat Tail in Practice

3.3 Origin of Fat Tails

Networks and Fat Tailed Distributions

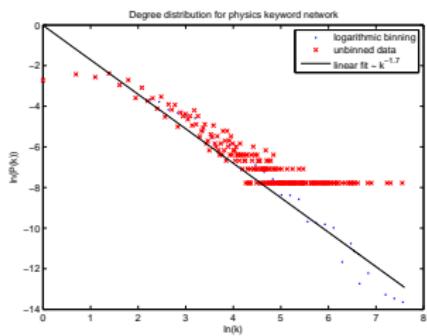


★ We will focus on the degree distribution of ★
networks, $n(k) = Np(k)$.

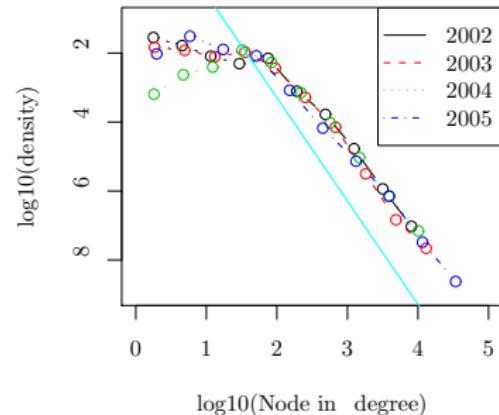
- Many of the comments apply to distributions of other network measures
- Many of the comments apply to distributions in other contexts e.g. wealth distributions in economics, word frequency, sizes of earthquakes

Examples of fat tails

All $\ln(p(k))$ vs. $\ln(k)$ plots



Degree distribution in network of words from titles of Imperial Physics publications, linked if appear in same title (unpublished)



Distribution of the in-degree of photos downloaded from a web site where photos connected if downloaded consecutively (Argent-Katwala et al, 2007).

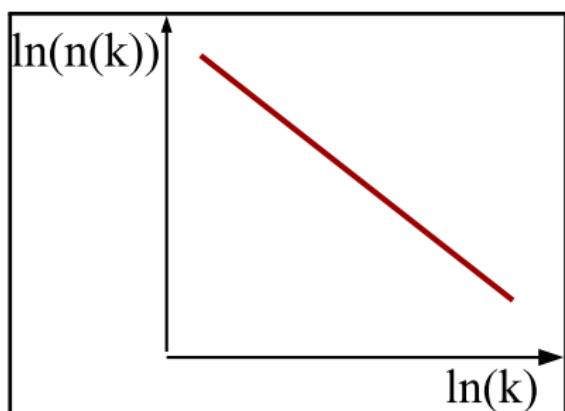
Loose Definition of a Fat Tail

Loose Definition of a Fat Tail

A fat-tailed distribution $p(k)$ is one where there is a “significant” probability of finding “large” values k .

How do we define what is

- “significant”?
- “large”?



Answer: we need to make a comparison

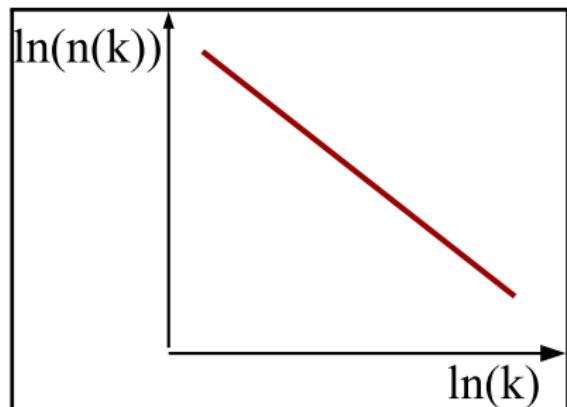
Loose Definition of a Fat Tail

Loose Definition of a Fat Tail

A fat-tailed distribution $p(k)$ is one where there is a “significant” probability of finding “large” values k .

How do we define what is

- “significant” ?
- “large” ?



Answer: we need to make a comparison

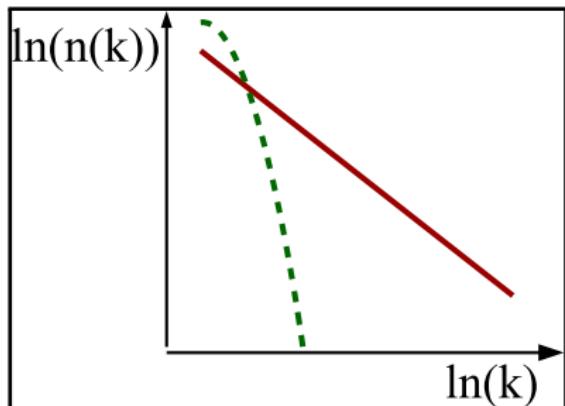
Precise Definition of a Fat Tail

Precise Definition of a Fat Tail

A fat-tailed distribution $p(k)$ is one where the probability of finding large measurements is power law

$$\lim_{k \rightarrow \infty} \frac{p(k)}{\exp\{-k/k_0\}} > O(1)$$

i.e. $p(k)$ decays **slower** than an exponential.



- Only useful mathematically as k always finite in practice.

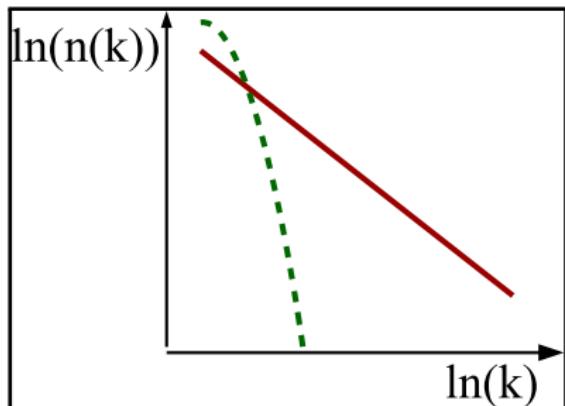
Precise Definition of a Fat Tail

Precise Definition of a Fat Tail

A fat-tailed distribution $p(k)$ is one where the probability of finding large measurements is power law

$$\lim_{k \rightarrow \infty} \frac{p(k)}{\exp\{-k/k_0\}} > O(1)$$

i.e. $p(k)$ decays **slower** than an exponential.



- Only useful mathematically as k always finite in practice.

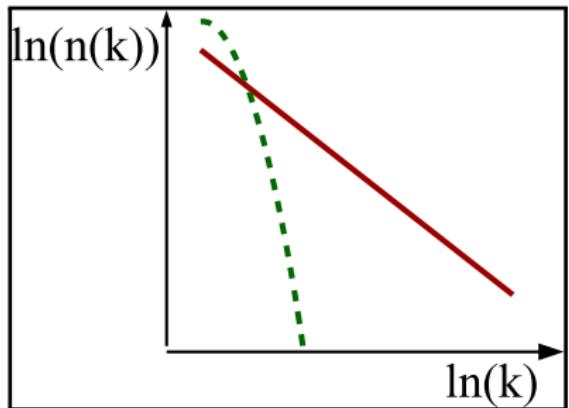
Practical Definition of a Fat Tail

Practical Definition of a Fat Tail

A fat-tailed distribution $p(k)$ is one where on a **log-log plot** we see a **roughly** linear fall off

i.e. power law $p(k) \propto k^{-\gamma}$

$$\ln(p(k)) \sim -\gamma \ln(k) + c$$



3. Fat-Tailed Distributions

3.1 What is a Fat-Tailed Distribution?

3.2 Dealing with a Fat Tail in Practice

3.3 Origin of Fat Tails

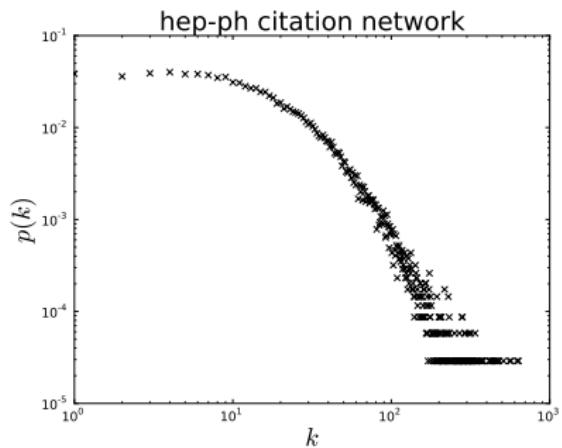
Issues with Fat Tails in Practice

Fat tails mean

- few vertices (data points) with large degrees (values)
- many large values k have no examples, $n(k) = 0$

BUT large values are crucial for a precise understanding of the shape
e.g. KC recommends 3 orders of magnitude for power laws

→ How can we improve the situation?

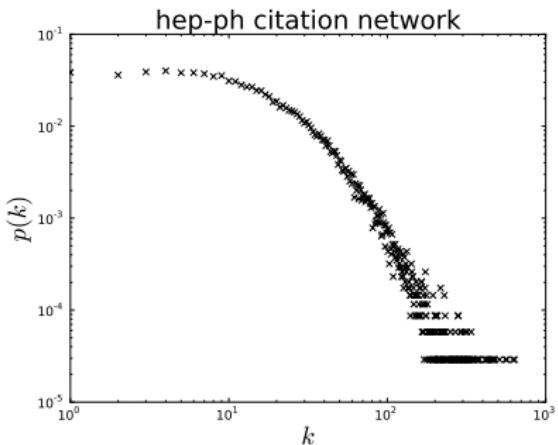


hep-ph data from KDD cup,
35,000 papers, $\langle k \rangle = 12.2$

Three Approaches to a Fat Tailed Distribution

Three ways to deal with few data points for large degree (values):-

- ① Log Binning
- ② CDF = Cumulative Distribution Function
- ③ Zipf plot, rank vs. degree



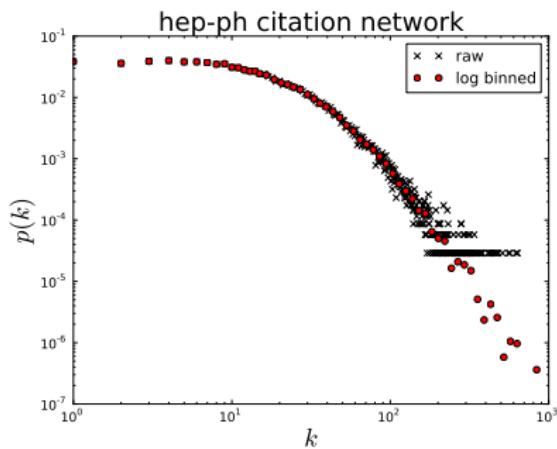
Log Binning

- Define i -th bin to range from degree b_i to b_{i+1} where

$$\frac{b_{i+1}}{b_i} = \exp\{\Delta\}$$

where $\Delta > 0$.

- Count number of vertices with degree in each bin.
- Plot $\text{density} = (\text{number in bins}) / (\text{bin width})$ at mid point of bin .
- On $\log(p(k))$ vs $\log(k)$ values are equally spaced at intervals of Δ



$(\exp\{\Delta\} = 1.1, \Delta = 0.095)$
Choose Δ to ensure no bin is empty.

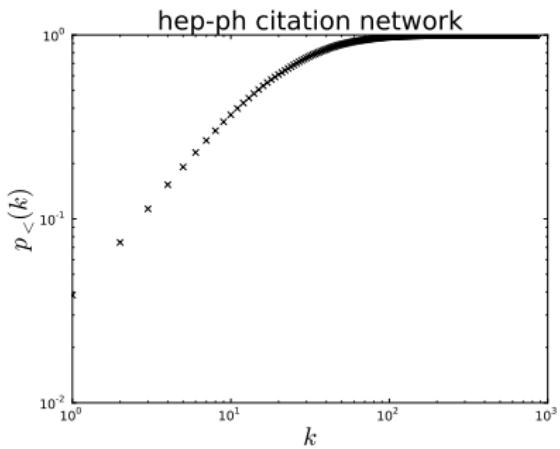
CDF (Cumulative Distribution Function)

The CDF (Cumulative Distribution Function) $p_<(k)$ is defined as

$$p_<(k) = \sum_{k'=0}^k p(k')$$

This is much smoother than $p(k)$ but is closely related in shape

e.g. CDF is fat tailed if
 $p(k)$ is fat tailed



Zipf

Zipf plots are named after George Kingsley ZIPF (1902–1950) who was an American linguist interested in the statistical properties of languages. Through that work he also looked at fat-tailed distributions in other contexts.



From Wikimedia.

Zipf Plots (1935, 1949)

Zipf Plot

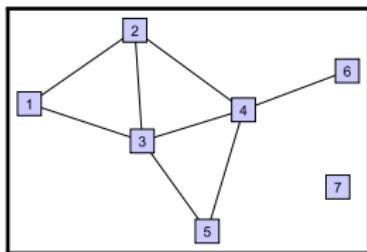
A ZIPF PLOT is a plot of the degree, k_i , of each node i against the RANK r_i of that degree value.

If node i has rank r then

- it has the r -th largest value.
- there are $(r - 1)$ nodes with larger degree (or same degree k_i).

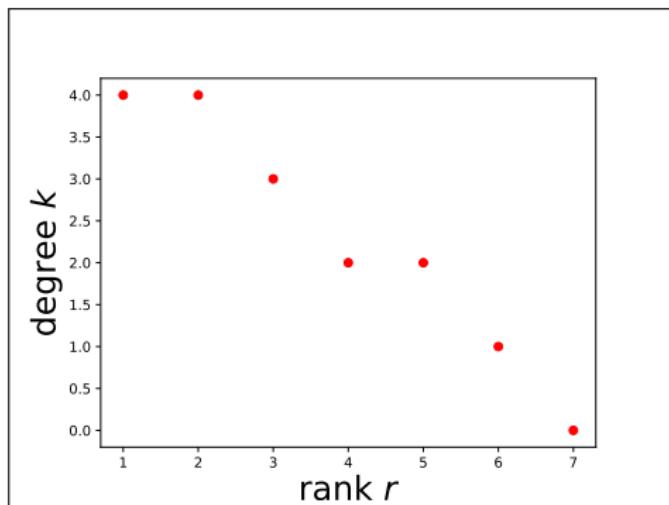
Note every node must have a different rank. If two nodes have the same degree you MUST give them different ranks

Zipf Plot: Simple Example



index	degree	rank
4	4	1
3	4	2
2	3	3
1	2	4
5	2	5
6	1	6
7	0	7

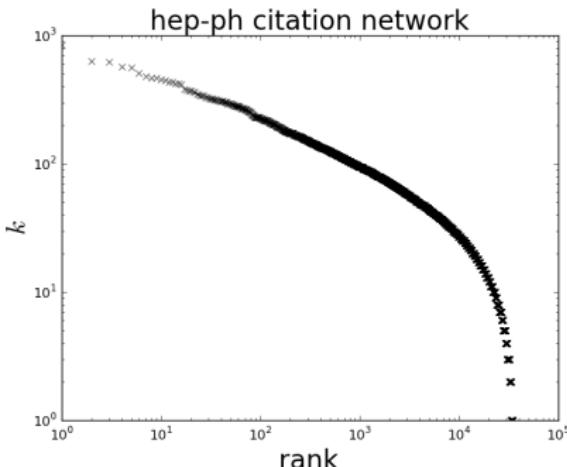
A ZIPF PLOT is a plot of the degree, k_i , of each node i against the RANK r_i of that degree value.



Zipf Plots (1935, 1949)

A ZIPF PLOT is a plot of degree k_r vs. RANK r — r -th largest value.
Fat-tailed degree distribution implies fat-tailed Zipf plot

$$n(k) = \frac{A}{k^\gamma} \quad \Rightarrow \quad \text{rank } r = \frac{A}{(\gamma - 1)} \frac{1}{(k_r)^{\gamma-1}} \quad (3.1)$$



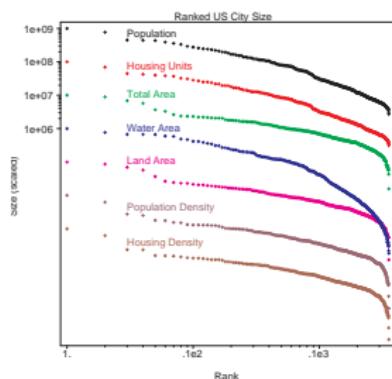
$$\text{where } r = \sum_{k'=k_r}^{\infty} n(k). \quad (3.2)$$

Only suitable for a
single data set

Zipf Plots (1935, 1949)

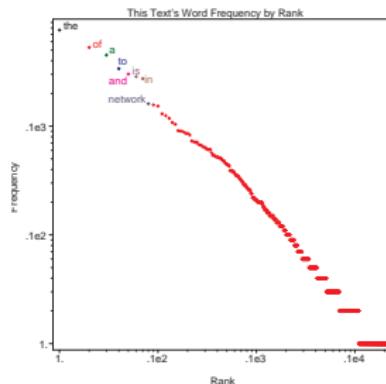
A ZIPF PLOT is a plot of degree k_r vs. RANK r — r -th largest value.
 Fat-tailed degree distribution implies fat-tailed Zipf plot

$$n(k) = \frac{A}{k^\gamma} \quad \Rightarrow \quad \text{rank } r = \frac{A}{(\gamma - 1)} \frac{1}{(k_r)^{\gamma-1}} \quad (3.1)$$



US City size by Rank

(Fig 10, Evans 2004)



Word frequency in network review (Fig 10, Evans 2004)

3. Fat-Tailed Distributions

3.1 What is a Fat-Tailed Distribution?

3.2 Dealing with a Fat Tail in Practice

3.3 Origin of Fat Tails

Classic Distributions without Fat Tails

k is random variable

- Binomial distribution
(k discrete, finite n)

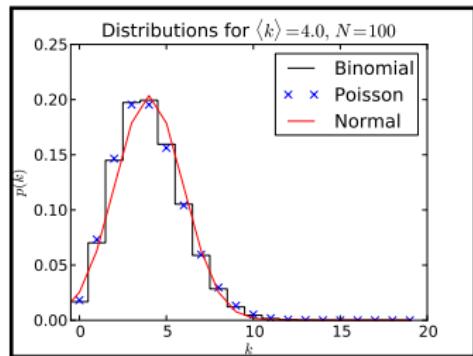
$$B(k, p, n) = \frac{n!}{(n - k)!k!} p^k (1 - p)^{n - k}$$

- Poisson Distribution (k discrete)

$$P(k, \lambda) = \frac{\lambda^{-k} e^{-\lambda}}{k!}$$

- Normal or Gaussian Distribution
(k continuous)

$$P(k, \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left\{ -\frac{(k - \mu)^2}{2\sigma^2} \right\}$$

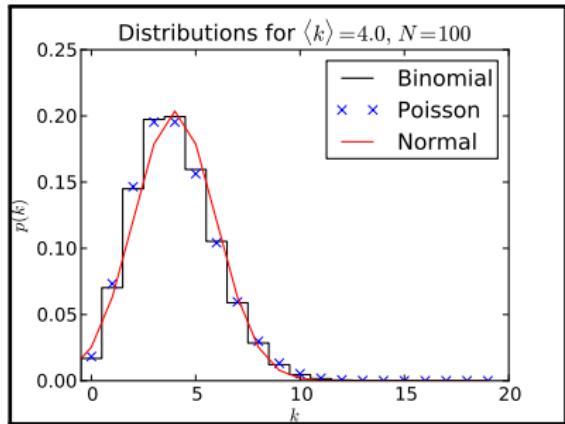


Relationships between Classic Distributions

- Binomial \rightarrow Poisson as $n \rightarrow \infty$ with $\lambda = pn$ finite.
- Poisson \approx Normal with

$$\mu = \lambda = np \gg 1,$$

$$\sigma^2 = \lambda = np(1 - p),$$
 as negative k unlikely in Normal distribution, and discrete k looks like continuous variable.



Central Limit Theorem

Binomial/Poisson/Normal distributions common because we often **ADD** many fluctuating values together, and the sizes of each value is fairly independent of the others e.g.

- Calculating the average height of humans
- Average number of coauthors on maths papers

Central Limit Theorem

The difference between the average of n measurements of some random variable k and a normal distribution tends to zero as $n \rightarrow \infty$ for *any* $p(k)$ provided $\langle k \rangle$ and $\langle k^2 \rangle$ are *finite*.

“data which are influenced by many small and unrelated random effects are approximately normally distributed” (MathWorld)

Multiplicative Processes

Suppose we **MULTIPLY** random positive numbers $y_i > 0$ ($i = 1, 2, \dots, n$) created independently with probability $p(y)$, and so consider the geometric mean

$$\tilde{Y} = Y^{1/n}, \quad Y = \prod_{i=1}^n y_i$$

What is the distribution of Y or the geometric mean \tilde{Y} ?

Multiplicative Processes

What is the distribution of the product \tilde{Y} of random variables y_i ; or the geometric mean Y ?

$$\tilde{Y} = \prod_{i=1}^n y_i, \quad Y = \tilde{Y}^{1/n}$$

Answer:

$$\text{Consider } X = \ln(Y) = \frac{1}{n} \sum_{i=1}^n \ln(y_i)$$

As long as the average and variance of $x_i = \ln(y_i)$ is finite then by central limit theorem we find the sum or average of $x_i = \ln(k_i)$ is always normally distributed

$$p(X) \approx \frac{1}{s\sqrt{2\pi}} \exp \left\{ -\frac{(X - m)^2}{2s^2} \right\}$$

Multiplicative Processes

What is the distribution of the product \tilde{Y} of random variables y_i , or the geometric mean Y ?

$$\tilde{Y} = \prod_{i=1}^n y_i, \quad Y = \tilde{Y}^{1/n}$$

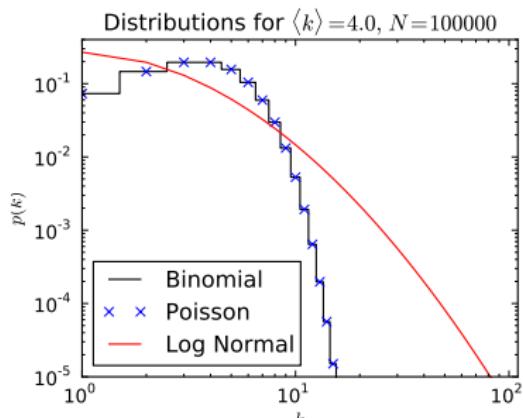
Answer: This means that the distribution of the geometric mean is a LOG-NORMAL DISTRIBUTION

$$p(Y) \approx \frac{1}{Ys\sqrt{2\pi}} \exp \left\{ -\frac{(\ln(Y) - m)^2}{2s^2} \right\}$$

CARE: The log normal parameters m and s are *not* the mean and standard deviation of Y but the mean and standard deviation of $X = \ln(Y)$, as $m = \langle \ln(y_i) \rangle$, $s = \langle (\ln(y_i) - m)^2 \rangle$.

Log Normal as a fat tail

We **MULTIPLY** random positive numbers $y_i > 0$ ($i = 1, 2, \dots, n$) and consider the geometric mean $Y = (\prod_i y_i)^{1/n}$ and this will be distributed as a log-normal



$$p(Y) \approx \frac{1}{sY\sqrt{2\pi}} \exp \left\{ -\frac{(\ln(Y) - m)^2}{2s^2} \right\}$$

This is a **fat-tailed distribution** in Y

e.g. compare against plot of random variable with same mean

$$\langle Y \rangle = \exp\{\mu + s^2/2\}$$

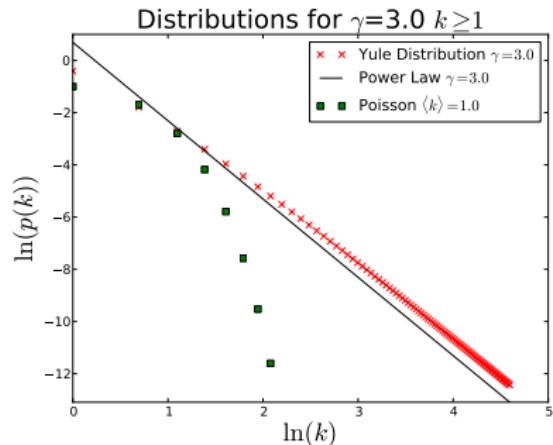
Yule/Power Law as a Fat Tail

- Simplest continuous distribution decaying slower than exponential is

$$p(k) = \frac{A}{k^\gamma} \quad k \in \mathbb{R} \text{ (reals)}$$

- The discrete equivalent (integer k) is the YULE-(SIMON) DISTRIBUTION (1925,1955)

$$p(k) = \frac{(r+1)!(k-1)!}{(k+r)!} \rightarrow \frac{1}{k^{r+1}},$$



★ What process gives ★
a Yule/power law
distribution?

Generic Fat Tails: Summary

We have

- defined fat-tailed degree distributions,
- seen how to organise fat-tailed data in useful ways e.g. for plotting,
- shown that multiplicative processes lead to a log-normal distribution as an example of a process leading to a fat-tailed distribution,
- defined the power law and Yule distributions as the standard examples of fat-tailed distributions for continuous and discrete variables respectively.



C & N Part II: End of Networks Lecture 6

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “Fat-Tailed Distributions”

Intended Learning Outcomes

- Students can define a fat-tailed distribution.
- Students can analyse a fat-tailed distribution
- Students can give some mathematical examples of fat-tailed distributions

4. Network Models

- 4.1 Why Model a Network?
- 4.2 Erdős-Rényi Random graph model
- 4.3 Configuration Model
- 4.4 Price Model
- 4.5 (#) Generating Functions

C & N Part II: Networks Lecture 7

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

The Simplest Network Model

Intended Learning Outcomes

- Students can explain why a network model is useful.
- Students can define the Erdős-Rényi (ER) random graph.
- Students know the key property of the ER random graph.

PS= See problem sheet, **EFS**= Exercise For Students

4. Network Models

4.1 Why Model a Network?

4.2 Erdős-Rényi Random graph model

4.3 Configuration Model

4.4 Price Model

4.5 (#) Generating Functions

Why Model a Network?

A simple model includes one or two key processes.

— typically very few model parameters

If model works

⇒ processes in model may be important for processes in real world

If model fails

⇒ model's processes do not explain major features in real world

e.g. Can we understand why so many networks have fat-tailed degree distributions?

Why Model a Network?

A simple model includes one or two key processes.

— typically very few model parameters

If model works

⇒ processes in model may be important for processes in real world

If model fails

⇒ model's processes do not explain major features in real world

e.g. Can we understand why so many networks have fat-tailed degree distributions?

Why Model a Network?

A simple model includes one or two key processes.

— typically very few model parameters

If model works

⇒ processes in model may be important for processes in real world

If model fails

⇒ model's processes do not explain major features in real world

e.g. Can we understand why so many networks have fat-tailed degree distributions?

Why Model a Network?

A simple model includes one or two key processes.

— typically very few model parameters

If model works

⇒ processes in model may be important for processes in real world

If model fails

⇒ model's processes do not explain major features in real world

e.g. Can we understand why so many networks have fat-tailed degree distributions?

Null Models

- Suppose we have a network created from a data set or a theoretical model.
- We then make some measurements:
e.g. degree distribution, number of components or whatever.
- How do we assess the values?
What is a “big” degree? What is a “small” average path length?

⇒ Make a comparison with a NULL MODEL

Null Model

- A model with same basic properties as the data of interest
- A baseline for comparison

e.g. for networks choose null model with same N and E

Null Models

- Suppose we have a network created from a data set or a theoretical model.
- We then make some measurements:
e.g. degree distribution, number of components or whatever.
- How do we assess the values?
What is a “big” degree? What is a “small” average path length?

⇒ Make a comparison with a NULL MODEL

Null Model

- A model with same basic properties as the data of interest
- A baseline for comparison

e.g. for networks choose null model with same N and E

4. Network Models

4.1 Why Model a Network?

4.2 Erdős-Rényi Random graph model

4.3 Configuration Model

4.4 Price Model

4.5 (#) Generating Functions

Erdős-Rényi Random graph model

The simplest null model.

Background of Erdős-Rényi Random Graph

- Paul Erdős (double acute accent) was a famous Hungarian itinerant mathematician who produced a prodigious number of papers with a vast number of different coauthors.
- Mathematicians calculate their Erdős number. The shortest path length from mathematician to Erdős via coauthors on papers, e.g. my Erdős number is at most 5.
- Around 1957 Erdős and Rényi produced results on the simplest model of a COMPLEX NETWORK.
- Model was studied earlier by Solomonoff and Rapoport (1951).



ER (Erdős-Rényi) Random Graphs

Two distinct ER ensembles of simple networks:

- A - $\mathcal{G}(N, p)$** N vertices in which each pair of vertices is connected with probability p , independent of all other pairs,
- B - $\mathcal{G}(N, E)$** N vertices with E edges connected chosen at random with uniform probability from $N(N - 1)/2$ pairs.

- Stochastic definitions \Rightarrow sets of graphs $\mathcal{G} = \{G_1, G_2, \dots\}$.
- Definition **A**, $\mathcal{G}(N, p)$, best mathematically.
- Definition **B**, $\mathcal{G}(N, E)$, best numerically.
- Definitions equivalent in the LARGE GRAPH LIMIT $N \rightarrow \infty$
c.f. canonical and microcanonical ensembles. **PS**
- Most calculations done in LARGE SPARSE GRAPH LIMIT:

$$N \rightarrow \infty \text{ and } \langle k \rangle \sim O(1).$$

★ Here we use definition **A**, $\mathcal{G}(N, p)$ ★

ER (Erdős-Rényi) Random Graphs

Two distinct ER ensembles of simple networks:

- A - $\mathcal{G}(N, p)$** N vertices in which each pair of vertices is connected with probability p , independent of all other pairs,
- B - $\mathcal{G}(N, E)$** N vertices with E edges connected chosen at random with uniform probability from $N(N - 1)/2$ pairs.

- Stochastic definitions \Rightarrow sets of graphs $\mathcal{G} = \{G_1, G_2, \dots\}$.
- Definition **A**, $\mathcal{G}(N, p)$, best mathematically.
- Definition **B**, $\mathcal{G}(N, E)$, best numerically.
- Definitions equivalent in the LARGE GRAPH LIMIT $N \rightarrow \infty$
c.f. canonical and microcanonical ensembles. **PS**
- Most calculations done in LARGE SPARSE GRAPH LIMIT:

$$N \rightarrow \infty \text{ and } \langle k \rangle \sim O(1).$$

★ Here we use definition **A**, $\mathcal{G}(N, p)$ ★

ER Random graph degree distribution

Using definition **A** $\mathcal{G}(N, p)$ — N vertices fixed, edges probability p

The degree distribution

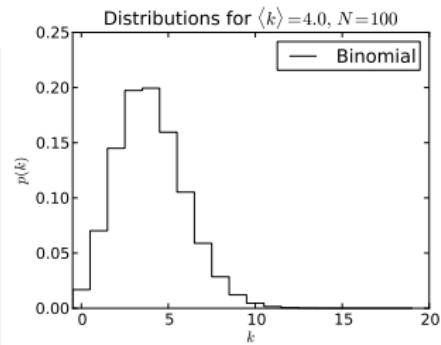
The degree distribution \tilde{p} is a binomial distribution

$$\tilde{p}(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}$$

where we have $(N-1)$ as the model does not allow self-loops.

$$\langle k \rangle = p(N-1), \sigma_k^2 = \langle k \rangle(1-p).$$

N.B. $\binom{N-1}{k} = \frac{(N-1)!}{k!(N-1-k)!}$



ER Random graph degree distribution

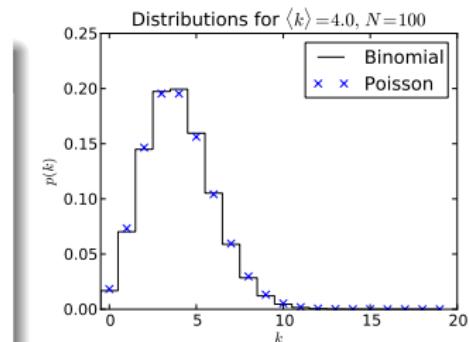
In the large sparse graph limit $N \rightarrow \infty$ and $\langle k \rangle \sim O(1)$ so $p \rightarrow 0$

The degree distribution

The degree distribution \tilde{p} is approximately a **Poisson distribution**

$$\tilde{p}(k) = \frac{(\langle k \rangle)^k e^{-\langle k \rangle}}{k!}$$

Variance $\sigma_k^2 = \langle k \rangle = pN$.



ER Random graph ensemble averages

If $Q(g)$ is value of quantity Q measured on graph g then average is

$$\langle Q \rangle = \frac{1}{N_A} \sum_{g \in \mathcal{G}_A} Q(g) \quad \text{where} \quad N_A = \sum_{g \in \mathcal{G}_A} 1 = \text{No. Graphs}$$

e.g. Average Degree

$$\langle k \rangle = \frac{1}{N_A} \sum_{g \in \mathcal{G}_A} \left(\frac{1}{N} \sum_{i \in \mathcal{V}} k_i \right) = \sum_k \tilde{p}(k)k$$

$$\Rightarrow \boxed{\langle k \rangle = p(N - 1)}$$

$$\boxed{E = \frac{\langle k \rangle N}{2} = \frac{pN(N - 1)}{2}}$$

★ Use this to choose p if want specific N and E in model. ★

ER Random graph ensemble averages

If $Q(g)$ is value of quantity Q measured on graph g then average is

$$\langle Q \rangle = \frac{1}{N_A} \sum_{g \in \mathcal{G}_A} Q(g) \quad \text{where} \quad N_A = \sum_{g \in \mathcal{G}_A} 1 = \text{No. Graphs}$$

e.g. Average Degree

$$\langle k \rangle = \frac{1}{N_A} \sum_{g \in \mathcal{G}_A} \left(\frac{1}{N} \sum_{i \in \mathcal{V}} k_i \right) = \sum_k \tilde{p}(k)k$$

$$\Rightarrow \boxed{\langle k \rangle = p(N - 1)}$$

$$\boxed{E = \frac{\langle k \rangle N}{2} = \frac{pN(N - 1)}{2}}$$

★ Use this to choose p if want specific N and E in model. ★

ER Random graph ensemble averages

If $Q(g)$ is value of quantity Q measured on graph g then average is

$$\langle Q \rangle = \frac{1}{N_A} \sum_{g \in \mathcal{G}_A} Q(g) \quad \text{where} \quad N_A = \sum_{g \in \mathcal{G}_A} 1 = \text{No. Graphs}$$

e.g. Average Degree

$$\langle k \rangle = \frac{1}{N_A} \sum_{g \in \mathcal{G}_A} \left(\frac{1}{N} \sum_{i \in \mathcal{V}} k_i \right) = \sum_k \tilde{p}(k)k$$

$$\Rightarrow \boxed{\langle k \rangle = p(N - 1)}$$

$$\boxed{E = \frac{\langle k \rangle N}{2} = \frac{pN(N - 1)}{2}}$$

★ Use this to choose p if want specific N and E in model. ★

C & N Part II: End of Networks Lecture 7

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “The Simplest Network Model”

Intended Learning Outcomes

- Students can explain why a network model is useful.
- Students can define the Erdős-Rényi (ER) random graph.
- Students know the key property of the ER random graph.

C & N Part II: Networks Lecture 8

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

The Configuration Model

Intended Learning Outcomes

- Students can define the Configuration Model.
- Students know the key property of this model.

PS= See problem sheet, **EFS**= Exercise For Students

4. Network Models

4.1 Why Model a Network?

4.2 Erdős-Rényi Random graph model

4.3 Configuration Model

4.4 Price Model

4.5 (#) Generating Functions

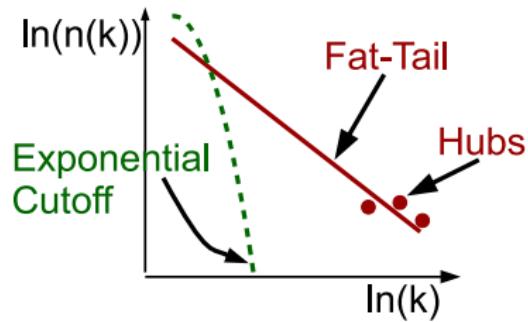
Configuration Model

A general random graph model.

Fat Tails and Null Models

- FAT-TAILED DEGREE DISTRIBUTIONS are common
e.g. many HUBS — vertices with large numbers of neighbours
- Erdős-Rényi random graphs with binomial/Poisson degree distribution have exponential cutoff, not fat-tailed,
- Need better NULL MODEL

⇒ Configuration Model



Definition of Configuration Model

Random graphs where the degree of each vertex $\{\tilde{k}_i\}$ is given.

Two Ensembles

A - $\mathcal{G}(N, p(k))$ N vertices in which each pair of vertices (i, j) is connected with probability $p(i, j) = \tilde{k}_i \tilde{k}_j / (2\tilde{E})^2$ with $2\tilde{E} = \sum_i \tilde{k}_i$, independent of all other pairs,

B - $\mathcal{G}(N, \{k_i\})$ N vertices, vertex i has degree $k_i = \tilde{k}_i$ exactly.

- Stochastic definitions \Rightarrow sets of graphs $\mathcal{G} = \{G_1, G_2, \dots\}$.
- Definition **A**, $\mathcal{G}(N, p(k))$, best mathematically.
- Definition **B**, $\mathcal{G}(N, \{k_i\})$, best numerically.
- Definitions equivalent in the large graph limit $N \rightarrow \infty$ c.f. canonical and microcanonical ensembles.
- If degree distribution is Binomial then get ER random graph

Definition of Configuration Model

Random graphs where the degree of each vertex $\{\tilde{k}_i\}$ is given.

Two Ensembles

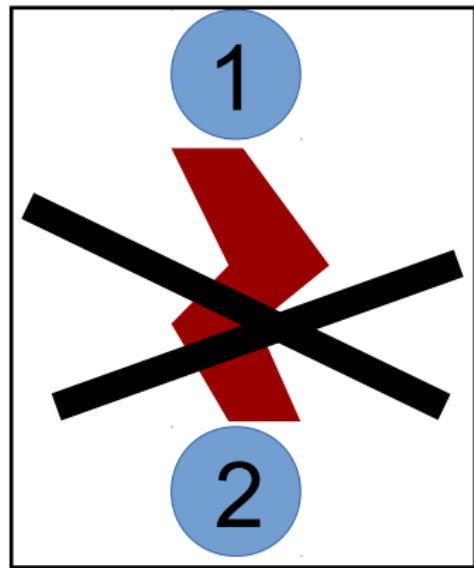
A - $\mathcal{G}(N, p(k))$ N vertices in which each pair of vertices (i, j) is connected with probability $p(i, j) = \tilde{k}_i \tilde{k}_j / (2\tilde{E})^2$ with $2\tilde{E} = \sum_i \tilde{k}_i$, independent of all other pairs,

B - $\mathcal{G}(N, \{k_i\})$ N vertices, vertex i has degree $k_i = \tilde{k}_i$ exactly.

- Stochastic definitions \Rightarrow sets of graphs $\mathcal{G} = \{G_1, G_2, \dots\}$.
- Definition **A**, $\mathcal{G}(N, p(k))$, best mathematically.
- Definition **B**, $\mathcal{G}(N, \{k_i\})$, best numerically.
- Definitions equivalent in the large graph limit $N \rightarrow \infty$ c.f. canonical and microcanonical ensembles.
- If degree distribution is Binomial then get ER random graph

Vertex Correlations

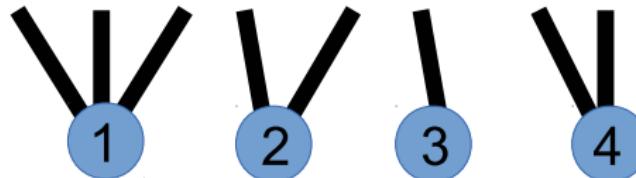
These random graphs by definition have **no statistical correlations** between neighbouring vertices.



Numerical Implementation: From $\{k_i\}$

Definition **B**, $\mathcal{G}(N, \{k_i\})$, with list of $\{k_i\}$ given.

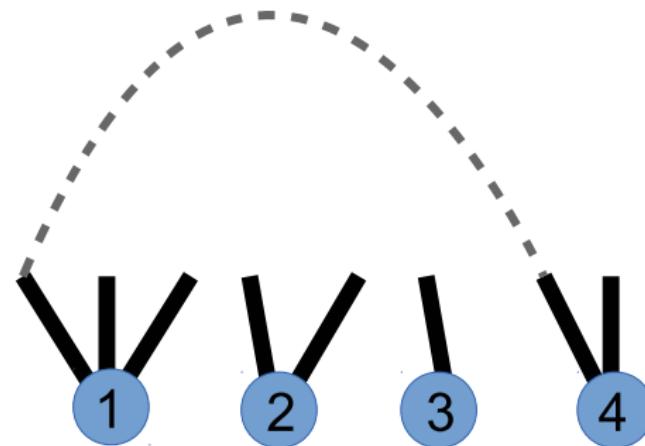
- ① Assign k_i stubs (half an edge) to each vertex i .
- ② Choose two stubs at random and join them into an edge.
- ③ Repeat until all stubs joined.



Numerical Implementation: From $\{k_i\}$

Definition **B**, $\mathcal{G}(N, \{k_i\})$, with list of $\{k_i\}$ given.

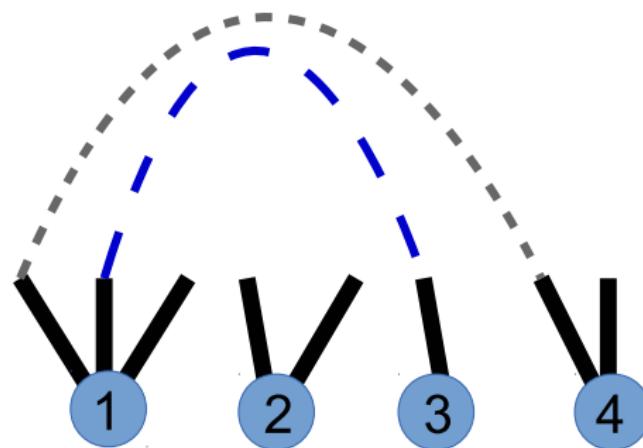
- ① Assign k_i stubs (half an edge) to each vertex i .
- ② Choose two stubs at random and join them into an edge.
- ③ Repeat until all stubs joined.



Numerical Implementation: From $\{k_i\}$

Definition **B**, $\mathcal{G}(N, \{k_i\})$, with list of $\{k_i\}$ given.

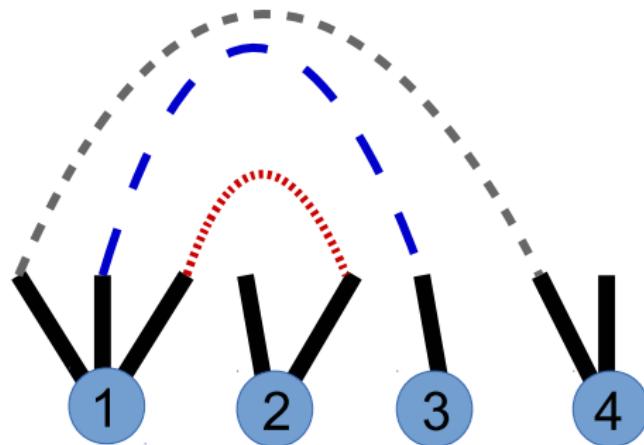
- ① Assign k_i stubs (half an edge) to each vertex i .
- ② Choose two stubs at random and join them into an edge.
- ③ Repeat until all stubs joined.



Numerical Implementation: From $\{k_i\}$

Definition **B**, $\mathcal{G}(N, \{k_i\})$, with list of $\{k_i\}$ given.

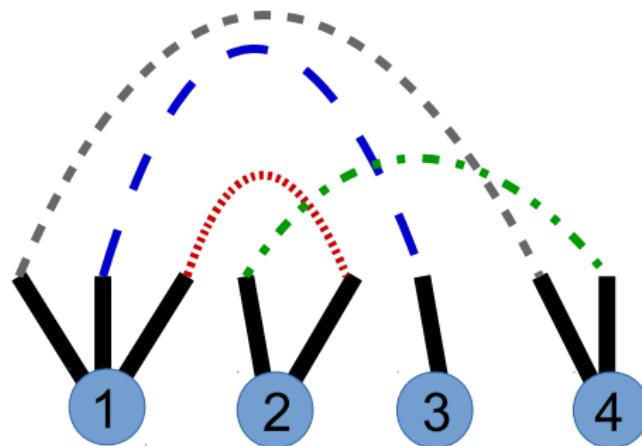
- ① Assign k_i stubs (half an edge) to each vertex i .
- ② Choose two stubs at random and join them into an edge.
- ③ Repeat until all stubs joined.



Numerical Implementation: From $\{k_i\}$

Definition **B**, $\mathcal{G}(N, \{k_i\})$, with list of $\{k_i\}$ given.

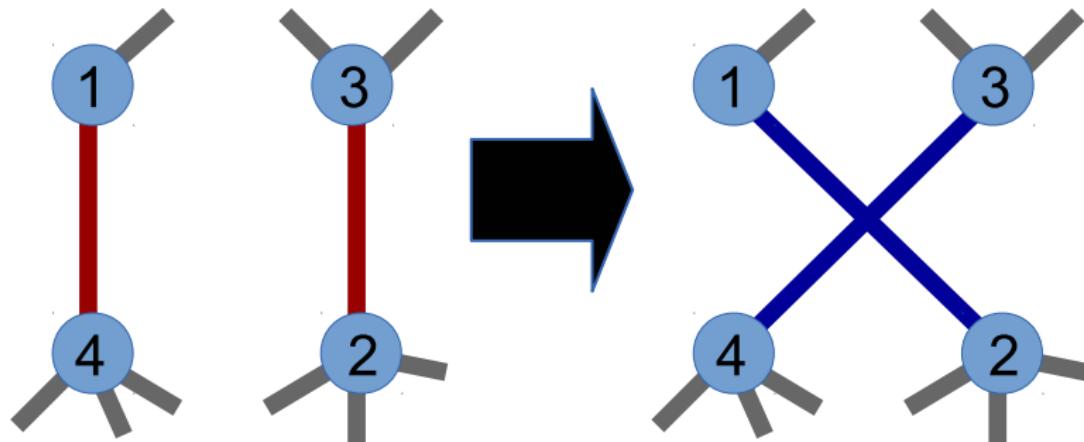
- ① Assign k_i stubs (half an edge) to each vertex i .
- ② Choose two stubs at random and join them into an edge.
- ③ Repeat until all stubs joined.



Numerical Implementation: From Network

Definition **B**, $\mathcal{G}(N, \{k_i\})$, with existing network given e.g. a *data set*

- ① Choose two edges at random.
- ② Swap one end of each edge with each other.
- ③ Repeat until every edge visited once.



Configuration Model: Summary

We have

- defined the Configuration Model,
- know two ways to create the Configuration Model numerically,

C & N Part II: End of Networks Lecture 8

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “The Configuration Model”

Intended Learning Outcomes

- Students can define the Configuration Model.
- Students know the key property of this model.

C & N Part II: Networks Lecture 9

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

The Price/Barabási-Albert Model

Intended Learning Outcomes

- Students can define the Price/BA model in terms of a growing network model
- Students can define the principle of cumulative advantage or preferential attachment in terms of numerical work, analytical work and in terms of real world examples.
- Students set up the master equation for the model
- Students can find the long-time large-degree solution for the degree distribution, a power law with power $\gamma = 3$.

PS= See problem sheet, **EFS**= Exercise For Students

4. Network Models

- 4.1 Why Model a Network?
- 4.2 Erdős-Rényi Random graph model
- 4.3 Configuration Model
- 4.4 Price Model**
- 4.5 (#) Generating Functions

Price or Barabási-Albert model



Derek de Solla Price

A simple model generating networks with power law fat-tailed degree distributions

★Will be the focus of the Networks part of the project ★

Citation Networks

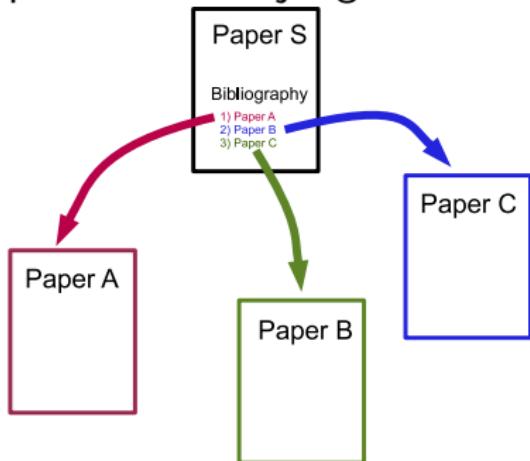
Citation Networks

Citation Network

In a citation network documents are the vertices, edges are made if one document cites a second.

A CITATION is a reference from one 'document' to an earlier one.

Examples: **Academic papers**, patents, court judgements



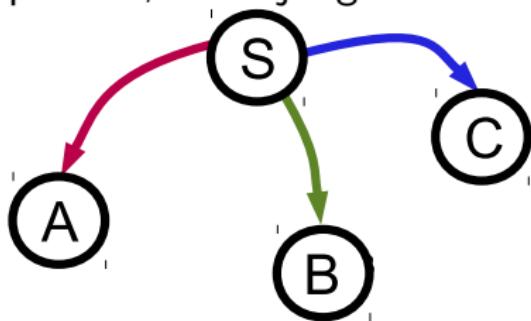
Citation Networks

Citation Network

In a citation network documents are the vertices, edges are made if one document cites a second.

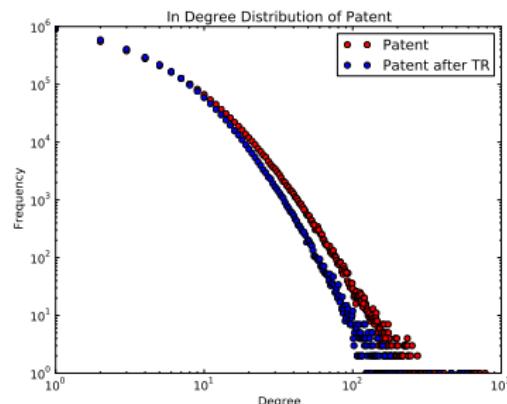
A CITATION is a reference from one 'document' to an earlier one.

Examples: **Academic papers**,
patents, court judgements



Key Citation Network Property

US Patents (1975-99)

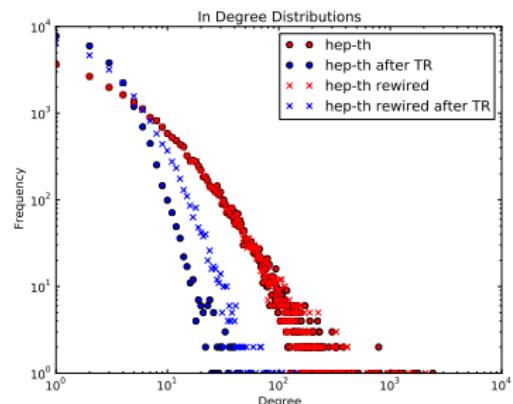


The number of citations
(the in-degree)
has a fat-tailed distribution.

$$(N = 3.810^6, \langle k^{(\text{in})} \rangle = 12.8)$$

[Figures: Clough et al. 2015, ignore the TR data]

Academic papers (arXiv:hep-th)



A few papers/patents get most of
the citations
Corollary: Many academic papers
(10% - 30%) have zero citations

$$(N = 2.710^4, \langle k^{(\text{in})} \rangle = 8.8)$$

Cumulative Advantage/Preferential Attachment

Price (1965) suggested that when a new paper is published, the probability that it cites an existing paper is

proportional to the number of citations it already has.

Principle of CUMULATIVE ADVANTAGE

“The more you have, the more you gain” appears in many contexts

- Pareto principle or the 80–20 rule, 19th cent.
- Matthew effect, Matthew’s gospel
 - “For everyone who has will be given more”
- Yule (1925) distribution of biological taxa and subtaxa
- Simon (1955) text frequency
- **BA** = BARABÁSI-ALBERT MODEL (1999) web sites,
 - ⇒ PREFERENTIAL ATTACHMENT

The Price/Barabási-Albert network model

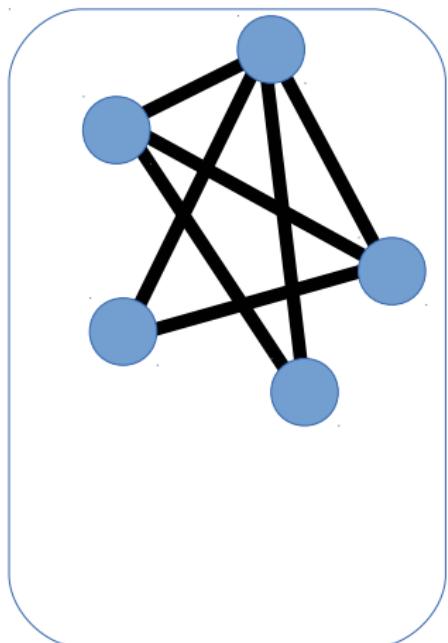
The Price/Barabási-Albert network model

Let us look at the mathematical analysis of this simple model of a growing network with a fat-tailed degree distribution used by Price for citations, Barabási-Albert for web pages.

★ For simplicity we will *choose* to work with **undirected** edges. ★
This follows the original mathematical analysis of Barabási and Albert but does *not* match the original context of either Price or Barabási-Albert where directed edges seem more appropriate

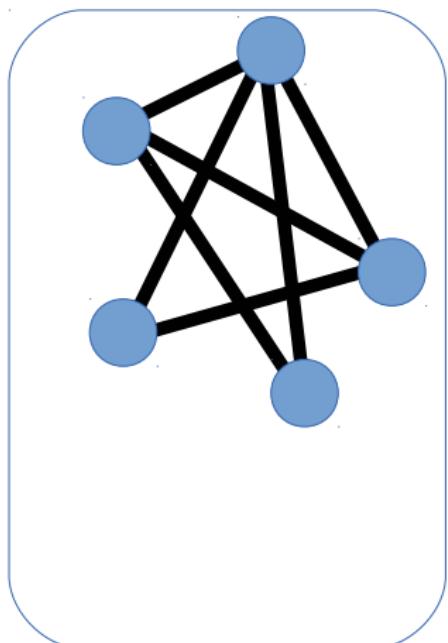
Rules for the BA (Price) Network Model

- ① Set up initial (citation) network.
- ② Increment time $t \rightarrow t + 1$ (order papers published)
- ③ Add one new vertex (new paper)
- ④ Add m edges (out-degree, bibliography length)
- ⑤ Connect each edge to an existing paper with probability proportional to in-degree $k^{(\text{in})}$ (citation count, Price) or total degree k (BA)
- ⑥ Repeat from 2



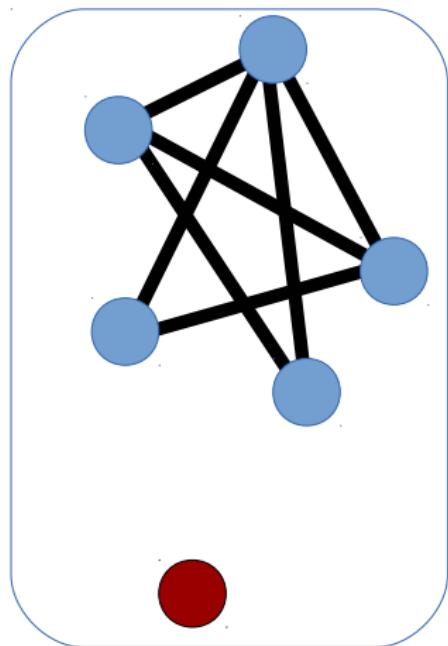
Rules for the BA (Price) Network Model

- ① Set up initial (citation) network.
- ② Increment time $t \rightarrow t + 1$ (order papers published)
- ③ Add one new vertex (new paper)
- ④ Add m edges (out-degree, bibliography length)
- ⑤ Connect each edge to an existing paper with probability proportional to in-degree $k^{(\text{in})}$ (citation count, Price) or total degree k (BA)
- ⑥ Repeat from 2



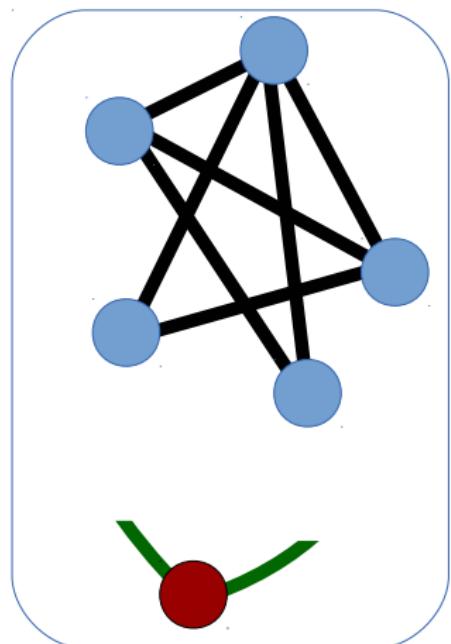
Rules for the BA (Price) Network Model

- ① Set up initial (citation) network.
- ② Increment time $t \rightarrow t + 1$ (order papers published)
- ③ Add one new vertex (new paper)
- ④ Add m edges (out-degree, bibliography length)
- ⑤ Connect each edge to an existing paper with probability proportional to in-degree $k^{(\text{in})}$ (citation count, Price) or total degree k (BA)
- ⑥ Repeat from 2



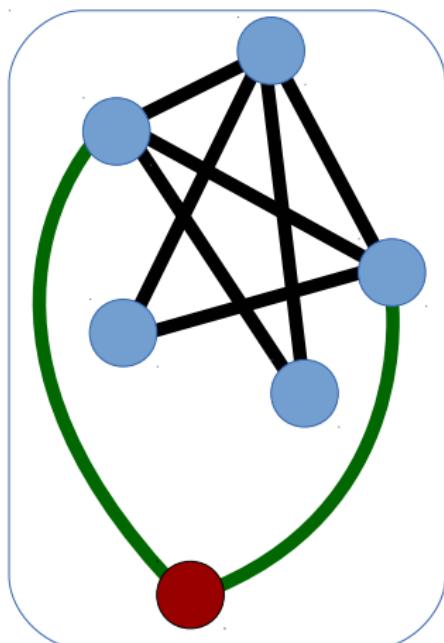
Rules for the BA (Price) Network Model

- ① Set up initial (citation) network.
- ② Increment time $t \rightarrow t + 1$ (order papers published)
- ③ Add one new vertex (new paper)
- ④ Add m edges (out-degree, bibliography length)
- ⑤ Connect each edge to an existing paper with probability proportional to in-degree $k^{(\text{in})}$ (citation count, Price) or total degree k (BA)
- ⑥ Repeat from 2



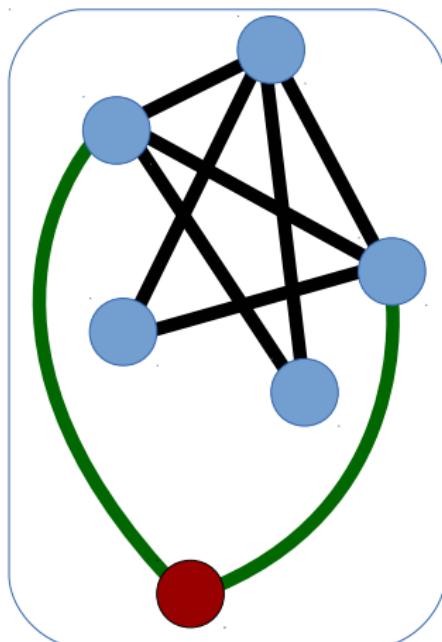
Rules for the BA (Price) Network Model

- ① Set up initial (citation) network.
- ② Increment time $t \rightarrow t + 1$ (order papers published)
- ③ Add one new vertex (new paper)
- ④ Add m edges (out-degree, bibliography length)
- ⑤ Connect each edge to an existing paper with probability proportional to in-degree $k^{(\text{in})}$ (citation count, Price) or total degree k (BA)
- ⑥ Repeat from 2



Rules for the BA (Price) Network Model

- ① Set up initial (citation) network.
- ② Increment time $t \rightarrow t + 1$ (order papers published)
- ③ Add one new vertex (new paper)
- ④ Add m edges (out-degree, bibliography length)
- ⑤ Connect each edge to an existing paper with probability proportional to in-degree $k^{(\text{in})}$ (citation count, Price) or total degree k (BA)
- ⑥ Repeat from 2



BA Model Master Equation

- k is the *total* degree of a vertex, edges undirected
- $n(k, t)$ is the number of nodes at time t with total degree k
- Add one new node attached to m new edges
- Each edge also attached to existing vertex with probability

$$\Pi(k, t) = \frac{k}{2E(t)}$$

where $E(t)$ is number of edges.

The MASTER EQUATION for the distribution is then roughly

$$\begin{aligned} n(k, t + 1) = & n(k, t) + m\Pi(k - 1, t)n(k - 1, t) \\ & - m\Pi(k, t)n(k, t) + \delta_{k,m} \end{aligned}$$

Master equations usually describe the evolution of the state of a system and enforce probability balance requirements.

BA Model Master Equation

The MASTER EQUATION for the distribution is then roughly

$$\begin{aligned}
 n(k, t+1) = & n(k, t) && \text{Start with this number} \\
 & + m\Pi(k-1, t)n(k-1, t) && \text{Add edges to vertices} \\
 & - m\Pi(k, t)n(k, t) && \text{with degree } k-1 \\
 & + \delta_{k,m}
 \end{aligned}$$

Add edges to vertices with degree k

One new $k = m$ node

with $\Pi(k, t)n(k, t) = 0$ for all $k < m$ (**PS**).

Master equations usually describe the evolution of the state of a system and enforce probability balance requirements.

BA Model Master Equation solution

$$\begin{aligned} N(t+1) &= N(t) + 1 \\ E(t+1) &= E(t) + m \end{aligned} \Rightarrow E(t) \approx m N(t)$$

(exact if $E(0) = m N(0)$ or in long-time limit)

$$n(k, t+1) = n(k, t) + \frac{m(k-1)n(k-1, t)}{2mN(t)} - \frac{mk n(k, t)}{2mN(t)} + \delta_{k,m}$$

Write in terms of probability distribution $p(k, t) = n(k, t)/N(t)$

$$\begin{aligned} N(t+1)p(k, t+1) - N(t)p(k, t) \\ = +\frac{1}{2}(k-1)p(k-1, t) - \frac{1}{2}k p(k, t) + \delta_{k,m} \end{aligned}$$

BA Model Long Time quick solution

$$\begin{aligned} N(t+1)p(k, t+1) - N(t)p(k, t) \\ = +\frac{1}{2}(k-1)p(k-1, t) - \frac{1}{2}k p(k, t) + \delta_{k,m} \end{aligned}$$

- Number of vertices always growing as $N(t+1) = N(t) + 1$
- $\Rightarrow n(k, t)$ must evolve too
but look for large t stable form for $p(k, t) = n(k, t)/N(t)$

$$\lim_{t \rightarrow \infty} p(k, t) = p_\infty(k)$$

$$p_\infty(k) = \frac{1}{2} [(k-1) p_\infty(k-1) - k p_\infty(k)] + \delta_{k,m}$$

BA Model Long Time quick solution

$$p_\infty(k) = \frac{1}{2} [(k-1)p_\infty(k-1) - k p_\infty(k)] + \delta_{k,m}$$

Take k large

- so we ignore low k boundary term $\delta_{k,m}$
- we may approximate the right hand side as a derivative ($\Delta k = 1$)

$$p(k) \approx -\frac{1}{2} \frac{k p_\infty(k) - (k - \Delta k) p_\infty(k - \Delta k)}{\Delta k} \quad (4.1)$$

$$\Rightarrow p(k) \approx -\frac{1}{2} \frac{\partial(k p_\infty(k))}{\partial k} \quad (4.2)$$

Can solve from here in a variety of ways (**PS**).

BA Long Time Approximate Solution

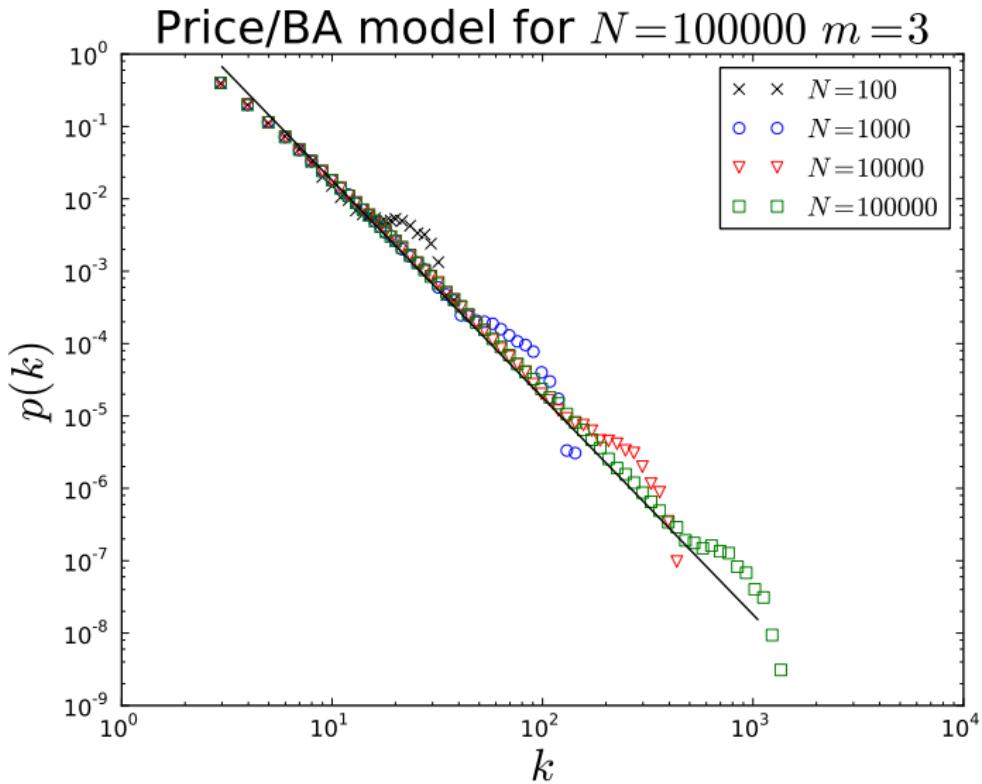
For long time and large k you find solution satisfies

$$\lim_{t \rightarrow \infty} p(k, t) = p_\infty(k) \propto \frac{1}{k^3}$$

Power law with power $\gamma = 3$.

- large k as used continuous k not discrete
- large k as we ignored low k boundary condition
- large t so the distribution $p(k, t)$ becomes stationary $p_\infty(k)$
- large t so the initial network values irrelevant
(we assumed $E(t)/N(t) = m$ and $\langle k \rangle = 2m$ exactly at all times.)

Price/BA Example



Price/BA model: Summary

We have

- defined the Price/BA model in terms of a growing network model and the principle of cumulative advantage or preferential attachment,
- given citations as an example where such a mechanism might be relevant
- set up the master equation for the model
- shown the long-time large-degree solution to be a power law with power $\gamma = 3$

C & N Part II: End of Networks Lecture 9

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “The Price/Barabási-Albert Model”

Intended Learning Outcomes

- Students can define the Price/BA model in terms of a growing network model
- Students can define the principle of cumulative advantage or preferential attachment in terms of numerical work, analytical work and in terms of real world examples.
- Students set up the master equation for the model
- Students can find the long-time large-degree solution for the degree distribution, a power law with power $\gamma = 3$.

C & N Part II: Networks Lecture 10

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

#Generating Functions

This lecture is optional and not examinable.

This introduces a very general technique that all students have used but may not have considered in a more abstract sense. Here we note that it can be used to help analyse the master equation of the Price/BA model.

Intended Learning Outcomes

- Students can give the basic properties of Generating Functions
- Students can cast the Price/BA master equation as a PDE for the generating function

PS= See problem sheet, **EFS**= Exercise For Students

4. Network Models

- 4.1 Why Model a Network?
- 4.2 Erdős-Rényi Random graph model
- 4.3 Configuration Model
- 4.4 Price Model
- 4.5 (#) Generating Functions

(#+) Generating Functions

Generating functions are a widely used trick in applied maths and theoretical physics.

Use to solve for size of GCC and average path length near phase transition for random graphs. **PS2**

The master equation of the Price/BA model can provide an illustration of their use.

(#) Integral Transforms

An extremely useful trick, e.g. for solving integrals and PDE

$$\text{Fourier} \quad \tilde{f}(\omega) = \int_{-\infty}^{+\infty} dt \exp\{-i\omega t\} f(t)$$

$$\text{Laplace} \quad \tilde{f}(s) = \int_0^{\infty} dt \exp\{-st\} f(t)$$

$$\text{Mellin} \quad \tilde{f}(w) = \int_0^{\infty} dt t^{w-1} f(t)$$

Relationship: $i\omega t \leftrightarrow st \leftrightarrow -w \ln(t)$

In each case replacing physical variable (time) with one more difficult to interpret — (angular) frequency ω , s (?), w (?).

(#) Generating Functions in Physics

Generating functions are ubiquitous

- Partition functions in statistical mechanics, sum over energy and number, sum over different members of the ensembles representing *fluctuations* in energy and number, to leave a function of β and μ — $Z(\beta, \mu) = \sum_i \exp\{-\beta(E_i - \mu N_i)\}$
- Quantum Field Theory, sum over different possible quantum field configurations of $\exp\{iS/\hbar\}$ gives generating function of the Green functions — $Z[j] = \sum_\phi \exp\{-(i/\hbar)(S[\phi] - j\phi)\}$

In each case

- Always sums/integrals over ensembles representing statistical/quantum fluctuations
- Often replacing physical variables with strange variables difficult to interpret — μ chemical potential, j sources in QFT
- Find generating function is a powerful mathematical tool

(#) Generating Functions for Degree Distributions

Define the GENERATING FUNCTION $G(z)$ for $p(k)$ as

$$\boxed{G(z) = \sum_{k=0}^{\infty} z^k p(k)} = p(0) + zp(1) + z^2 p(2) + \dots$$

- z has no simple interpretation, replaces degree k .
- $G(z)$ has all the information of $p(k)$ wrapped up in a different mathematical form,
a discrete version of an integral transform (Laplace, $e^{-s} = z$)
- So can invert by evaluating the k -th derivative at $z = 0$

$$p(k) = \frac{1}{k!} \left. \frac{d^k G}{dz^k} \right|_{z=0}.$$

- Often $G(z)$ satisfies a PDE while $p(k)$ satisfies a difference equation — we have lots of experience with PDE

(#+) Generating Function for ER degree distribution

For example, the degree distribution for an ER graph is a binomial distribution

$$\tilde{p}(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}$$

(#) Generating Function for ER degree distribution

For example, the degree distribution for an ER graph is a binomial distribution

$$\tilde{p}(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}$$

The generating function for this $p(k)$ is

$$\begin{aligned} G(z) &= \sum_{k=0}^{N-1} z^k p(k) \\ &= \dots \end{aligned} \tag{4.3}$$

(#) Generating Function for ER degree distribution

For example, the degree distribution for an ER graph is a binomial distribution

$$\tilde{p}(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}$$

The generating function is

$$\begin{aligned}
 G(z) &= \sum_{k=0}^{N-1} z^k p(k) \\
 &= \sum_{k=0}^{N-1} \binom{N-1}{k} (z p)^k (1-p)^{N-1-k} \\
 &= \dots
 \end{aligned} \tag{4.3}$$

(#) Generating Function for ER degree distribution

For example, the degree distribution for an ER graph is a binomial distribution

$$\tilde{p}(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}$$

The generating function is

$$\begin{aligned} G(z) &= \sum_{k=0}^{N-1} z^k p(k) \\ &= \sum_{k=0}^{N-1} \binom{N-1}{k} (z p)^k (1-p)^{N-1-k} \\ &= ((z p) + (1-p))^{N-1} \end{aligned}$$

(#) Properties of Generating Functions

- Can invert by taking k -th derivative at $z = 0$

$$p(k) = \frac{1}{k!} \left. \frac{d^k G}{dz^k} \right|_{z=0}.$$

- Integral transform also inverts the series

$$p(k) = \frac{1}{2\pi i} \oint dz z^{-k-1} G(z).$$

- MOMENTS of distribution are easy to obtain

$$\langle k^n \rangle = \left. \left(z \frac{d}{dz} \right)^k G(z, t) \right|_{z=1} = \left. \frac{d^k}{d(\ln(z))^k} G(z, t) \right|_{z=1}.$$

- FACTORIAL MOMENTS of distribution are even easier to obtain

$$\langle k(k-1)\dots(k-n+1) \rangle = \left\langle \frac{\Gamma(k+1)}{\Gamma(k-n)} \right\rangle = \left. \frac{d^n G}{dz^n} \right|_{z=1}.$$

(#) Generating Functions for Master Equations

Take our Price/BA model master *difference* equation

$$p_\infty(k) = \frac{1}{2}(k-1)p_\infty(k-1) - \frac{1}{2}k p_\infty(k) + \delta_{k,m}.$$

To write this in terms of $G(z) = \sum_{k=0}^{\infty} z^k p(k)$,

multiply by z^k and sum over k

$$\sum_{k=0}^{\infty} z^k k p_\infty(k) = \sum_{k=0}^{\infty} \left(z \frac{\partial z^k}{\partial z} \right) p_\infty(k)$$

$$= z \frac{\partial}{\partial z} \sum_{k=0}^{\infty} z^k p_\infty(k)$$

$$= z \frac{\partial}{\partial z} G(z)$$

(#) Generating Functions for Master Equations

Take our Price/BA model master *difference* equation

$$p_\infty(k) = \frac{1}{2}(k-1)p_\infty(k-1) - \frac{1}{2}k p_\infty(k) + \delta_{k,m}.$$

To write this in terms of $G(z) = \sum_{k=0}^{\infty} z^k p(k)$,

multiply by z^k and sum over k

$$\sum_{k=0}^{\infty} z^k k p_\infty(k) = \sum_{k=0}^{\infty} \left(z \frac{\partial z^k}{\partial z} \right) p_\infty(k)$$

$$= z \frac{\partial}{\partial z} \sum_{k=0}^{\infty} z^k p_\infty(k)$$

$$= z \frac{\partial}{\partial z} G(z)$$

(#) Generating Functions for Master Equations

Take our Price/BA model master *difference* equation

$$p_\infty(k) = \frac{1}{2}(k-1)p_\infty(k-1) - \frac{1}{2}k p_\infty(k) + \delta_{k,m}.$$

To write this in terms of $G(z) = \sum_{k=0}^{\infty} z^k p(k)$,

multiply by z^k and sum over k

$$\sum_{k=0}^{\infty} z^k k p_\infty(k) = \sum_{k=0}^{\infty} \left(z \frac{\partial z^k}{\partial z} \right) p_\infty(k)$$

$$= z \frac{\partial}{\partial z} \sum_{k=0}^{\infty} z^k p_\infty(k)$$

$$= z \frac{\partial}{\partial z} G(z)$$

(#) Generating Functions for Master Equations

Take our Price/BA model master *difference* equation

$$p_\infty(k) = \frac{1}{2}(k-1)p_\infty(k-1) - \frac{1}{2}k p_\infty(k) + \delta_{k,m}.$$

To write this in terms of $G(z) = \sum_{k=0}^{\infty} z^k p(k)$,

multiply by z^k and sum over k , e.g.

$$\sum_{k=0}^{\infty} z^k k p_\infty(k) = \sum_{k=0}^{\infty} \left(z \frac{\partial z^k}{\partial z} \right) p_\infty(k)$$

$$= z \frac{\partial}{\partial z} \sum_{k=0}^{\infty} z^k p_\infty(k)$$

$$= z \frac{\partial}{\partial z} G(z)$$

(#) Generating Functions for Master Equations

So if we multiply by z^k and sum over k ,
our Price/BA model master *difference* equation

$$p_\infty(k) = \frac{1}{2}(k-1)p_\infty(k-1) - \frac{1}{2}k p_\infty(k) + \delta_{k,m}.$$

becomes

$$G_\infty(z) = \frac{1}{2}z(z-1)\frac{\partial G_\infty(z)}{\partial z} + z^m.$$

Now we have a PDE — much more familiar

In fact discrete Price/BA is easy to solve if know difference equations
PS

(#) Generating Functions: Summary

This lecture is optional and not examinable.

We have

- seen that integral transforms of function of a continuous variable play the same role as generating functions for functions of discrete variables,
- looked at some the simple properties of Generating Functions,
- given the Price/BA master equation as a PDE for the generating function.

C & N Part II: End of Networks Lecture 10

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “(#) Generating Functions”

This lecture is optional and not examinable.

Intended Learning Outcomes

- Students can give the basic properties of Generating Functions
- Students can cast the Price/BA master equation as a PDE for the generating function

5. Phase Transitions in Random Graphs

- 5.1 The Phase Transition in an ER Random graph
- 5.2 Shortest Path Length in ER Random graph
- 5.3 The Degree of Your Neighbour
- 5.4 Phase Transition in Configuration model

C & N Part II: Networks Lecture 11

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

The Phase transition in an ER Random graph

Intended Learning Outcomes

- Students will be able to describe the behaviour of the model as p varies.
- Students will be able to calculate the size of LCC as a function of p .
- Students will be able to estimate the length scale in the model as p varies.

PS= See problem sheet, **EFS**= Exercise For Students

5. Phase Transitions in Random Graphs

5.1 The Phase Transition in an ER Random graph

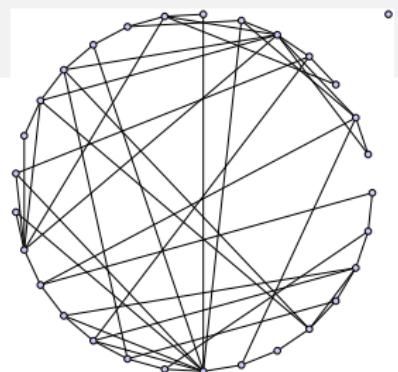
5.2 Shortest Path Length in ER Random graph

5.3 The Degree of Your Neighbour

5.4 Phase Transition in Configuration model

ER (Erdős-Rényi) Random Graphs

Here we use definition **A**, $\mathcal{G}(N, p)$



ER model definition **A** - $\mathcal{G}(N, p)$

N vertices in which each pair of vertices is connected with probability p , independent of all other pairs,

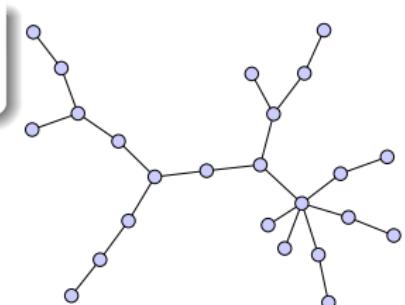
Stochastic definitions \Rightarrow sets of graphs $\mathcal{G} = \{G_1, G_2, \dots\}$.

Technical Note II: Trees

A Tree

A tree is a connected graph with no loops

- Undirected trees $E = N - 1$
Aside: $E - N + 1$ is the number of independent loops in a connected component of a simple graph
- Trees which are subgraphs of a network are useful tools e.g. MINIMAL SPANNING TREE
- Trees are fundamental in Computer Science
- The BETHE LATTICE is a regular tree (see Statistical Mechanics course)



You could give a whole course on trees, trees appear here briefly.

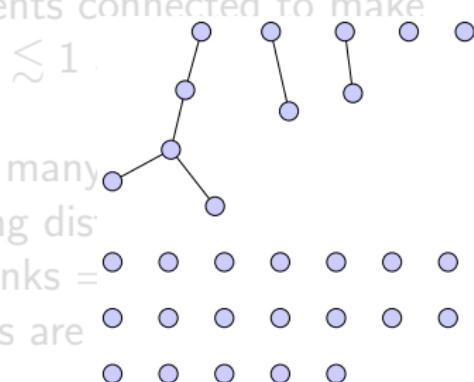
ER random graph Phase Transition

- at $\langle k \rangle = 0, p = 0$, simply N disconnected vertices
- at small p , $\langle k \rangle \ll 1$, only small connected components, short distances.
- as $\langle k \rangle$ and p increase, small components connected to make bigger clusters — lines or trees if $\langle k \rangle \lesssim 1$ as $\sigma_k^2 = \langle k \rangle$ means most $k_i = 0, 1$ or $2 (\sim \langle k \rangle \pm \sigma)$.
- at some point $\langle k \rangle = k_c \sim 1, p = p_c$, many "connected" in a tree structure i.e. long distances
- as $\langle k \rangle$ and p increase further, more links =
- for $\langle k \rangle > k_c$, *finite fraction* of vertices are GIANT CONNECTED COMPONENT
- at $p = 1$ we have the COMPLETE GRAPH of N nodes.

★ In the large sparse graph limit, $N \rightarrow \infty, \langle k \rangle = pN$, there is a ★ phase transition at $\langle k \rangle = k_c = 1$ — note $p = k_c/N \rightarrow 0$.

ER random graph Phase Transition

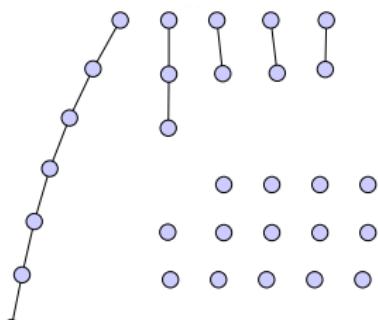
- at $\langle k \rangle = 0, p = 0$, simply N disconnected vertices
 - at small p , $\langle k \rangle \ll 1$, only small connected components, short distances.
 - as $\langle k \rangle$ and p increase, small components connected to make bigger clusters — lines or trees if $\langle k \rangle \lesssim 1$.
most $k_i = 0, 1$ or $2 (\sim \langle k \rangle \pm \sigma)$.
 - at some point $\langle k \rangle = k_c \sim 1, p = p_c$, many "connected" in a tree structure i.e. long distances
 - as $\langle k \rangle$ and p increase further, more links =
 - for $\langle k \rangle > k_c$, finite fraction of vertices are in the GIANT CONNECTED COMPONENT
 - at $p = 1$ we have the COMPLETE GRAPH of N nodes.
- ★ In the large sparse graph limit, $N \rightarrow \infty$, $\langle k \rangle = pN$, there is a ★ phase transition at $\langle k \rangle = k_c = 1$ — note $p = k_c/N \rightarrow 0$.



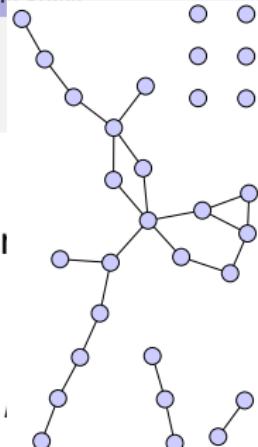
ER random graph Phase Transition

- at $\langle k \rangle = 0, p = 0$, simply N disconnected vertices
- at small p , $\langle k \rangle \ll 1$, only small connected components, short distances.
- as $\langle k \rangle$ and p increase, small components connected to make bigger clusters — lines or trees if $\langle k \rangle \lesssim 1$ as $\sigma_k^2 = \langle k \rangle$ means most $k_i = 0, 1$ or $2 (\sim \langle k \rangle \pm \sigma)$.
- at some point $\langle k \rangle = k_c \sim 1, p = p_c$, many "connected" in a tree structure i.e. long distances
- as $\langle k \rangle$ and p increase further, more links =
- for $\langle k \rangle > k_c$, finite fraction of vertices are in the GIANT CONNECTED COMPONENT
- at $p = 1$ we have the COMPLETE GRAPH

★ In the large sparse graph limit, $N \rightarrow \infty, \langle k \rangle = p/N$, there is a ★
phase transition at $\langle k \rangle = k_c = 1$ — note $p = k_c/N \rightarrow 0$.



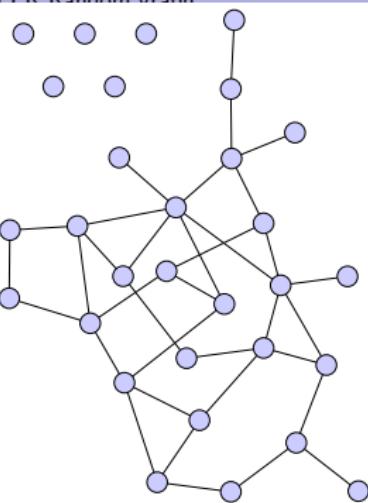
ER random graph Phase Transition



- at $\langle k \rangle = 0, p = 0$, simply N disconnected vertices
 - at small p , $\langle k \rangle \ll 1$, only small connected components.
 - as $\langle k \rangle$ and p increase, small components connect to bigger clusters — lines or trees if $\langle k \rangle \lesssim 1$ as $\sigma_k^2 = \langle \text{most } k_i = 0, 1 \text{ or } 2 \rangle (\sim \langle k \rangle \pm \sigma)$.
 - at some point $\langle k \rangle = k_c \sim 1, p = p_c$, many vertices “just connected” in a tree structure i.e. long distance scales.
 - as $\langle k \rangle$ and p increase further, more links \Rightarrow reduced distances
 - for $\langle k \rangle > k_c$, *finite fraction* of vertices are in a GCC, the GIANT CONNECTED COMPONENT
 - at $p = 1$ we have the COMPLETE GRAPH of N nodes.
- ★ In the large sparse graph limit, $N \rightarrow \infty$, $\langle k \rangle = pN$, there is a ★ phase transition at $\langle k \rangle = k_c = 1$ — note $p = k_c/N \rightarrow 0$.

ER random graph Phase Transition

- at $\langle k \rangle = 0, p = 0$, simply N disconnected vertices.
 - at small p , $\langle k \rangle \ll 1$, only small connected components — short distances.
 - as $\langle k \rangle$ and p increase, small components combine to form bigger clusters — lines or trees if $\langle k \rangle \lesssim 1$ as most $k_i = 0, 1$ or 2 ($\sim \langle k \rangle \pm \sigma$).
 - at some point $\langle k \rangle = k_c \sim 1, p = p_c$, many vertices "just connected" in a tree structure i.e. long distance scales.
 - as $\langle k \rangle$ and p increase further, more links \Rightarrow reduced distances
 - for $\langle k \rangle > k_c$, *finite fraction* of vertices are in a GCC, the GIANT CONNECTED COMPONENT
 - at $p = 1$ we have the COMPLETE GRAPH of N nodes.
- ★ In the large sparse graph limit, $N \rightarrow \infty, \langle k \rangle = pN$, there is a ★ phase transition at $\langle k \rangle = k_c = 1$ — note $p = k_c/N \rightarrow 0$.



ER random graph Phase Transition

- at $\langle k \rangle = 0, p = 0$, simply N disconnected vertices
 - at small p , $\langle k \rangle \ll 1$, only small connected components, short distances.
 - as $\langle k \rangle$ and p increase, small components connected to make bigger clusters — lines or trees if $\langle k \rangle \lesssim 1$ as $\sigma_k^2 = \langle k \rangle$ means most $k_i = 0, 1$ or 2 ($\sim \langle k \rangle \pm \sigma$).
 - at some point $\langle k \rangle = k_c \sim 1, p = p_c$, many vertices “just connected” in a tree structure i.e. long distance scales.
 - as $\langle k \rangle$ and p increase further, more links \Rightarrow reduced distances
 - for $\langle k \rangle > k_c$, *finite fraction* of vertices are in a GCC, the GIANT CONNECTED COMPONENT
 - at $p = 1$ we have the COMPLETE GRAPH of N nodes.
- ★ In the large sparse graph limit, $N \rightarrow \infty$, $\langle k \rangle = pN$, there is a ★
phase transition at $\langle k \rangle = k_c = 1$ — note $p = k_c/N \rightarrow 0$.

ER random graph Phase Transition

- at $\langle k \rangle = 0, p = 0$, simply N disconnected vertices
- at small p , $\langle k \rangle \ll 1$, only small connected components, short distances.
- as $\langle k \rangle$ and p increase, small components connected to make bigger clusters — lines or trees if $\langle k \rangle \lesssim 1$ as $\sigma_k^2 = \langle k \rangle$ means most $k_i = 0, 1$ or 2 ($\sim \langle k \rangle \pm \sigma$).
- at some point $\langle k \rangle = k_c \sim 1, p = p_c$, many vertices “just connected” in a tree structure i.e. long distance scales.
- as $\langle k \rangle$ and p increase further, more links \Rightarrow reduced distances
- for $\langle k \rangle > k_c$, *finite fraction* of vertices are in a GCC, the GIANT CONNECTED COMPONENT
- at $p = 1$ we have the COMPLETE GRAPH of N nodes.

★ In the large sparse graph limit, $N \rightarrow \infty$, $\langle k \rangle = pN$, there is a ★
phase transition at $\langle k \rangle = k_c = 1$ — note $p = k_c/N \rightarrow 0$.

ER random graph Phase Transition

- at $\langle k \rangle = 0, p = 0$, simply N disconnected vertices
 - at small p , $\langle k \rangle \ll 1$, only small connected components, short distances.
 - as $\langle k \rangle$ and p increase, small components connected to make bigger clusters — lines or trees if $\langle k \rangle \lesssim 1$ as $\sigma_k^2 = \langle k \rangle$ means most $k_i = 0, 1$ or 2 ($\sim \langle k \rangle \pm \sigma$).
 - at some point $\langle k \rangle = k_c \sim 1, p = p_c$, many vertices “just connected” in a tree structure i.e. long distance scales.
 - as $\langle k \rangle$ and p increase further, more links \Rightarrow reduced distances
 - for $\langle k \rangle > k_c$, *finite fraction* of vertices are in a GCC, the GIANT CONNECTED COMPONENT
 - at $p = 1$ we have the COMPLETE GRAPH of N nodes.
- ★ In the large sparse graph limit, $N \rightarrow \infty, \langle k \rangle = pN$, there is a ★
phase transition at $\langle k \rangle = k_c = 1$ — note $p = k_c/N \rightarrow 0$.

Technical Note: LCC and GCC

GCC: The Giant Connected Component

The Giant Connected Component is only defined in the $N \rightarrow \infty$ limit.

Definition: The GCC contains a *finite* fraction of the infinite number of vertices.

Only exists in some regions, here for $k > k_c$

Giant = Percolating Cluster

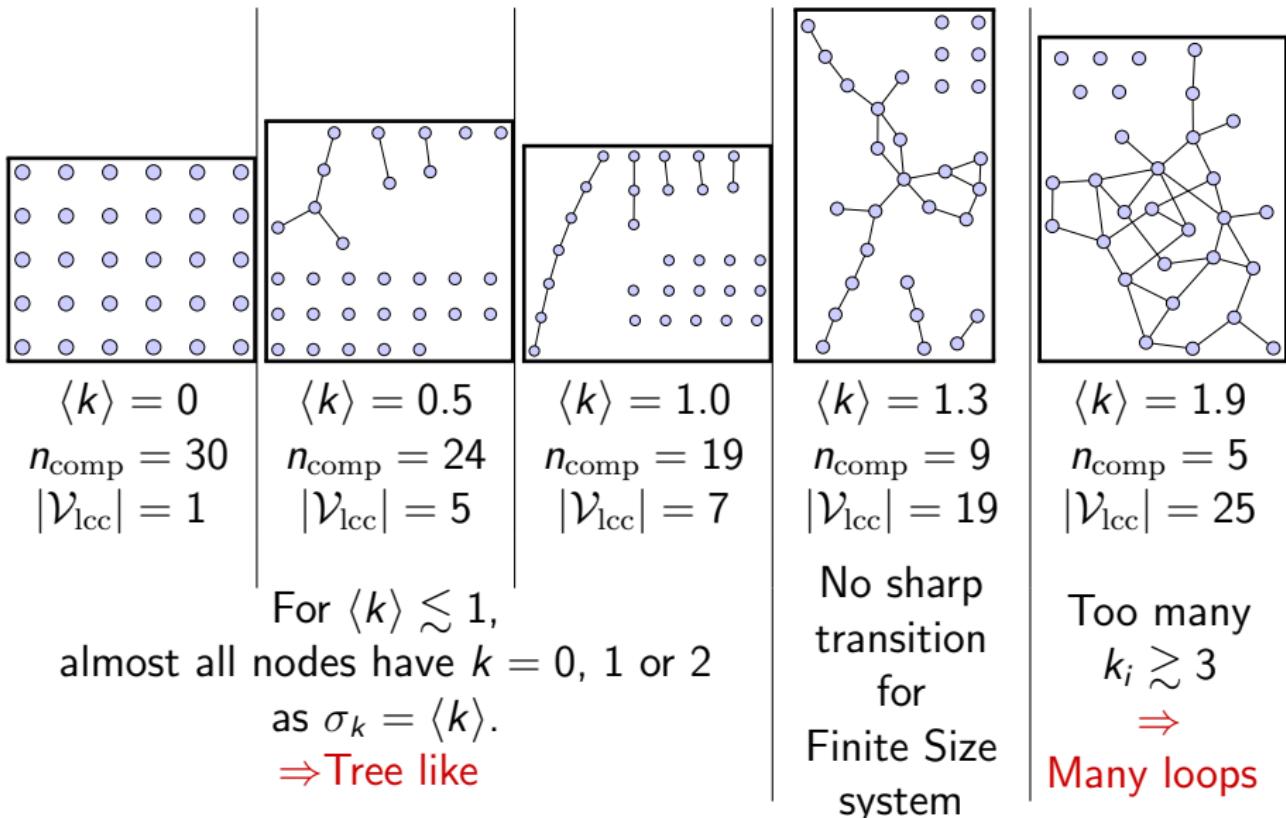
LCC: The Largest Connected Component

The Largest Connected Component is always defined.

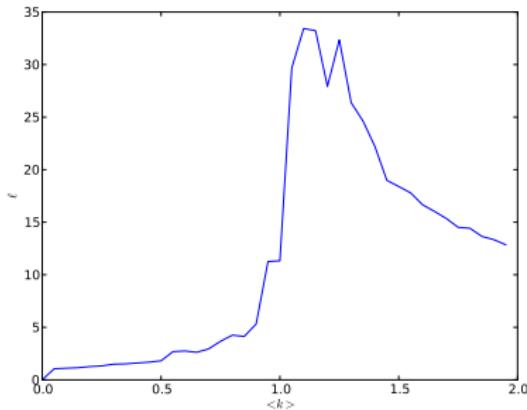
Easy to find numerically.

Approximates the GCC when N finite.

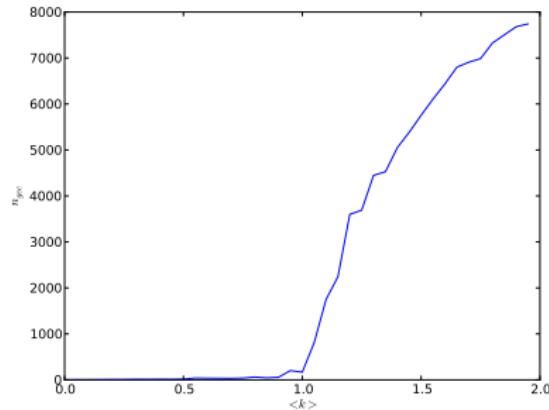
ER Random graph Transition when $N = 30$



ER Random graph Phase Transition: Plots



$\langle l \rangle$ vs. $\langle k \rangle$ for $N = 10^4$



Size of LCC vs. $\langle k \rangle$ for $N = 10^4$

★ Note: lines here are some fit to the data points with little meaning. Used here for visualisation purposes only. For scientific work always use data points with error bars, lines for known functions e.g. fits. Never ^(almost never) connect data points with meaningless lines.

ER Random graph Phase Transition: Calculation

Let u be the average fraction of vertices which are **not** in the GCC.
Put another way, u is the probability that a randomly chosen vertex does not lie in the GCC.

- Consider a vertex i which is not in the GCC
- Now pick one of the other $(N - 1)$ vertices, say j .
- If i is not in the GCC then two allowed cases for j :-
 - ① No edge from i to j — probability $(1 - p)$.
 - ② There is an edge to j (probability p) but j is not in the GCC (probability u) — overall probability pu .

Compare status of i to all other $(N - 1)$ vertices j gives

$$u = ((1 - p) + up)^{N-1} \quad (5.1)$$

ER Random graph Phase Transition: Calculation

Thus u , the average fraction of vertices **not** in the GCC is

$$u = ((1 - p) + up)^{N-1} \quad (5.2)$$

$$\ln(u) = (N - 1) \ln(1 - p + up) \quad (5.3)$$

$$\approx N.p(u - 1) \quad \text{for } N \rightarrow \infty, p \rightarrow 0 \quad (5.4)$$

$$\Rightarrow u = \exp\{-\langle k \rangle(1 - u)\} \quad (5.5)$$

Note: this is *exact* in the large sparse limit $N \rightarrow \infty$, $\langle k \rangle \sim O(1)$.

Writing in terms of the fractional size of the GCC, $S = 1 - u$, we have

$S = 1 - \exp\{-\langle k \rangle S\}$

for $N \rightarrow \infty, \langle k \rangle \sim O(1)$. (5.6)

ER Random graph Phase Transition: Solution

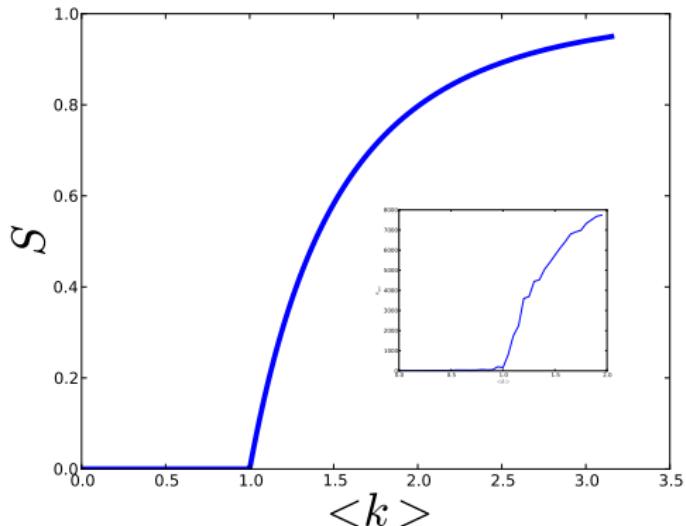
The fraction of vertices in the GCC, S , is

$$S = 1 - e^{-\langle k \rangle S} \quad (5.7)$$

Transcendental equation
 \Rightarrow no general solution,
 but you can show **PS**

$$k_c = 1 \quad (5.8)$$

Easy to solve numerically.
 Plot also shows $k_c \approx 1$



(Inset: Size of LCC vs. $\langle k \rangle$ for $N = 10^4$)

5. Phase Transitions in Random Graphs

5.1 The Phase Transition in an ER Random graph

5.2 Shortest Path Length in ER Random graph

5.3 The Degree of Your Neighbour

5.4 Phase Transition in Configuration model

ER Random graph Path Lengths

In the large sparse graph regime, $N \rightarrow \infty$, $\langle k \rangle \sim O(1)$:

- Find a *core* of vertices forming a small but connected subgraph \mathcal{G}_0 of $n_0 = |\mathcal{V}_0| \ll N$ vertices.
- The *average* number of edges these have to the other $(N - n_0)$ vertices is

$$p(N - n_0)n_0 \approx \langle k \rangle n_0 \quad (5.9)$$

- Unlikely that two edges from two different core vertices will connect to same new vertex — too many to choose $(N - n_0) \approx N$.
- Add these edges and the neighbouring vertices gives a new connected subgraph, \mathcal{G}_1 containing $\langle k \rangle n_0$ extra vertices.
- Repeating we find at distance ℓ from the core are $\langle k \rangle^\ell n_0$ vertices.

ER Random graph Path Lengths

In the large sparse graph regime, $N \rightarrow \infty$, $\langle k \rangle \sim O(1)$:

- Find a *core* of vertices forming a small but connected subgraph \mathcal{G}_0 of $n_0 = |\mathcal{V}_0| \ll N$ vertices.
- The *average* number of edges these have to the other $(N - n_0)$ vertices is

$$p(N - n_0)n_0 \approx \langle k \rangle n_0 \quad (5.9)$$

- Unlikely that two edges from two different core vertices will connect to same new vertex — too many to choose $(N - n_0) \approx N$.
- Add these edges and the neighbouring vertices gives a new connected subgraph, \mathcal{G}_1 containing $\langle k \rangle n_0$ extra vertices.
- Repeating we find at distance ℓ from the core are $\langle k \rangle^\ell n_0$ vertices.

ER Random graph Path Lengths

In the large sparse graph regime, $N \rightarrow \infty$, $\langle k \rangle \sim O(1)$:

- Find a *core* of vertices forming a small but connected subgraph \mathcal{G}_0 of $n_0 = |\mathcal{V}_0| \ll N$ vertices.
- The *average* number of edges these have to the other $(N - n_0)$ vertices is

$$p(N - n_0)n_0 \approx \langle k \rangle n_0 \quad (5.9)$$

- Unlikely that two edges from two different core vertices will connect to same new vertex — too many to choose $(N - n_0) \approx N$.
- Add these edges and the neighbouring vertices gives a new connected subgraph, \mathcal{G}_1 containing $\langle k \rangle n_0$ extra vertices.
- Repeating we find at distance ℓ from the core are $\langle k \rangle^\ell n_0$ vertices.

ER Random graph Path Lengths

In the large sparse graph regime, $N \rightarrow \infty$, $\langle k \rangle \sim O(1)$:

- Find a *core* of vertices forming a small but connected subgraph \mathcal{G}_0 of $n_0 = |\mathcal{V}_0| \ll N$ vertices.
- The *average* number of edges these have to the other $(N - n_0)$ vertices is

$$p(N - n_0)n_0 \approx \langle k \rangle n_0 \quad (5.9)$$

- Unlikely that two edges from two different core vertices will connect to same new vertex — too many to choose $(N - n_0) \approx N$.
- Add these edges and the neighbouring vertices gives a new connected subgraph, \mathcal{G}_1 containing $\langle k \rangle n_0$ extra vertices.
- Repeating we find at distance ℓ from the core are $\langle k \rangle^\ell n_0$ vertices.

ER Random graph Path Lengths

In the large sparse graph regime, $N \rightarrow \infty$, $\langle k \rangle \sim O(1)$:

- Find a *core* of vertices forming a small but connected subgraph \mathcal{G}_0 of $n_0 = |\mathcal{V}_0| \ll N$ vertices.
- The *average* number of edges these have to the other $(N - n_0)$ vertices is

$$p(N - n_0)n_0 \approx \langle k \rangle n_0 \quad (5.9)$$

- Unlikely that two edges from two different core vertices will connect to same new vertex — too many to choose $(N - n_0) \approx N$.
- Add these edges and the neighbouring vertices gives a new connected subgraph, \mathcal{G}_1 containing $\langle k \rangle n_0$ extra vertices.
- Repeating we find at distance ℓ from the core are $\langle k \rangle^\ell n_0$ vertices.

ER Random graph Path Lengths

Number of vertices at distance ℓ from core is

$$\langle k \rangle^\ell n_0 \quad (5.10)$$

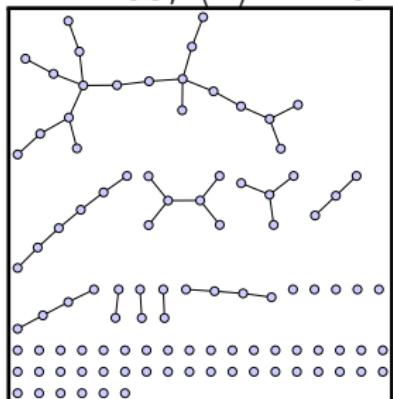
We have a BRANCHING PROCESS,

- a good approximation for a tree.
- ER random graphs are true trees in the sparse graph limit $N \rightarrow \infty$, $\langle k \rangle \sim O(1)$ (can be shown rigorously).

The critical point is at $\langle k \rangle = 1$ since

- $\langle k \rangle^\ell \rightarrow 0$ if $\langle k \rangle < 1$
- $\langle k \rangle^\ell \rightarrow \infty$ if $\langle k \rangle > 1$

$$N = 100, \langle k \rangle = 1.0$$



ER Random graph Phase Transition

ER random graphs are trees in the sparse graph limit $N \rightarrow \infty$

$\langle k \rangle \sim O(1)$ with $n_0 \langle k \rangle^\ell$ vertices distance ℓ from core n_0 vertices
 Critical point $\langle k \rangle = 1$

Sub Critical $\langle k \rangle < 1$

- Number added each step $n_0 \langle k \rangle^\ell \rightarrow 0$ as ℓ increases
- ⇒ Total number connected to core is a finite number
- ⇒ Any component must be zero fraction of the infinite total vertices N .
- ⇒ Path lengths, $\langle \ell \rangle, D$, finite even as $N \rightarrow \infty$. Can show that **PS**

$$\langle \ell \rangle \approx \frac{\langle k \rangle}{(1 - \langle k \rangle)} = \frac{p}{(N^{-1} - p)}, \quad \langle k \rangle < 1. \quad (5.11)$$

ER Random graph Phase Transition

ER random graphs are trees in the sparse graph limit $N \rightarrow \infty$

$\langle k \rangle \sim O(1)$ with $n_0 \langle k \rangle^\ell$ vertices distance ℓ from core n_0 vertices

Critical point $\langle k \rangle = 1$

Super Critical $\langle k \rangle > 1$

- Think of N very large but finite, take $N \rightarrow \infty$ at end
- Number added each step $n_0 \langle k \rangle^\ell$ grows exponentially as $\ell \uparrow$
- ⇒ Must stop when $n_0 \langle k \rangle^\ell \approx N$
- ⇒ Largest distance from core is

$$\ell_{\max} \approx \frac{1}{\ln(\langle k \rangle)} \ln(N) \quad (5.11)$$

Most people are connected $S > 0$ yet all paths lengths scale as $\ln(N)$ not as $N^{1/d}$ ⇒ Small World

ER Random graph Phase Transition

ER random graphs are trees in the sparse graph limit $N \rightarrow \infty$

$\langle k \rangle \sim O(1)$ with $n_0 \langle k \rangle^\ell$ vertices distance ℓ from core n_0 vertices

Critical point $\langle k \rangle = 1$

Critical Point $\langle k \rangle = 1$

\Rightarrow Largest distance from core is diverging even for large finite N

$$\ell_{\max} \approx \frac{1}{\ln(\langle k \rangle)} \ln(N) = \infty \quad (5.11)$$

N.B. For $\langle k \rangle < 1$ we also see $\langle \ell \rangle \approx \frac{\langle k \rangle}{(1-\langle k \rangle)}$ diverges as $\langle k \rangle \rightarrow 1^-$ **PS**

ER Random graph above the Phase Transition

Super Critical $\langle k \rangle > 1$

- Think of N very large but finite
- Number added each step $n_0 \langle k \rangle^\ell$ grows exponentially as ℓ increases
- ⇒ Must stop when $n_0 \langle k \rangle^\ell \approx N$
- ⇒ Largest distance from core is

$$\ell_{\max} \approx \frac{1}{\ln(\langle k \rangle)} \ln(N) \quad (5.12)$$

Most people are connected $S > 0$ yet all paths lengths scale as $\ln(N)$ not as $N^{1/d}$ ⇒ Small World

C & N Part II: End of Networks Lecture 11

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “The Phase transition in an ER Random graph”

Intended Learning Outcomes

- Students will be able to describe the behaviour of the model as p varies.
- Students will be able to calculate the size of LCC as a function of p .
- Students will be able to estimate the length scale in the model as p varies.

C & N Part II: Networks Lecture 12

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

General Properties of the Configuration Model

Intended Learning Outcomes

- Students will be able to derive the degree distribution of neighbouring nodes for such graphs
- Students will be able to estimate the typical length scales in this model
- Students will be use the length scales to probe the phase transition in this model.
- Students will be able to demonstrate the small world property of the model.

PS= See problem sheet, **EFS**= Exercise For Students

5. Phase Transitions in Random Graphs

5.1 The Phase Transition in an ER Random graph

5.2 Shortest Path Length in ER Random graph

5.3 The Degree of Your Neighbour

5.4 Phase Transition in Configuration model

General Random Graphs

Definition

A RANDOM GRAPH is one network from a family of graphs defined by a stochastic process.

CARE not everyone uses the term 'random graph in the same way.

- Sometimes the term refers to ER random graph only.
- Sometimes it means a graph with **no statistical correlations between neighbouring vertices**.
- Some will use it for any graph made from a stochastic process

Look at the context to see what 'random graph' means.

Correlations and Random Graphs

In this section we look at UNCORRELATED RANDOM GRAPHS with no statistical correlations between neighbouring vertices.

This includes

- ER random graph,
- The Configuration model gives uncorrelated random graphs.

The Price/BA model is stochastic (random) but it does NOT one of these types of uncorrelated random graphs.

- Certain vertices are more likely to be connected to other vertices than in a random graph.
- The Configuration model procedure applied to a Price/BA network produces a different family of networks,

#Aside. A few uncorrelated random graphs are not reachable from the configuration model but these are not important.

Degree of Your Neighbour in a Random Graph

Exploit lack of statistical correlations between neighbouring vertices in an uncorrelated random graph.

- ① Choose a vertex at random, all are equivalent statistically, so we choose i with probability $1/N$
- ② Choose one neighbour j , again all are equivalent
- ③ Probability of arriving at a neighbouring vertex with degree is **not** $1/N$
- ④ We have actually chosen an edge at random (to leave i) so probability vertex j is at end of a random edge is proportional to k_j — the number of different ways of arriving at a vertex

Degree of Your Neighbour in a Random Graph

Exploit lack of statistical correlations between neighbouring vertices in an uncorrelated random graph.

- ① Choose a vertex at random, all are equivalent statistically, so we choose i with probability $1/N$
- ② Choose one neighbour j , again all are equivalent
- ③ Probability of arriving at a neighbouring vertex with degree is **not** $1/N$
- ④ We have actually chosen an edge at random (to leave i) so probability vertex j is at end of a random edge is proportional to k_j — the number of different ways of arriving at a vertex

Degree of Your Neighbour in a Random Graph

Exploit lack of statistical correlations between neighbouring vertices in an uncorrelated random graph.

- ① Choose a vertex at random, all are equivalent statistically, so we choose i with probability $1/N$
- ② Choose one neighbour j , again all are equivalent
- ③ Probability of arriving at a neighbouring vertex with degree is **not** $1/N$
- ④ We have actually chosen an edge at random (to leave i) so probability vertex j is at end of a random edge is proportional to k_j — the number of different ways of arriving at a vertex

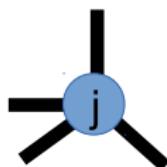
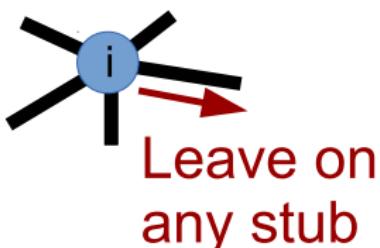
Degree of Your Neighbour in a Random Graph

Exploit lack of statistical correlations between neighbouring vertices in an uncorrelated random graph.

- ① Choose a vertex at random, all are equivalent statistically, so we choose i with probability $1/N$
- ② Choose one neighbour j , again all are equivalent
- ③ Probability of arriving at a neighbouring vertex with degree is **not** $1/N$
- ④ We have actually chosen an edge at random (to leave i) so probability vertex j is at end of a random edge is proportional to k_j — the number of different ways of arriving at a vertex

Degree of Your Neighbour in a Random Graph

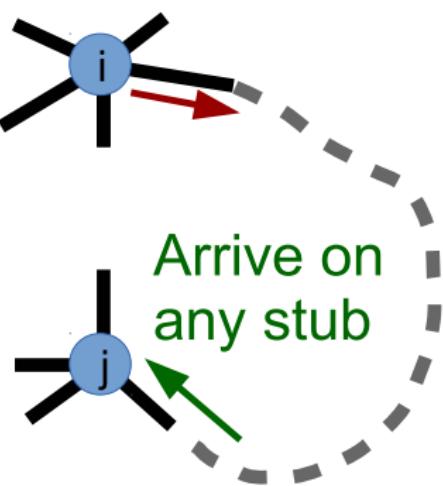
Exploit lack of statistical correlations between neighbouring vertices in an uncorrelated random graph.



For a random graph with no vertex correlations, this is like choosing the random end of a randomly chosen edge

Degree of Your Neighbour in a Random Graph

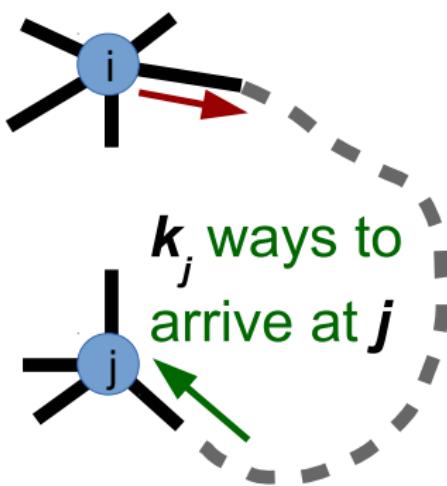
Exploit lack of statistical correlations between neighbouring vertices in an uncorrelated random graph.



For a random graph with no vertex correlations, this is like choosing the random end of a randomly chosen edge

Degree of Your Neighbour in a Random Graph

Exploit lack of statistical correlations between neighbouring vertices in an uncorrelated random graph.



For a random graph with no vertex correlations, this is like choosing the random end of a randomly chosen edge

Degree of Your Neighbour in a Random Graph

No statistical correlations between neighbouring vertices in our uncorrelated random graphs.

Probability distribution of neighbour's degree being k

$$p_{\text{nn}}(k) = \frac{k}{\langle k \rangle} p(k) \quad (5.13)$$

- ① It is NOT just $p(k)$, we are not simply choosing a random vertex
- ② The average degree of your neighbour $\langle k_{\text{nn}} \rangle$ is

$$\langle k_{\text{nn}} \rangle = \sum_k k p_{\text{nn}}(k) = \sum_k k \frac{k}{\langle k \rangle} p(k) = \frac{\langle k^2 \rangle}{\langle k \rangle} \quad (5.14)$$

Your friend is more popular than you! (as $\langle k^2 \rangle \geq (\langle k \rangle)^2$)
But only if friendship is random.

Degree of Your Neighbour in a Random Graph

No statistical correlations between neighbouring vertices in our uncorrelated random graphs.

Probability distribution of neighbour's degree being k

$$p_{\text{nn}}(k) = \frac{k}{\langle k \rangle} p(k) \quad (5.13)$$

- ① It is NOT just $p(k)$, we are not simply choosing a random vertex
- ② The average degree of your neighbour $\langle k_{\text{nn}} \rangle$ is

$$\langle k_{\text{nn}} \rangle = \sum_k k p_{\text{nn}}(k) = \sum_k k \frac{k}{\langle k \rangle} p(k) = \frac{\langle k^2 \rangle}{\langle k \rangle} \quad (5.14)$$

Your friend is more popular than you! (as $\langle k^2 \rangle \geq (\langle k \rangle)^2$)
 But only if friendship is random.

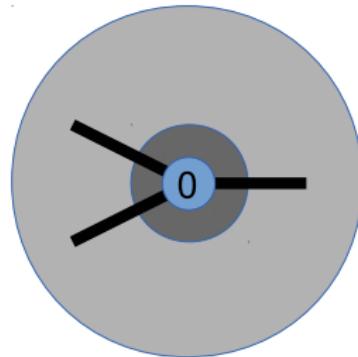
5. Phase Transitions in Random Graphs

- 5.1 The Phase Transition in an ER Random graph
- 5.2 Shortest Path Length in ER Random graph
- 5.3 The Degree of Your Neighbour
- 5.4 Phase Transition in Configuration model

From Root to First Neighbours

In large sparse graph regime, $N \rightarrow \infty$, $\langle k \rangle \sim O(1)$

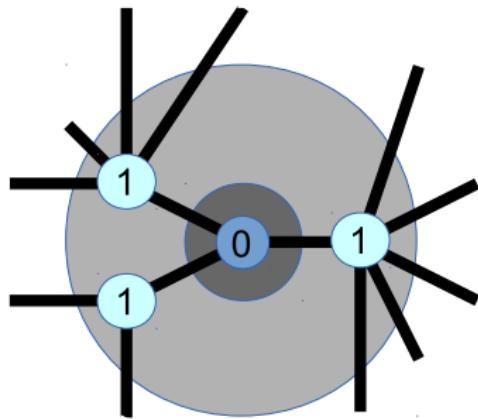
- Consider any one root vertex i
- Number of nodes distance $\ell = 0$ from root i is $n_0 = 1$
- The root i has k_i neighbours.
- So the number of nodes distance $\ell = 1$ from root i is k_i .
 $n_1 = k_i n_0 = k_i$.



From First Neighbours to Second Neighbours

In large sparse graph regime, $N \rightarrow \infty$, $\langle k \rangle \sim O(1)$

- On average the neighbours of i each have $z = \langle k^2 \rangle / \langle k \rangle$ neighbours.
- One of these edges returns to node i
- $(z - 1)$ edges go out to new vertices at $\ell = 2$ from root.
- Number of nodes $\ell = 2$ steps from root
 $n_2 = (z - 1)n_1$

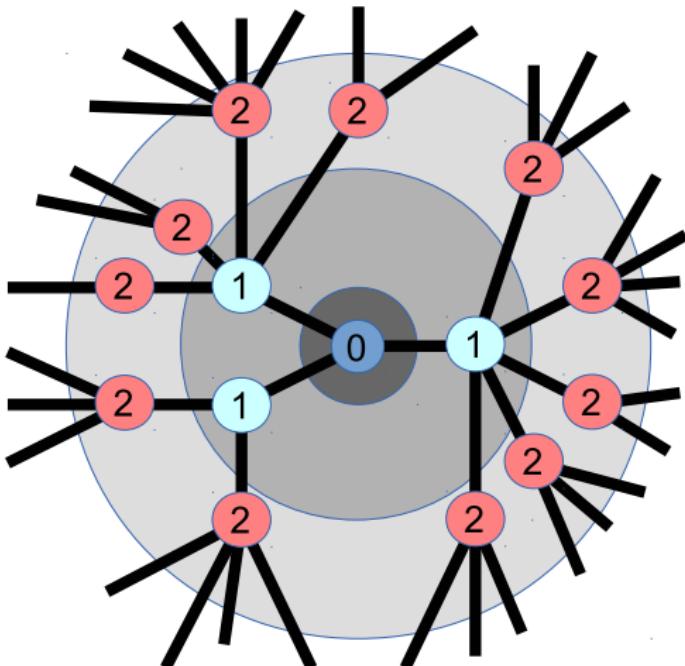


As $(z - 1)n_1$ is finite and with $N \rightarrow \infty$, all target vertices must be unique.

From Second Neighbours to Third Neighbours

In large sparse graph regime, $N \rightarrow \infty$, $\langle k \rangle \sim O(1)$

- On average the second neighbours of i each have $z = \langle k^2 \rangle / \langle k \rangle$ neighbours.
- One of these edges returns to an $\ell = 1$ node
- $(z - 1)$ edges go out to new $\ell = 3$ vertices.
- Number of nodes $\ell = 3$ steps from root
 $n_3 = (z - 1)n_2$



As $(z - 1)n_2$ is finite and with $N \rightarrow \infty$, all target vertices must be unique.

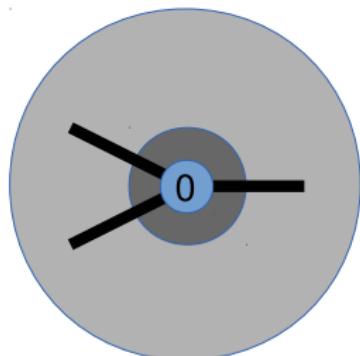
Branching Process and Breadth First Search

We are looking at a
BRANCHING PROCESS, a
good approximation if the
network looks like a tree —
no loops

$$n_\ell = (z - 1)^{\ell-1} n_1 \quad (5.15)$$

$$z = \frac{\langle k^2 \rangle}{\langle k \rangle} \quad (5.16)$$

This process is known as
the BREADTH FIRST
SEARCH of a graph.



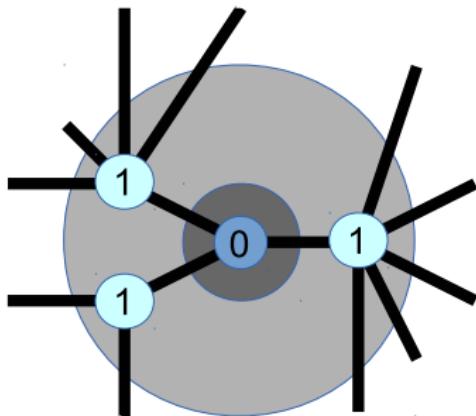
Branching Process and Breadth First Search

We are looking at a
BRANCHING PROCESS, a
good approximation if the
network looks like a tree —
no loops

$$n_\ell = (z - 1)^{\ell-1} n_1 \quad (5.15)$$

$$z = \frac{\langle k^2 \rangle}{\langle k \rangle} \quad (5.16)$$

This process is known as
the BREADTH FIRST
SEARCH of a graph.



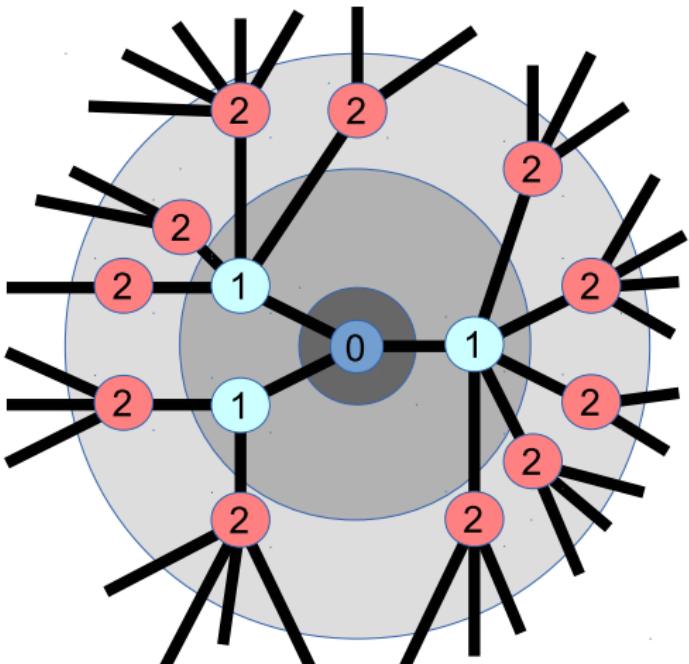
Branching Process and Breadth First Search

We are looking at a
BRANCHING PROCESS, a
good approximation if the
network looks like a tree —
no loops

$$n_\ell = (z - 1)^{\ell-1} n_1 \quad (5.15)$$

$$z = \frac{\langle k^2 \rangle}{\langle k \rangle} \quad (5.16)$$

This process is known as
the BREADTH FIRST
SEARCH of a graph.



Path Lengths in Uncorrelated Random Graphs

In large sparse graph regime, $N \rightarrow \infty$, $\langle k \rangle \sim O(1)$

- Consider one initial vertex i with degree k_i .
- This has $n_1 = k_i$ nearest neighbours — vertices one step away.
- The average degree of these neighbours is $z = \langle k^2 \rangle / \langle k \rangle$ not $\langle k \rangle$.
- One edge is back to vertex i , remaining $(z - 1)$ may be to new neighbours.
- Sparse large graph so these new neighbours are unlikely to be the same vertex, never if $N \rightarrow \infty$.
- Thus the number of vertices two steps away, n_2 , is

$$n_2 = (z - 1)n_1 \quad \text{with } z = \frac{\langle k^2 \rangle}{\langle k \rangle} \quad (5.17)$$

- Repeating we find the number of vertices ℓ steps away is
 $n_\ell = (z - 1)^{\ell-1} n_1$

Phase Transition: Critical Point

Random graphs are trees in the large sparse graph limit

$N \rightarrow \infty$, $\langle k^2 \rangle, \langle k \rangle \sim O(1)$ with average degree in tree $z = \langle k^2 \rangle / \langle k \rangle$

Critical point $z = 2$

A constant number of nodes are added each step — $n_{\ell+1} = n_\ell$

⇒ Critical point is when neighbours have average degree of 2:
 $\langle k_{nn} \rangle = 2$, i.e. **one edge in, one edge out**

Aside: For those who did Statistical Mechanics, this is the same mathematics as used for studying percolation on the Bethe Lattice. Only the context and hence meaning of some terms has changed.

Phase Transition: Sub Critical region

Random graphs are trees in the large sparse graph limit

$N \rightarrow \infty$, $\langle k^2 \rangle, \langle k \rangle \sim O(1)$ with average degree in tree $z = \langle k^2 \rangle / \langle k \rangle$

Sub Critical $z < 2$

- Nodes added each step $n_1(z - 1)^{\ell-1} \rightarrow 0$ as ℓ increases
- Total number connected to core is a finite number

The sum $\sum_{n=0}^{\infty} x^n$ converges if $|x| < 1$.

- ⇒ Any component must be zero fraction of the infinite total vertices N .
- ⇒ Path lengths, $\langle \ell \rangle, D$, finite

Phase Transition: Super Critical region

Random graphs are trees in the large sparse graph limit

$N \rightarrow \infty$ $\langle k^2 \rangle, \langle k \rangle \sim O(1)$ with average degree in tree $z = \langle k^2 \rangle / \langle k \rangle$

Super Critical $z > 2$

- Think of N very large but finite, take $N \rightarrow \infty$ at end
 - Number at distance ℓ grows exponentially as ℓ increases
- ⇒ Must stop when $n_1(z - 1)^{\ell-1} \approx N$
- ⇒ Largest distance from core is

$$\ell_{\max} \approx \frac{1}{\ln(z - 1)} \ln(N) \quad (5.18)$$

Most people are connected and path lengths scale very slowly as $\ln(N)$ not as $N^{1/d}$ Random graphs are ⇒ Small World

Small World Definition using Null Models

No precise definition, but here are two reasonable ones:-

Mathematical Definition

A SMALL WORLD NETWORK is one where $N \rightarrow \infty$ the lengths scale as $\ln(N)$ not as $N^{1/d}$

Mathematical definition of little use for practical data. This is an example of when a null model may be useful.

Null Model Definition

A SMALL WORLD NETWORK is one which has similar length scales as a random graph and much shorter length scales than a regular lattice with same number of vertices N and edges E .

C & N Part II: End of Networks Lecture 12

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “General Properties of the Configuration Model”

Intended Learning Outcomes

- Students will be able to derive the degree distribution of neighbouring nodes for such graphs
- Students will be able to estimate the typical length scales in this model
- Students will be use the length scales to probe the phase transition in this model.
- Students will be able to demonstrate the small world property of the model.

6. Network Measures without Eigenvalue Calculations

6.1 Recap

6.2 Betweenness

6.3 Clustering Coefficient

6.4 The Small World model

C & N Part II: Networks Lecture 13

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

Network Measures without Eigenvalue Calculations

Intended Learning Outcomes

- Students will know how to calculate various types of betweenness and cluster coefficients.
- Students can interpret these measures in term of social science concepts.
- Students will be able to define the Watts-Strogatz small world model.

PS= See problem sheet, **EFS**= Exercise For Students

6. Network Measures without Eigenvalue Calculations

6.1 Recap

6.2 Betweenness

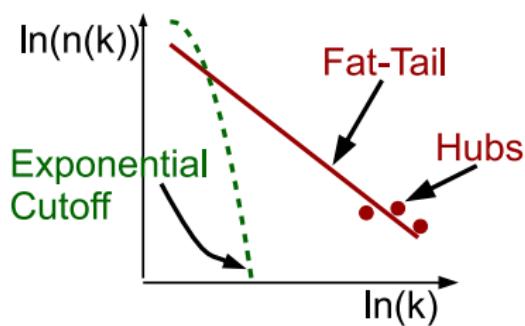
6.3 Clustering Coefficient

6.4 The Small World model

Degree

Degree k

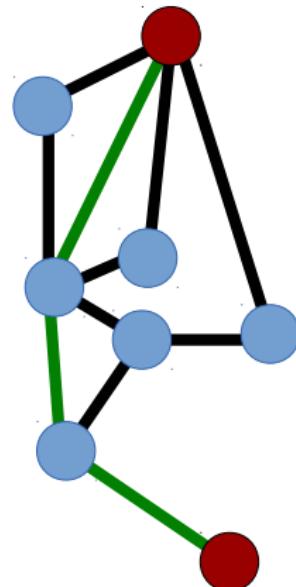
- most local measure of network properties
- still a useful measure of importance = CENTRALITY
- degree distribution $p(k)$
 - fat tail vs. exponential cutoffs
 - a few large hubs vs. not much variation



Shortest path

Paths

- AVERAGE SHORTEST PATH $\langle \ell \rangle$
- DIAMETER D — longest shortest path
- COMPONENT: number and their size,
- BETWEENNESS — Granovetter “Strength of Weak Links”.
- (#) CLOSENESS
 - not discussed here, see TSE notes.



6. Network Measures without Eigenvalue Calculations

6.1 Recap

6.2 Betweenness

6.3 Clustering Coefficient

6.4 The Small World model

Betweenness

Vertex Betweenness (Simple Definition)

Vertex Betweenness

The BETWEENNESS of a vertex is the number of shortest paths passing through a vertex, where we consider paths between all connected vertex pairs.

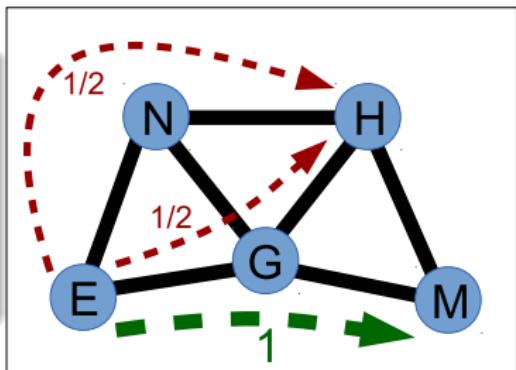
BUT

often more than one

shortest path between vertices

⇒ weight each path by $1/(\text{number paths})$

(N=Newton, G=Galileo, M=Mersenne, H=Huygens, E=Euclid)



Vertex Betweenness (Simple Definition)

Vertex Betweenness

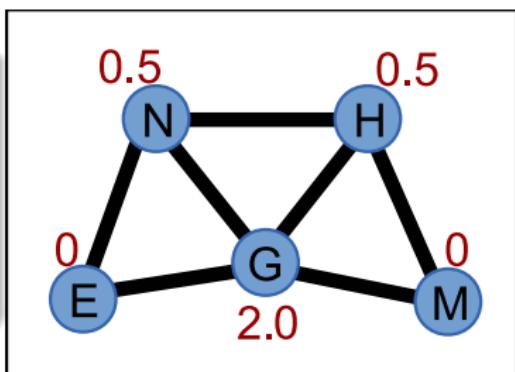
The BETWEENNESS of a vertex is the number of shortest paths passing through a vertex, where we consider paths between all connected vertex pairs.

BUT

often more than one

shortest path between vertices

⇒ weight each path by $1/(\text{number paths})$



Two shortest paths
between E and H ,
⇒ weight each by $1/2$

(N=Newton, G=Galileo, M=Mersenne, H=Huygens, E=Euclid)

Vertex Betweenness (full definition)

Vertex Betweenness

The BETWEENNESS of a vertex is the number of shortest paths passing through a vertex, where we weight each path by the number of shortest paths between each pair of points and vertices at either end of a path do not contribute to betweenness.

Vertex betweenness $b(i)$ of node i

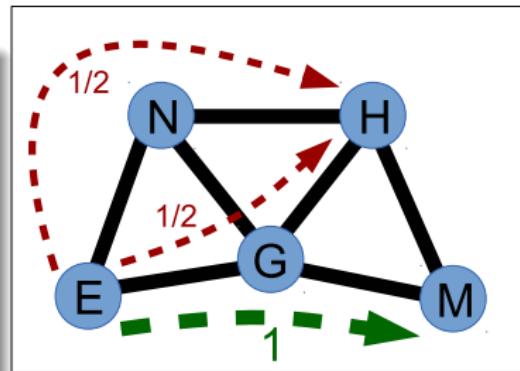
$$b(i) = \frac{1}{Z_b} \sum_{s,t \in V} \sum_{P \in \mathcal{P}_{\text{sp}}(s,t)} \sum_{j \in P_b \subseteq P} \frac{1}{|\mathcal{P}_{\text{sp}}(s,t)|},$$

- $\mathcal{P}_{\text{sp}}(s, t)$ is the set of shortest paths from s to t
- End points of paths usually excluded, $P_b(s, t) = P \setminus \{s, t\}$
- Constant normalisation Z_b , e.g. $Z_b = 2$ for undirected graphs.

Edge Betweenness

Edge Betweenness

The BETWEENNESS of an edge is the number of shortest paths passing through an edge, where we consider paths between all connected vertex pairs, excluding endpoints of paths and weighting multiple shortest paths appropriately.



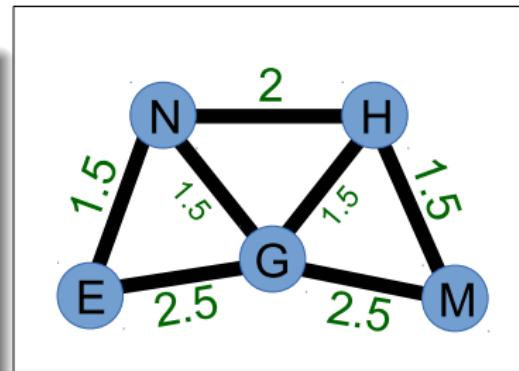
Again, usually exclude endpoints from shortest paths and weight paths by number of shortest paths between two points.

(N=Newton, G=Galileo, M=Mersenne, H=Huygens, E=Euclid)

Edge Betweenness

Edge Betweenness

The BETWEENNESS of an edge is the number of shortest paths passing through an edge, where we consider paths between all connected vertex pairs, excluding endpoints of paths and weighting multiple shortest paths appropriately.



Again, usually exclude endpoints from shortest paths and weight paths by number of shortest paths between two points.

(N=Newton, G=Galileo, M=Mersenne, H=Huygens, E=Euclid)

Granovetter — The Strength of Weak Ties

Granovetter “The Strength of Weak Ties” (1973)

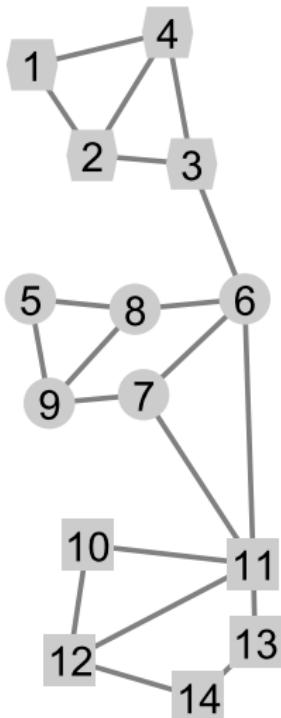
Some of the most important links are those used only occasionally



e.g. meeting someone from a different company can provide information, inspiration you don't get in your regular social interactions.

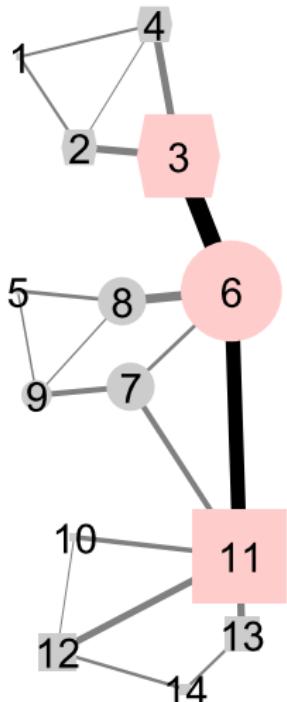
Quantitative: EDGE PERCOLATION — remove edges of highest/lowest betweenness and look at network structure e.g. number and size of components.

Strength of Weak Ties & Betweenness



Which nodes and edges have the highest betweenness?

Strength of Weak Ties & Betweenness



Larger the node,
wider the edge,
higher the betweenness.

6. Network Measures without Eigenvalue Calculations

6.1 Recap

6.2 Betweenness

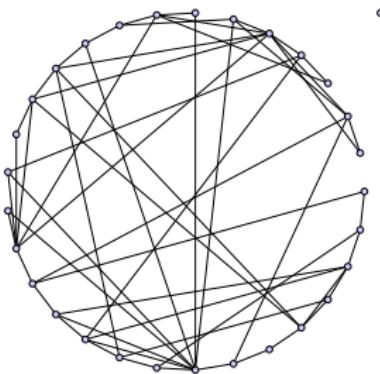
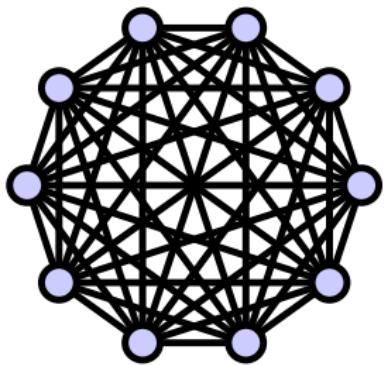
6.3 Clustering Coefficient

6.4 The Small World model

The Clustering Coefficient

The Clustering Coefficient

How can we capture the idea that any friends of mine are likely to be friends of each other?



Complete graph, $N = 10$

ER random graph, $N = 30$, $\langle k \rangle = 4.0$

"All my friends are friends"

"My friends don't know each other"

★ Need to look at edges of neighbours so moving at least one ★
step further away than degree

Vertex Clustering Coefficient

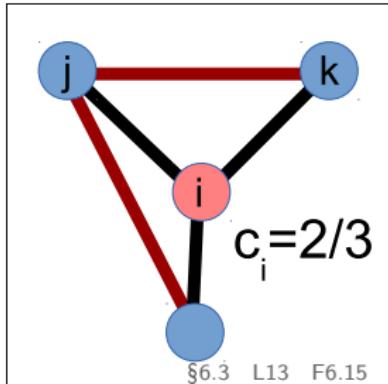
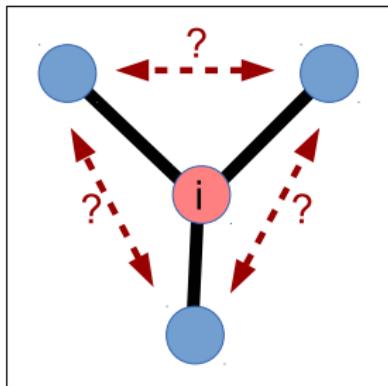
Vertex Clustering Coefficient

The CLUSTERING COEFFICIENT c_i of a vertex i is the fraction of complete triangles centred on i given number of possible triangles with nearest neighbours.

Clustering coefficient gives 1 if all possible triangles are present, 0 if there are no triangles

$$c_i = \frac{2}{k_i(k_i - 1)} \sum_{j < k} A_{ij}A_{ik}A_{jk} \text{ (if simple)} \quad (6.1)$$

Note: undefined if degree less than 2 (no triangles possible)



Vertex Clustering Coefficient

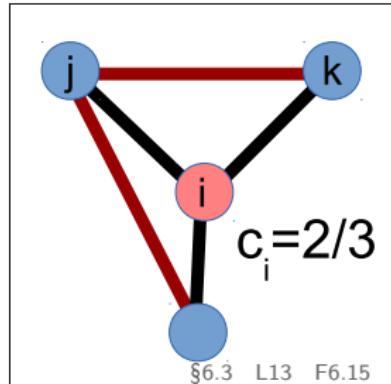
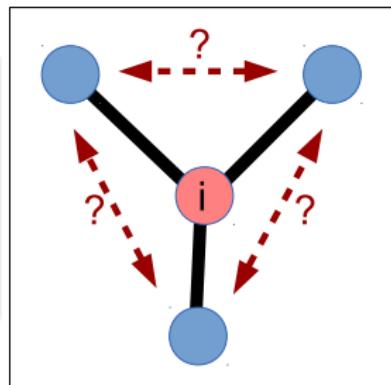
Vertex Clustering Coefficient

The CLUSTERING COEFFICIENT c_i of a vertex i is the fraction of complete triangles centred on i given number of possible triangles with nearest neighbours.

Clustering coefficient gives 1 if all possible triangles are present, 0 if there are no triangles

$$c_i = \frac{2}{k_i(k_i - 1)} \sum_{j < k} A_{ij}A_{ik}A_{jk} \text{ (if simple)} \quad (6.1)$$

Note: undefined if degree less than 2 (no triangles possible)



Network Clustering Coefficient I: c

Network Clustering Coefficient

The CLUSTERING COEFFICIENT c of a network is the average of the vertex clustering coefficients

$$c = \frac{1}{N} \sum_i c_i \quad (6.2)$$

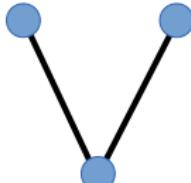
This is not defined if c_i poorly defined so there is a second definition of a network clustering coefficient, $c^{(II)}$. . .

Network Clustering Coefficient II: $c^{(II)}$

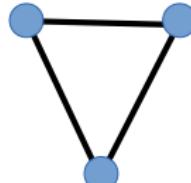
Network Clustering Coefficient II: $c^{(II)}$

A second definition of a CLUSTERING COEFFICIENT c for a network is the fraction of 'wedges' (two edges with a common vertex) which are part of triangles.

$$c^{(II)} = \frac{3(\text{number of triangles})}{(\text{number of wedges})} \quad (6.3)$$



A 'wedge'



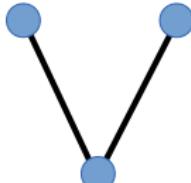
Every triangle contains three wedges

Network Clustering Coefficient II: $c^{(II)}$

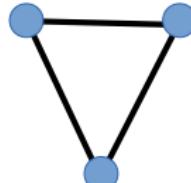
Network Clustering Coefficient II: $c^{(II)}$

A second definition of a CLUSTERING COEFFICIENT c for a network is the fraction of 'wedges' (two edges with a common vertex) which are part of triangles.

$$c^{(II)} = \frac{3(\text{number of triangles})}{(\text{number of wedges})} \quad (6.3)$$



A 'wedge'

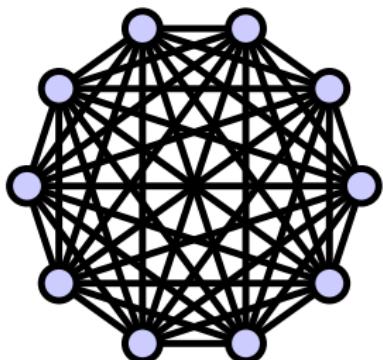


Every triangle contains three wedges

Clustering Coefficient Examples

- Complete graph $c = c_i = 1$
- Triangular lattice: per site have 6 triangles, $15 (= 6 \times 5/2)$ pairs of edges, $c = c_i = 0.4$
- ER Random graph
 $\langle c \rangle = p = \langle k \rangle / (N - 1)$
- Square lattice $c = c_i = 0$: want a square count measure here

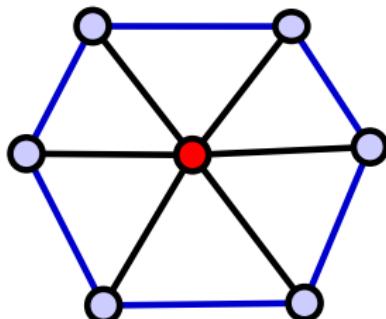
The definition should give 1 if all possible triangles are present (e.g. in a complete graph) and 0 if there are no triangles (e.g. square lattice).



Clustering Coefficient Examples

- Complete graph $c = c_i = 1$
- Triangular lattice: per site have 6 triangles, $15 (= 6 \times 5/2)$ pairs of edges, $c = c_i = 0.4$
- ER Random graph
 $\langle c \rangle = p = \langle k \rangle / (N - 1)$
- Square lattice $c = c_i = 0$: want a square count measure here

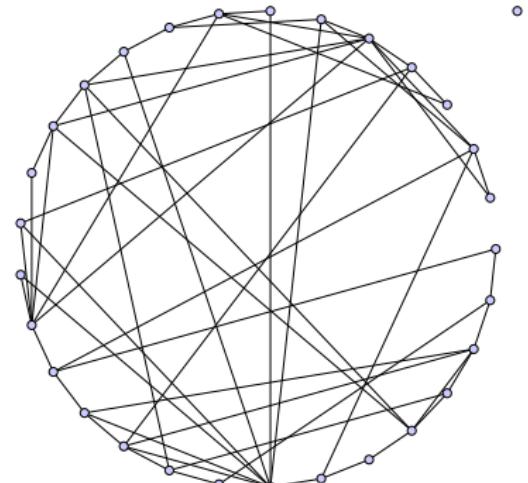
The definition should give 1 if all possible triangles are present (e.g. in a complete graph) and 0 if there are no triangles (e.g. square lattice).



Clustering Coefficient Examples

- Complete graph $c = c_i = 1$
- Triangular lattice: per site have 6 triangles, $15 (= 6 \times 5/2)$ pairs of edges, $c = c_i = 0.4$
- ER Random graph
 $\langle c \rangle = p = \langle k \rangle / (N - 1)$
- Square lattice $c = c_i = 0$: want a square count measure here

The definition should give 1 if all possible triangles are present (e.g. in a complete graph) and 0 if there are no triangles (e.g. square lattice).

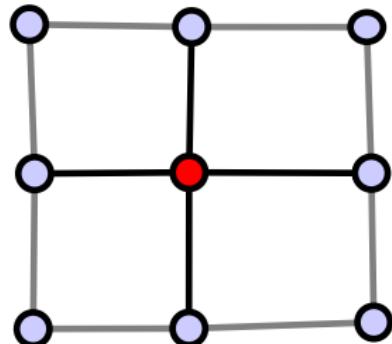


ER random graph, $N = 30$,
 $\langle k \rangle = 4.0$

Clustering Coefficient Examples

- Complete graph $c = c_i = 1$
- Triangular lattice: per site have 6 triangles, $15 (= 6 \times 5/2)$ pairs of edges, $c = c_i = 0.4$
- ER Random graph
 $\langle c \rangle = p = \langle k \rangle / (N - 1)$
- Square lattice $c = c_i = 0$: want a square count measure here

The definition should give 1 if all possible triangles are present (e.g. in a complete graph) and 0 if there are no triangles (e.g. square lattice).



Clearly highly clustered in terms of squares so triangle clustering c is not the full story here.

6. Network Measures without Eigenvalue Calculations

6.1 Recap

6.2 Betweenness

6.3 Clustering Coefficient

6.4 The Small World model

The Small World Model

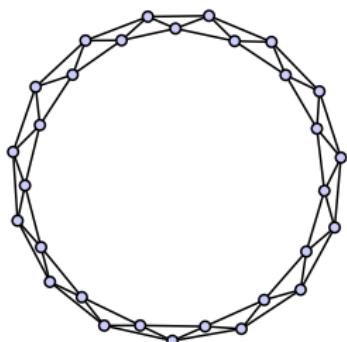
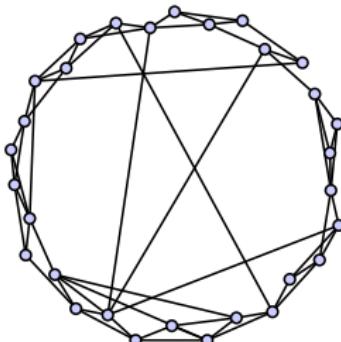
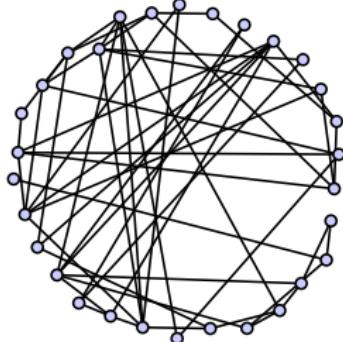
Null Model for Social Networks?

- Social networks are highly clustered ($c \sim 0.1$) and short path lengths ($\ell \sim \ln(N)$)
e.g. hep-th citation network has $c = 0.25$.
- Random networks have little clustering ($c \sim O(1/N)$) and short path lengths ($\ell \sim \ln(N)$)
- Regular lattices have high local structure ($c \sim 0.1$) but long path lengths ($\ell \sim N^{1/d}$)

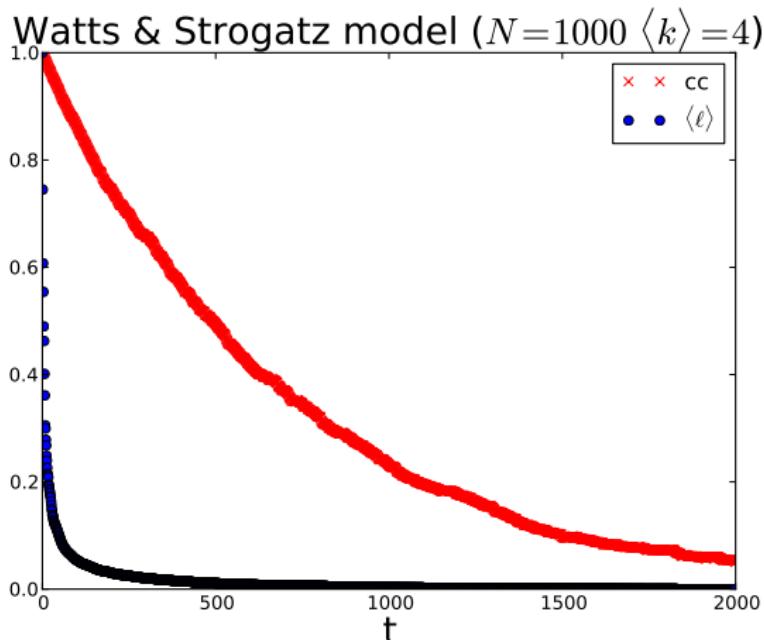
★ What would be a good null model for a social network?

Watts & Strogatz Small World model (1998)

- Start with a regular lattice ($c \sim 0.1$, $\ell \sim N^{1/d}$)
- Rewire t edges, move one end from one vertex to another
- After many rewirings ($\sim O(N)$) have a random graph ($c \sim O(1/N)$, $\ell \sim \ln(N)$)

 $t = 0$
 $c = 0.5, \ell = 4.1$  $t = 9$
 $c = 0.34, \ell = 3.0$  $t = 60$
 $c = 0.15, \ell = 2.7$

Watts & Strogatz Small World model



ℓ rescaled to be 1 at $t = 0$, and 0 at $t = \infty$.

Look close to origin for small-world region with high clustering like a social network

($t=2000$ = number of rewirings
= number of edges)

Network Measures without Eigenvalue Calculations

We have

- recalled that
 - ▶ degree k_i and degree distributions $p(k)$ are useful measures
 - ▶ paths are used to define connectedness and components
 - ▶ shortest paths ℓ_{ij} are used to define average shortest path length $\langle \ell \rangle$, diameter
- defined vertex and edge betweenness using shortest paths
- discussed Granovetter's idea of the strength of weak ties
- defined the clustering coefficient for vertices, and (in two ways) for networks
- discussed Watts & Strogatz' Small World model.

C & N Part II: End of Networks Lecture 13

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “Network Measures without Eigenvalue Calculations”

Intended Learning Outcomes

- Students will know how to calculate various types of betweenness and clustering coefficients.
- Students can interpret these measures in term of social science concepts.
- Students will be able to define the Watts-Strogatz small world model.

7. Processes on Networks

7.1 Introduction

7.2 Eigenvectors and Eigenvalues

7.3 Perron-Frobenius Theorem

7.4 Broadcasting & Centrality measures

7.5 Diffusion & PageRank Centrality

7.6 Summary

C & N Part II: Networks Lecture 14

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

Generalised Eigenvector & Eigenvalues

Intended Learning Outcomes

- Students can work with right- and left-eigenvectors and eigenvalues of non-hermitian matrices.
- Students can state the key results of the Peron-Frobenius theorem for non-negative matrices.

PS= See problem sheet, **EFS**= Exercise For Students

7. Processes on Networks

7.1 Introduction

7.2 Eigenvectors and Eigenvalues

7.3 Perron-Frobenius Theorem

7.4 Broadcasting & Centrality measures

7.5 Diffusion & PageRank Centrality

7.6 Summary

Processes on Networks

★ I will use the convention that A_{ji} is the weight of the edge ★
from source i to j . Only matters for directed networks.



Centrality measures

We have come across a couple of ways to assess the importance or, as it is called in social network analysis, the CENTRALITY of a vertex. However they have drawbacks

degree The more popular you are, the more important you are.

A reasonable starting point but has no information about the wider structure of the network,

e.g. perhaps you are popular in a small but remote community so the rest of the world barely knows you exist.

betweenness The shortest paths see the whole network structure.

The shortest paths are rarely the only important paths

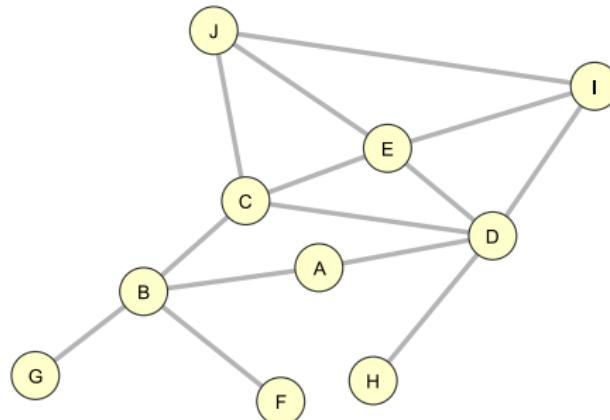
Many paths are of similar distance, some popular paths become congested forcing us to seek alternatives, we often do not know the shortest/cheapest route.

⇒ Look for measures which use all the links and routes in the network.

Centrality Quiz

menti.com, <https://www.menti.com/9f8280>

Which vertex is the most important?



(updated 5/2/20)

Processes on Networks

- Networks are not an end in themselves.
- Networks are invariably just the substrate for other dynamical processes
 - e.g. friends spreading rumours on a social network.
 - e.g. pattern of web pages views as person ‘surfs’ the web
- If we can model these processes we can produce a more ‘natural’ set of network measures,
 - e.g. of the centrality/importance of vertices.
- Even if our models are not so realistic, they may provide an alternative/better null model
 - e.g. we do not always know or use the shortest path used in betweenness.

Local Processes on Networks

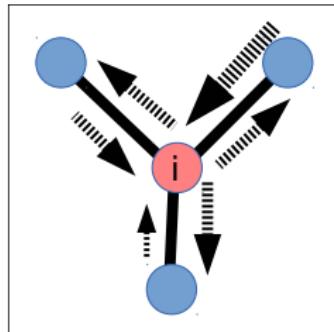
We will consider **local processes** on a network.

- Think of an agent at each vertex and all they know is who their neighbours are. They do not even know how big the network is.
- At each time step in the process the agent can decide to pass on information, or not, to some or all of its neighbours according to some simple rules based only on local information.

Two main types of process

- broadcast
- diffusion

Both can be analysed with linear algebra
(eigenvectors etc)



★ We first use such processes to define further centrality measures.

7. Processes on Networks

7.1 Introduction

7.2 Eigenvectors and Eigenvalues

7.3 Perron-Frobenius Theorem

7.4 Broadcasting & Centrality measures

7.5 Diffusion & PageRank Centrality

7.6 Summary

Eigenvectors and Eigenvalues

Eigenvectors and Eigenvalues

We will first set out the key results for Eigenvectors and Eigenvalues.

★ I will assume there is a path from every vertex to every other ★ vertex, in both directions, i.e. network is strongly connected.

- Most results hold for general matrices with some appropriate alterations.
- Proofs for more general matrices are much more involved.

Eigenvector properties

(Right) Eigenvector equation for general matrix M

$$\lambda_n v_i^{(n)} = \sum_j M_{ij} v_j^{(n)} \quad \lambda_n v^{(n)} = M v^{(n)} \quad (7.1)$$

Left Eigenvector equation

$$\lambda_n u_j^{(n)} = \sum_i u_i^{(n)} M_{ij} \quad \lambda_n (u^{(n)})^T = (u^{(n)})^T M \quad (7.2)$$

- ① The set of left- and right-eigenvalues are identical.
- ② The left- and right-eigenvectors need not be the same.
- ③ The left- and right-eigenvectors can be chosen to be orthogonal.
$$(u^{(n)})^T \cdot v^{(m)} = \delta_{nm}$$

Proof of these results is optional but is on a problem sheet. **PS**

Diagonalisation

Define matrix V with columns such that

- The j -th column of V is the j -th right-eigenvector $v^{(j)}$, $V_{ij} = v_i^{(j)}$.

$$V = \begin{pmatrix} v_1^{(1)} & v_1^{(2)} & \dots & v_1^{(j)} & \dots \\ v_2^{(1)} & v_2^{(2)} & \dots & v_2^{(j)} & \dots \\ v_3^{(1)} & v_3^{(2)} & \dots & v_3^{(j)} & \dots \\ \vdots & \vdots & \dots & \vdots & \dots \\ v_i^{(1)} & v_i^{(2)} & \dots & v_i^{(j)} & \dots \\ \vdots & \vdots & \dots & \vdots & \dots \end{pmatrix} \quad (7.3)$$

Diagonalisation

Define second matrix U such that

- The j -th column of U is the j -th left-eigenvector $u^{(j)}$, $U_{ij} = u_i^{(j)}$.

$$U = \begin{pmatrix} u_1^{(1)} & u_1^{(2)} & \cdots & u_1^{(j)} & \cdots \\ u_2^{(1)} & u_2^{(2)} & \cdots & u_2^{(j)} & \cdots \\ u_3^{(1)} & u_3^{(2)} & \cdots & u_3^{(j)} & \cdots \\ \vdots & \vdots & \cdots & \vdots & \cdots \\ u_i^{(1)} & u_i^{(2)} & \cdots & u_i^{(j)} & \cdots \\ \vdots & \vdots & \cdots & \vdots & \cdots \end{pmatrix} \quad (7.4)$$

Diagonalisation

Define matrices U and V such that

- The i -th row of U^T is the i -th left-eigenvector $(u^{(i)})^T$, $U_{ji} = u_j^{(i)}$.
- The j -th column of V is the j -th right-eigenvector $v^{(j)}$, $V_{ij} = v_i^{(j)}$.

Orthogonality of the left-/right-eigenvectors means

$$U^T \cdot V = \mathbb{1}, \quad (V)^{-1} = U^T. \quad (7.5)$$

$$U^T V = \begin{pmatrix} u_1^{(1)} & u_2^{(1)} & \cdots & u_k^{(1)} & \cdots \\ u_1^{(2)} & u_2^{(2)} & \cdots & u_k^{(2)} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ u_1^{(i)} & u_2^{(i)} & \cdots & u_k^{(i)} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} v_1^{(1)} & v_1^{(2)} & \cdots & v_1^{(j)} & \cdots \\ v_2^{(1)} & v_2^{(2)} & \cdots & v_2^{(j)} & \cdots \\ \vdots & \vdots & \cdots & \vdots & \cdots \\ v_k^{(1)} & v_k^{(2)} & \cdots & v_k^{(j)} & \cdots \\ \vdots & \vdots & \cdots & \vdots & \cdots \end{pmatrix} \quad (7.6)$$

Eigenvalue Equations as Matrices

The right-eigenvector equation can be written as the matrix equation

$$MV = V\Lambda, \quad (7.7)$$

and the left-eigenvector equation can be written as the matrix equation

$$U^T \cdot M = \Lambda U^T \quad (7.8)$$

where

- The i -th row of U^T is the i -th left-eigenvector $(u^{(i)})^T$, $U_{ji} = u_j^{(i)}$.
- The j -th column of V is the j -th right-eigenvector $v^{(j)}$, $V_{ij} = v_i^{(j)}$.
- Λ is a diagonal matrix with eigenvalue λ_i as the i -the entry,
 $\Lambda_{ij} = \lambda_i \delta_{ij}$.

Just see how multiplying one row with one column in MV and $U^T \cdot M$ can be understood from the definition of left- and right-eigenvectors, then check the right-hand side gives the correct answer in terms of an eigenvalue and the relevant eigenvector.

Diagonalisation

Pre-multiply (7.8) by U^T , or post-multiply (7.8) by V ,
Then use orthogonality of eigenvectors in (7.5),
Find

$$U^T \cdot M V = \Lambda, \quad M = (U^T)^{-1} \Lambda V^{-1} \quad (7.9)$$

where

- The i -th row of U^T is the i -th left-eigenvector $(u^{(i)})^T$, $U_{ji} = u_j^{(i)}$.
- The j -th column of V is the j -th right-eigenvector $v^{(j)}$, $V_{ij} = v_i^{(j)}$.
- Λ is a diagonal matrix with eigenvalue λ_i as the i -the entry,
 $\Lambda_{ij} = \lambda_i \delta_{ij}$.

Diagonalisation

So the orthogonality of the left-/right-eigenvectors means

$$U^T \cdot V = \mathbb{1}, \quad (V)^{-1} = U^T, \quad M = V \Lambda U^T = V \Lambda V^{-1} \quad (7.10)$$

where

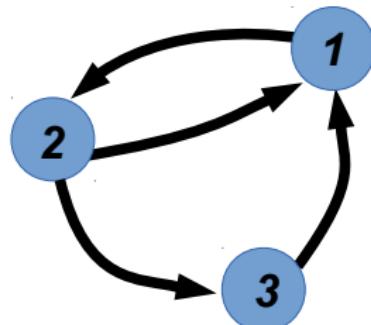
- The i -th row of U^T is the i -th left-eigenvector $(u^{(i)})^T$, $U_{ji} = u_j^{(i)}$.
- The j -th column of V is the j -th right-eigenvector $v^{(j)}$, $V_{ij} = v_i^{(j)}$.
- Λ is a diagonal matrix with eigenvalue λ_i as the i -the entry, $\Lambda_{ij} = \lambda_i \delta_{ij}$.

This will always work on a single strongly connected network. Formally, you need the eigenvectors to span the space.

Simple example

Eigenvalues and Eigenvectors of the adjacency matrix

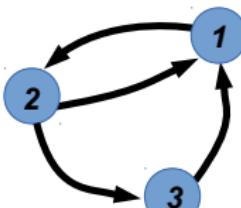
$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$



n	0	1	2
λ	1.32	-0.66	-0.66
$v_1^{(n)}$	-0.55	$0.43 - 0.37i$	$0.43 + 0.37i$
$v_2^{(n)}$	-0.73	$-0.08 + 0.49i$	$-0.08 - 0.49i$
$v_3^{(n)}$	-0.41	-0.66	-0.66
$u_1^{(n)}$	-0.73	$0.08 - 0.49i$	$0.08 + 0.49i$
$u_2^{(n)}$	-0.55	$-0.43 + 0.37i$	$-0.43 - 0.37i$
$u_3^{(n)}$	-0.41	0.66	0.66

Simple example

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad (7.11)$$



$$= \begin{pmatrix} -0.55 & 0.43 - 0.37i & 0.43 + 0.37i \\ -0.73 & -0.08 + 0.49i & -0.08 - 0.49i \\ -0.41 & -0.66 & -0.66 \end{pmatrix} \quad (7.12)$$

$$\times \begin{pmatrix} 1.32 & 0 & 0 \\ 0 & -0.66 & 0 \\ 0 & 0 & -0.66 \end{pmatrix} \quad (7.13)$$

$$\times \begin{pmatrix} -0.73 & -0.55 & -0.41 \\ 0.08 - 0.49i & -0.43 + 0.37i & 0.66 \\ 0.08 + 0.49i & -0.43 - 0.37i & 0.66 \end{pmatrix} \quad (7.14)$$

Diagonalisation issues

Our eigenvectors (both the left and right sets independently) may not span the vector space, we may have less than N . In these situations we can still follow the programme set out below but it takes a lot more effort

- Use JORDAN NORMAL FORMS.
- Add a small (positive) but different term to every entry which will create a better behaved matrix

Functions of Matrices

Functions of Matrices

The function f of a matrix M is defined using the Taylor series of f .

$$f(M) = \sum_{n=0}^{\infty} \frac{c^{(n)}}{n!} M^n, \quad c^{(n)} = \left. \frac{d^n f}{dx^n} \right|_{x=0} \quad (7.15)$$

Examples:

$$e^M = \sum_{n=0}^{\infty} \frac{1}{n!} M^n \quad (7.16)$$

$$\frac{1}{\mathbb{1} - M} = (\mathbb{1} - M)^{-1} = \sum_{n=0}^{\infty} M^n \quad (7.17)$$

Properties of Exponentials of Matrices

- ★ Functions of matrices have **some** but rarely all the properties ★
of the same function of a number.

$$e^A e^B = e^{A+B} \text{ iff } [A, B] = 0 \quad (7.18)$$

$$e^A e^{-A} = e^{A-A} = e^0 = \mathbb{1} \quad (7.19)$$

Thus the inverse matrix of the matrix $\exp\{A\}$ is what the notation suggests

$$(e^A)^{-1} = e^{-A} \quad (7.20)$$

Must check all manipulations of function f to see if they still work.
Always go back to Taylor expansion definition to do this.

Sum of All Powers of Matrices

★ Functions of matrices have **some** but rarely all the properties ★
of the same function of a number.

$$(\mathbb{1} - M) \cdot (\mathbb{1} - M)^{-1} = \mathbb{1} \quad (7.21)$$

where the inverse of $(\mathbb{1} - M)$ is written as $(\mathbb{1} - M)^{-1}$.
This notation suggests

$$(\mathbb{1} - M)^{-1} = \frac{1}{\mathbb{1} - M} = \sum_{n=0}^{\infty} (M)^n \quad (7.22)$$

Are these uses of $(\dots)^{-1}$ notation consistent?

In this case YES — use Taylor expansion definition to show this.

Functions of Matrices and Diagonalised Form

Since $A = V \Lambda V^{-1}$ you can show that (PS)

$$f(A) = V f(\Lambda) V^{-1}, \quad (7.23)$$

$$f(\Lambda) = \begin{pmatrix} f(\lambda_1) & 0 & 0 & \dots \\ 0 & f(\lambda_2) & 0 & \dots \\ 0 & 0 & f(\lambda_3) & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (7.24)$$

Use this to show convergence of these Matrix function definitions depends on convergence of Taylor series in terms of the different eigenvalues.

7. Processes on Networks

7.1 Introduction

7.2 Eigenvectors and Eigenvalues

7.3 Perron-Frobenius Theorem

7.4 Broadcasting & Centrality measures

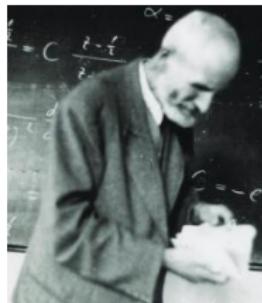
7.5 Diffusion & PageRank Centrality

7.6 Summary

Perron-Frobenius Theorem (1907,1912)

Perron-Frobenius Theorem (1907,1912)

Oskar Perron (1880-1975) in 1907 for
POSITIVE MATRICES $A_{ij} > 0$.



Ferdinand Georg Frobenius (1849-1917)
extended this in 1912 to
NON-NEGATIVE MATRICES $A_{ij} \geq 0$
which are also IRREDUCIBLE
(definition below).



Perron-Frobenius Theorem (1907,1912)

Consider a real $N \times N$ non-negative irreducible matrix A.

Real All entries are real $A_{ij} \in \mathbb{R}$

Non-negative All entries are 0 or positive $A_{ij} > 0$

Irreducible Can **not** rearrange rows and columns to produce a block upper-triangular form i.e. if we have square blocks on the diagonal, there is an off-diagonal block of zeros

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \\ 0 & 0 & 0 & A_{44} & A_{45} \\ 0 & 0 & 0 & A_{54} & A_{55} \end{pmatrix} \text{ is } \mathbf{\text{reducible}} \quad (7.25)$$

★ A strongly connected network \Rightarrow Irreducible adjacency matrix ★

Perron-Frobenius Theorem: Largest

Let M be a real $N \times N$ non-negative irreducible matrix with eigenvalues λ_i ($i = 1, 2, \dots, N$) labelled such that $|\lambda_i| \geq |\lambda_{i+1}|$. Such a matrix M has the following properties:-

- ① There is a **unique real** eigenvalue, λ_1 , with modulus equal to the largest of all eigenvalues,

$$\lambda_1 \neq \lambda_j, \quad \lambda_1 \geq |\lambda_j| \quad \forall j \neq 1. \quad (7.26)$$

- ▶ For **positive** matrices $\lambda_1 > |\lambda_j|$ (Perron's case)
- ▶ If $\lambda_1 > |\lambda_j|$ the matrix is called PRIMITIVE.
We will assume this is true. Only common case where this may be an issue for strongly connected networks is for a bipartite network.

Perron-Frobenius Theorem: Largest

Let M be a real $N \times N$ non-negative irreducible matrix with eigenvalues λ_i ($i = 1, 2, \dots, N$) labelled such that $|\lambda_i| \geq |\lambda_{i+1}|$. Let $v^{(i)}$ be the eigenvector with eigenvalue λ_i . Such a matrix M has the following properties:-

- ② The PERRON VECTOR $p = v^{(1)}$ is the eigenvector associated with this eigenvalue λ_1 and it is the **only** eigenvector with the same sign for all entries, w.l.o.g. $p_i = v_i^{(1)} > 0 \quad \forall i$.
- ③ The largest eigenvalue λ_1 is bounded by the sums of the rows,

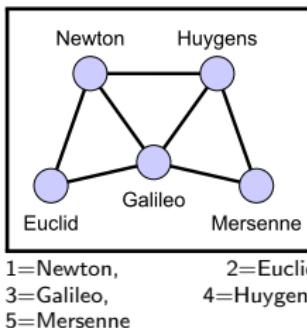
$$\min_i \left\{ \sum_j M_{ij} \right\} \leq \lambda_1 \leq \max_i \left\{ \sum_j M_{ij} \right\} \quad (7.27)$$

and by the sums of the columns

$$\min_j \left\{ \sum_i M_{ij} \right\} \leq \lambda_1 \leq \max_j \left\{ \sum_i M_{ij} \right\} \quad (7.28)$$

Perron-Frobenius on 5 Mathematicians

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$



Simple network of 5 mathematicians from MacTutor website.

Row/column sums
 $\Rightarrow 2.0 \leq \lambda_1 \leq 4.0$

n	1	2	3	4	5
λ	2.94	0.62	-0.46	-1.62	-1.47
$v_1^{(n)}$	-0.47	-0.37	0.51	0.60	0.14
$v_2^{(n)}$	-0.35	-0.60	-0.44	-0.37	0.43
$v_3^{(n)}$	-0.56	-0.00	-0.31	-0.00	-0.77
$v_4^{(n)}$	-0.47	0.37	0.51	-0.60	0.14
$v_5^{(n)}$	-0.35	0.60	-0.44	0.37	0.43

Example of Perron-Frobenius in action

Example (test1 in PerronFrobeniusExample.py)

$$A = \begin{pmatrix} 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 \\ 1 & 4 & 2 & 4 \\ 3 & 1 & 3 & 2 \end{pmatrix} \quad \begin{array}{l} \text{Row/Column sums give} \\ 8.0 \leq \lambda_1 \leq 12.0 \end{array} \quad (7.29)$$

n	0	1	2	3
λ	10.16	-1.85	0.45	0.24
$v_0^{(n)}$	0.29	0.26	-0.18	0.18
$v_1^{(n)}$	0.67	0.02	-0.68	-0.56
$v_2^{(n)}$	0.56	0.65	0.04	-0.40
$v_3^{(n)}$	0.39	-0.71	0.71	0.70

Example of Perron-Frobenius in action

Example (test2 in PerronFrobeniusExample.py)

$$A = \begin{pmatrix} 1 & 2 & -1 & -2 \\ 3 & -4 & 3 & 4 \\ 1 & 4 & -2 & -4 \\ 3 & 1 & 3 & 2 \end{pmatrix} \quad \begin{array}{l} \text{Row/Column sums give} \\ 0.0 \leq \lambda_1 \leq 8.0 \end{array} \quad (7.30)$$

n	0	1	2	3
λ	$1.41 + 2.10i$	$1.41 - 2.10i$	0.16	-5.98
$v_0^{(n)}$	-0.09+0.19i	-0.09-0.19i	-0.15	0.25
$v_1^{(n)}$	0.45 + 0.11i	0.45 - 0.11i	-0.28	-0.79
$v_2^{(n)}$	-0.21+0.31i	-0.21-0.31i	0.66	0.53
$v_3^{(n)}$	0.77	0.77	-0.68	-0.19

C & N Part II: End of Networks Lecture 14

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “Generalised Eigenvector & Eigenvalues”

Intended Learning Outcomes

- Students can work with right- and left-eigenvectors and eigenvalues of non-hermitian matrices.
- Students can state the key results of the Peron-Frobenius theorem for non-negative matrices.

C & N Part II: Networks Lecture 15

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

Broadcasting & Centrality measures

Intended Learning Outcomes

- Students can describe a broadcasting process on a network.
- Students can derive expressions for Eigenvalue and Katz centrality measures in terms of the adjacency matrix of a network.

PS= See problem sheet, **EFS**= Exercise For Students

7. Processes on Networks

7.1 Introduction

7.2 Eigenvectors and Eigenvalues

7.3 Perron-Frobenius Theorem

7.4 Broadcasting & Centrality measures

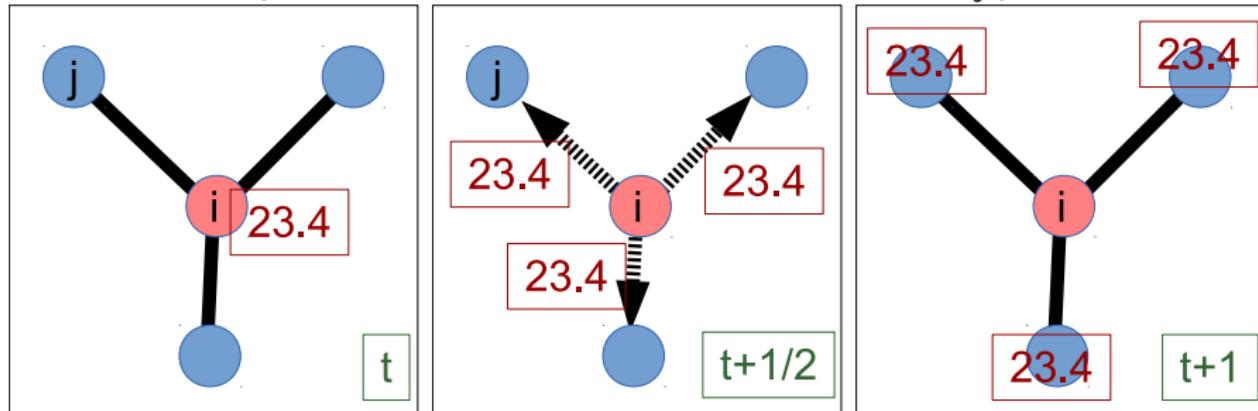
7.5 Diffusion & PageRank Centrality

7.6 Summary

Broadcasting & Centrality measures

Broadcast processes

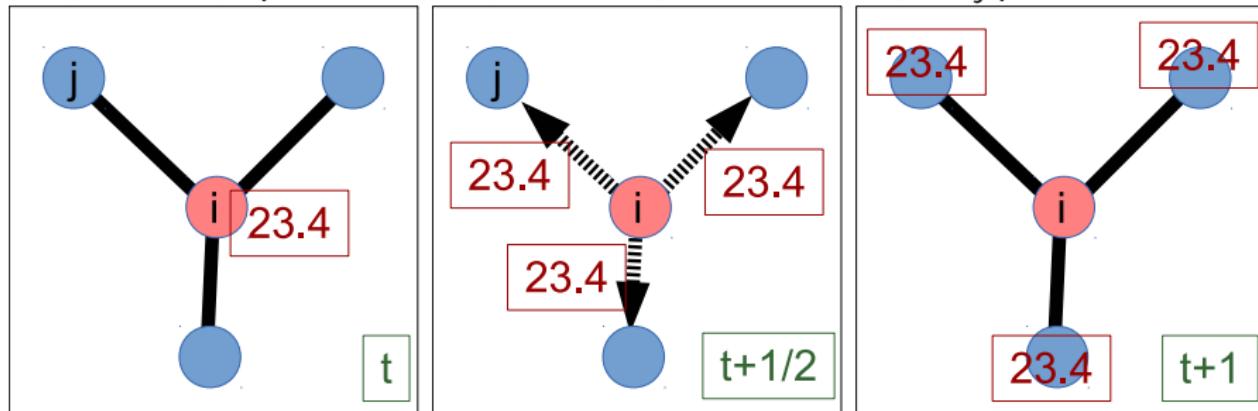
Each node i copies or **broadcasts** its current 'value' to **all** its neighbours j (multiplied by the weight of the edge A_{ji})



EIGENVALUE CENTRALITY and KATZ CENTRALITY are based on the behaviour of a **broadcast process** on the network.

Broadcast processes

Each node i copies or **broadcasts** its current 'value' to **all** its neighbours j (multiplied by the weight of the edge A_{ji})



EIGENVALUE CENTRALITY and KATZ CENTRALITY are based on the behaviour of a **broadcast process** on the network.

Eigenvalue centrality

Eigenvalue Centrality

The Eigenvalue Centrality of vertex i is the i -th entry in the Perron vector $p_i = v_i^{(1)}$ of the adjacency matrix

From the Perron-Frobenius Theorem:-

- ② The PERRON VECTOR $p = v^{(1)}$ is the eigenvector associated with the largest eigenvalue λ_1 and it is the **only** eigenvector with the same sign for all entries, w.l.o.g.

$$p_i = v_i^{(1)} > 0 \quad \forall i . \quad (7.31)$$

Eigenvalue centrality

Justification:

- ① Each vertex j starts with centrality value $w(0)_j$ at step $t = 0$
- ② They **broadcast** this value to all their neighbours (weighted by the edge weight)
- ③ The neighbours sum up all these ‘votes’ to get their new centrality value for next step $t + 1$.

Thus strong links to important vertices give big effect, but weak links to unimportant vertices low contributions.

$$w(t+1)_i = \sum_j A_{ij} w(t)_j \quad (7.32)$$

- ④ Repeat for long time $t \rightarrow \infty$.
- ⑤ Final centrality value of i -th vertex, $w(\infty)_i$, is proportional to p_i

Eigenvalue centrality proof

Decompose initial vector in terms of eigenvectors

$$w(t=0)_j = \sum_n c_n v_j^{(n)} \quad (7.33)$$

Then find (e.g. by induction) that

$$w(t)_i = \sum_j [\mathbf{A}^t]_{ij} w(t=0)_i = \sum_n c_n (\lambda_n)^t v_i^{(n)} \quad (7.34)$$

Perron vector term dominates since it has the largest eigenvalue,
 $(|\lambda_n|/\lambda_1)^t \rightarrow 0$, so

$$\lim_{t \rightarrow \infty} w(t)_i \propto p_i \quad (7.35)$$

Eigenvalue Centrality is given by the Perron vector

Path Interpretation of Eigenvalue centrality

We mentioned that if $A_{ij} \in \{0, 1\}$, we have that $[A^t]_{ij}$ is the total number of walks of length t from vertex j to vertex i .

$$w(t) = Aw(t-1) = AAw(t-2) = (A)^t w(t=0) \quad (7.36)$$

Thus Eigenvalue centrality is proportional to the number of very long walks ($t \rightarrow \infty$) passing through a vertex.

Problems with Eigenvalue Centrality

- Not all $N \times N$ matrices have N linearly independent eigenvectors, e.g. projection or nilpotent matrices,
- In terms of networks this corresponds nodes that are unreachable from some starting points or can not reach all other vertices. e.g. a web site which is not linked from any other site or web sites with no hyperlinks on their pages.

One solution:- give a minimum value for centrality, β to each vertex

$$w(t+1)_i = \beta + \alpha \sum_j A_{ij} w(t)_j \quad (7.37)$$

This leads us to KATZ CENTRALITY

Katz Centrality

Katz Centrality (1953)

The KATZ CENTRALITY K_i of vertex i is proportional to the long-time limit of the following process

$$K_i = \lim_{t \rightarrow \infty} w(t)_i, \quad w(t)_i = \beta + \alpha \sum_j A_{ij} w(t-1)_j \quad (7.38)$$

- β term means that this is well defined for *any* network if α is small enough.
- β represents the enforcement of a minimum value for Katz centrality.

Katz Centrality solution

$$\mathbf{w}(t) = \beta \mathbf{1} + \alpha \mathbf{A} \mathbf{w}(t-1) \quad (7.39)$$

$$= \beta \mathbf{1} + \alpha \mathbf{A} (\beta \mathbf{1} + \alpha \mathbf{A} \mathbf{w}(t-2)) \quad (7.40)$$

$$= \beta \mathbf{1} + \beta \alpha \mathbf{A} \mathbf{1} + (\alpha)^2 \mathbf{A} \mathbf{A} \mathbf{w}(t-2) \quad (7.41)$$

$$= \beta \mathbf{1} + \beta \alpha \mathbf{A} \mathbf{1} + \beta (\alpha \mathbf{A})^2 \mathbf{1} + \dots \\ \dots + \beta (\alpha \mathbf{A})^{t-1} + (\alpha \mathbf{A})^t \mathbf{w}(0) \quad (7.42)$$

where $\mathbf{1}$ is a vector of 1's, $\mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{pmatrix}$

$$(7.43)$$

Katz Centrality solution

$$w(t) = \beta 1 + \beta \alpha A 1 + \beta (\alpha A)^2 1 + \dots + (\alpha A)^t w(0) \quad (7.44)$$

If α is small enough ($|\alpha A_{ij}| < 1 \forall i, j$ is sufficient),

$$\lim_{t \rightarrow \infty} (\alpha A)^t w(0) = 0 \Rightarrow \text{initial conditions don't matter.} \quad (7.45)$$

\Rightarrow Leaving (β is now just an irrelevant rescaling)

$$K \propto \sum_{t=0}^{\infty} (\alpha A)^t 1 = \frac{1}{\mathbb{1} - \alpha A} 1 = (\mathbb{1} - \alpha A)^{-1} 1 \quad (7.46)$$

Katz Centrality for Simple Graphs

$$K \propto \sum_{t=0}^{\infty} (\alpha A)^t \mathbf{1} = \frac{1}{\mathbb{1} - \alpha A} \mathbf{1} = (\mathbb{1} - \alpha A)^{-1} \mathbf{1} \quad (7.47)$$

Interpretation

- If $A_{ij} \in \{0, 1\}$ (e.g. simple graphs) then $(A^t)_{ij}$ is number of walks (cycles allowed) from j to i of length t .
- So the Katz centrality of vertex i is proportional to the number of walks which end at i from *any* starting vertex, where we **weight** walks by α^t for a walk of length t .

Katz Centrality convergence

If we use $A = V\Lambda V^{-1}$ then $(A)^n = V(\Lambda)^n V^{-1}$ so

$$K \propto V \left(\sum_{t=0}^{\infty} (\alpha \Lambda)^t \right) V^{-1} \mathbf{1} \quad (7.48)$$

$$= V \begin{pmatrix} (1 - \alpha \lambda_1)^{-1} & 0 & 0 & \cdots \\ 0 & (1 - \alpha \lambda_2)^{-1} & 0 & \cdots \\ 0 & 0 & (1 - \alpha \lambda_3)^{-1} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} V^{-1} \quad (7.49)$$

provided $|\alpha \lambda_n| < 1$ for convergence of sums. Thus we have

$$|\alpha| < \frac{1}{\lambda_1}. \quad (7.50)$$

Katz Centrality solution

$$K \propto \sum_{t=0}^{\infty} (\alpha A)^t \mathbf{1} = \frac{1}{\mathbb{1} - \alpha A} \mathbf{1} = (\mathbb{1} - \alpha A)^{-1} \mathbf{1} \quad (7.51)$$

- Valid if

$$|\alpha| < \frac{1}{\lambda_1}. \quad (7.52)$$

- Only relative values of Katz centrality have any meaning, only K_i/K_j is important.
 $\Rightarrow \beta$ drops out (as long as β is non-zero).
- For simple graphs, Katz Centrality counts the number of walks of length ℓ where each walk is weighted by $(\alpha)^\ell$

C & N Part II: End of Networks Lecture 15

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “Broadcasting & Centrality measures”

Intended Learning Outcomes

- Students can describe a broadcasting process on a network.
- Students can derive expressions for Eigenvalue and Katz centrality measures in terms of the adjacency matrix of a network.

C & N Part II: Networks Lecture 16

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

Diffusion & Centrality measures

Intended Learning Outcomes

- Students can describe a diffusion process on a network.
- Students can derive expressions for the PageRank centrality measure in terms of the adjacency matrix of a network.

PS= See problem sheet, **EFS**= Exercise For Students

7. Processes on Networks

7.1 Introduction

7.2 Eigenvectors and Eigenvalues

7.3 Perron-Frobenius Theorem

7.4 Broadcasting & Centrality measures

7.5 Diffusion & PageRank Centrality

7.6 Summary

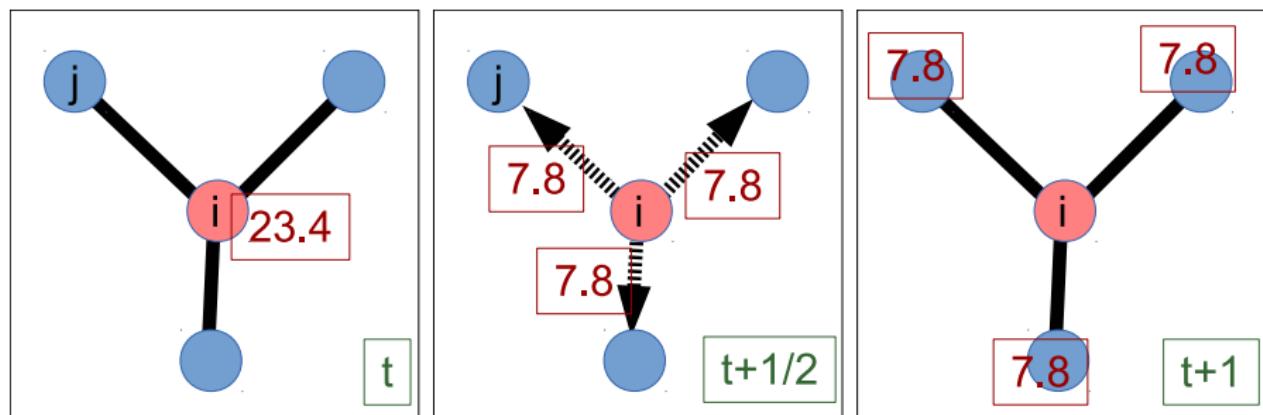
Diffusion & PageRank Centrality

Diffusion processes

Instead, could consider a **diffusion process** on the network.

Each node i divides its current 'value' between its neighbours j in proportion to the edge weights A_{ji}

Flow of 'particles' divides in proportion to edge weights in general.



The Transfer Matrix for Diffusion

The Transfer Matrix for Diffusion

The diffusion process is described by the TRANSFER MATRIX T , where T_{ji} describes the fraction of the value from source vertex i passed onto one neighbour j .

$$w(t+1)_j = \sum_i T_{ji} w(t)_i \quad (7.53)$$

where

$$T_{ji} = \frac{1}{s_i^{(\text{out})}} A_{ji}$$

$$s_i^{(\text{out})} = \sum_j A_{ji} \quad (= k^{(\text{out})} \text{ if } A_{ij} \in \{0, 1\})$$

(7.54)

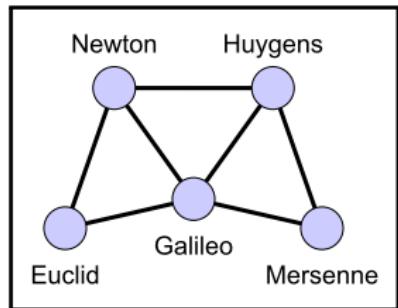


A_{ji} is strength of connection from source i to target j

Example Transfer Matrix: 5 Mathematicians

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \quad (7.55)$$

$$T = \begin{pmatrix} 0 & 1/2 & 1/4 & 1/3 & 0 \\ 1/3 & 0 & 1/4 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/3 & 1/2 \\ 1/3 & 0 & 1/4 & 0 & 1/2 \\ 0 & 0 & 1/4 & 1/3 & 0 \end{pmatrix} \quad (7.56)$$



1=Newton, 2=Euclid, 3=Galileo, 4=Huygens,
5=Mersenne

★ Columns all sum to one

Properties of the Transfer Matrix (Proofs in PS)

- Columns sum to one, $\sum_j T_{ji} = 1$ for any i .
- Largest eigenvalue is 1, $\lambda_1 = 1$.
- The left eigenvector associated with the largest eigenvalue is always proportional to 1

$$u_i^{(1)} \propto 1. \quad (7.57)$$

- All the other right eigenvectors of T , $v^{(n)}$ for $n \geq 2$, have entries which sum to zero, i.e.

$$\sum_i v_i^{(n)} = 0, \quad (n \geq 2). \quad (7.58)$$

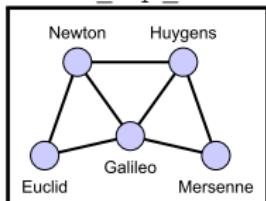
- For **undirected** network only ($A = A^T$) the right eigenvector of the largest eigenvalue of T is proportional to strength

$$v_i^{(1)} \propto s_i = \sum_j A_{ij} \quad (= k_i \text{ degree if unweighted}). \quad (7.59)$$

Example Transfer Matrix: 5 Mathematicians

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \quad T = \begin{pmatrix} 0 & 1/2 & 1/4 & 1/3 & 0 \\ 1/3 & 0 & 1/4 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/3 & 1/2 \\ 1/3 & 0 & 1/4 & 0 & 1/2 \\ 0 & 0 & 1/4 & 1/3 & 0 \end{pmatrix}$$

Row/column sums
 $\Rightarrow 1.00 \leq \lambda_1 \leq 1.00$



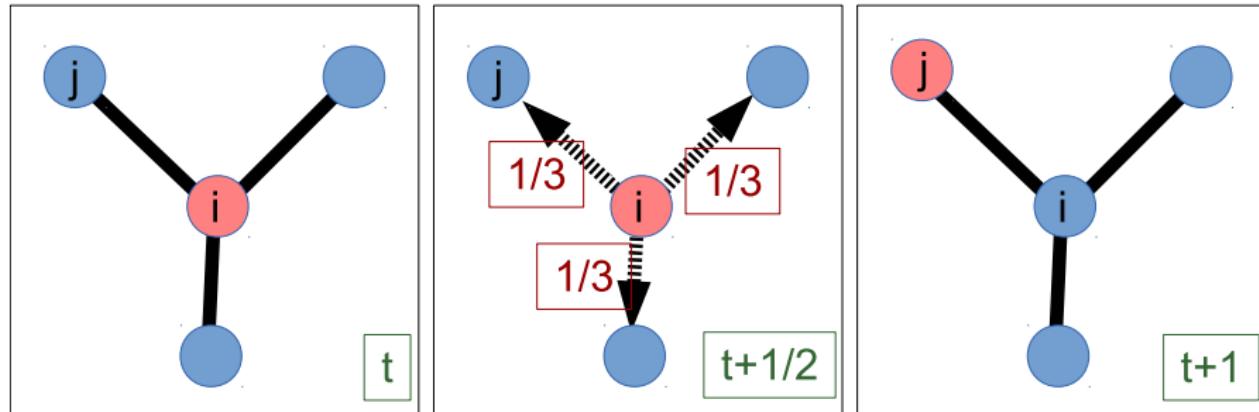
1=Newton,
 3=Galileo,
 5=Mersenne

2=Euclid,
 4=Huygens,

n	1	2	3	4	5
λ	1.00	0.27	-0.17	-0.61	-0.50
$v_1^{(n)}$	-0.46	-0.45	-0.53	-0.62	-0.00
$v_2^{(n)}$	-0.31	-0.55	0.27	0.34	0.41
$v_3^{(n)}$	-0.62	-0.00	0.53	0.00	-0.82
$v_4^{(n)}$	-0.46	0.45	-0.53	0.62	0.00
$v_5^{(n)}$	-0.31	0.55	0.27	-0.34	0.41

The Transfer Matrix for Random Walkers

The transfer matrix for the diffusion process also describes the probability that a **random walker** at source vertex i moves to target vertex j at the next time step.



Transfer Matrix Column property

The random walker must go somewhere.

or

All the flow described by T must move somewhere.

⇒ the sums of all the columns of T must equal 1

e.g. i -th column is

$$\sum_j T_{ji} = \sum_j \frac{1}{s_i^{(\text{out})}} A_{ji} = \frac{1}{s_i^{(\text{out})}} \left(\sum_j A_{ji} \right) = 1 \quad (7.60)$$

Use fact that max/min of sums of columns provides upper/lower bounds on λ_1

⇒ ★ Largest eigenvalue is always $\lambda_1 = 1$ ★

Conservation Property

If number of random walkers/flow is conserved

\Rightarrow the total value $W = \sum_i w(t)_i$ same at all times.

$$W(t+1) = \sum_j w(t+1)_j \quad (7.61)$$

$$= \sum_j \sum_i T_{ji} w(t)_i \quad (7.62)$$

$$= \sum_i \left(\sum_j T_{ji} \right) w(t)_i \quad (7.63)$$

$$= \sum_i w(t)_i = W(t) \quad (7.64)$$

PageRank centrality measure (simple version)

PageRank R (simple version)

The PageRank R_i of vertex i is proportional to the long-time limit of the following process

$$R_i = \lim_{t \rightarrow \infty} w(t)_i, \quad w(t+1)_j = \sum_i T_{ji} w(t)_i \quad (7.65)$$

The number of random walkers at vertex i at time t is $w(t)_i$.

The PageRank R_i of vertex i is the number of random
walkers at vertex i in equilibrium.

Problems with Simple PageRank

If network is *not* strongly connected we get some issues:-

- What if there were no exits from one vertex $k_i^{(\text{out})} = 0$?
 $T_{ji} = 0/0$ so badly defined process
One solution: define $T_{ji} = 1/N$ if $k_i^{(\text{out})} = 0$ i.e. random walkers jump to randomly chosen vertex if reach a dead end.
- What if there were no entrances from one vertex $k_i^{(\text{in})} = 0$?
 $R_i = 0$
Solution: live with this or find another method. PageRank will always give lowest rank to vertices with no in-degree.

PageRank centrality measure (general version)

Brin & Page (1998)

PageRank R (general version)

The PageRank R_i of vertex i is proportional to the long-time limit of the following process

$$R_i = \lim_{t \rightarrow \infty} w(t)_i, \quad w(t+1)_i = (1 - \alpha)h_i + \alpha \sum_j T_{ij}w(t)_j \quad (7.66)$$

May interpret the random walkers choosing to move along an edge to a neighbour fraction α of the time, or making a ‘hyperjump’ to a random vertex (chosen i with probability $\propto h_i$) fraction $(1 - \alpha)$ of the time.

N.B. Normally hyperjump to any vertex equally likely, $h_i = 1/N$

PageRank analysis

For convergence we want $|\alpha|\lambda_1 < 1$ (proof as Katz) so $|\alpha| < 1$ as then $\lim_{t \rightarrow \infty} (\alpha T)^t = 0$ giving us the PAGERANK centrality measure R_i as of vertex i as the long time limit of this diffusion+hyperjump process

$$R = \lim_{t \rightarrow \infty} w(t) = (1 - \alpha) \left[\sum_{s=0}^{\infty} (\alpha T)^s \right] h = (1 - \alpha) [I - \alpha T]^{-1} h \quad (7.67)$$

May interpret this the random walkers choosing to move along an edge to a neighbour fraction α of the time, or making a 'hyperjump' to a random vertex (prob. $\propto h_i$) fraction $(1 - \alpha)$ of the time.

$$\text{N.B. may write } R = \frac{(1-\alpha)}{(1-\alpha T)} h$$

PageRank, Web Pages and Google

Brin and Page (1998) defined a web search engine in terms of the PageRank algorithm.

The basis for the ranking of web sites in a google search.

The higher the PageRank (centrality) of a web page in a web page network, the higher it is placed when returned by a search engine.



A screenshot of a Google search results page. The search query "complex networks" is entered in the search bar. The results show several links, including one from Wikipedia about complex networks, a link to a paper titled "The structure and function of complex networks" by M.E.J. Newman, and a link to the "Center for Complex Network Research". Below the search results, there are thumbnail images related to complex networks, such as network graphs and brain scans.

PageRank, Web Pages and Google

Brin and Page (1998) defined a web search engine in terms of the PageRank algorithm.

The basis for the ranking of web sites in a google search.

The higher the PageRank (centrality) of a web page in a web page network, the higher it is placed when returned by a search engine.



A screenshot of a Google search results page. The search query "complex networks" is entered in the search bar. The results show several links, including one from Wikipedia about complex networks, a link to a paper titled "The structure and function of complex networks" by M.E.J. Newman, and a link to the "Center for Complex Network Research". Below the search results, there are thumbnail images related to complex networks, showing various network structures like brain activity maps and social networks.

PageRank, Web Pages and Google

The basis of Google's success is

- Web pages form a directed network
- Web pages do not record which other pages link to them

which means that

- PageRank is not trivial yet it is a better match statistically to how we view the web
- You need to know about the links on all web pages to calculate PageRank

⇒ Google performs a difficult non-trivial yet useful task



A screenshot of a Google search results page. The search query "complex networks" is entered in the search bar. The top result is a link to Wikipedia titled "Complex network". Below the search bar, there are tabs for "Web", "Images", "Books", "News", "Videos", "More", and "Search tools". The main search results area shows several snippets and images related to complex networks, including a small thumbnail image of a brain labeled "BrainNet" and another labeled "Complex network - BrainNet".

7. Processes on Networks

7.1 Introduction

7.2 Eigenvectors and Eigenvalues

7.3 Perron-Frobenius Theorem

7.4 Broadcasting & Centrality measures

7.5 Diffusion & PageRank Centrality

7.6 Summary

Processes on Networks: Summary

We have

- looked at eigenvalues, left- and right-eigenvectors and the relationships between them.
- stated the Perron-Frobenius Theorem and interpreted primitive and irreducible matrices as strongly connected networks.
- described local processes of broadcasting and diffusion on a network in terms of linear algebra.
- defined Eigenvalue centrality & Katz centrality in terms of a broadcast process
- defined PageRank Centrality in terms of diffusion
- we have derived expressions for Eigenvalue Katz, & PageRank centrality using linear algebra

C & N Part II: End of Networks Lecture 16

Lecturer: Tim Evans, Theoretical Physics & Centre for Complexity Science

End of “Diffusion & Centrality measures”

Intended Learning Outcomes

- Students can describe a broadcasting process on a network.
- Students can derive expressions for Eigenvalue and Katz centrality measures in terms of the adjacency matrix of a network.