Statistical Network Analysis

Debarghya Ghoshdastidar

University of Tübingen
Winter Semester 2018/19

Updated on January 28, 2019
Slides may contain errors/inaccuracies

Table of contents 1

Table of contents 2

Table of contents 3

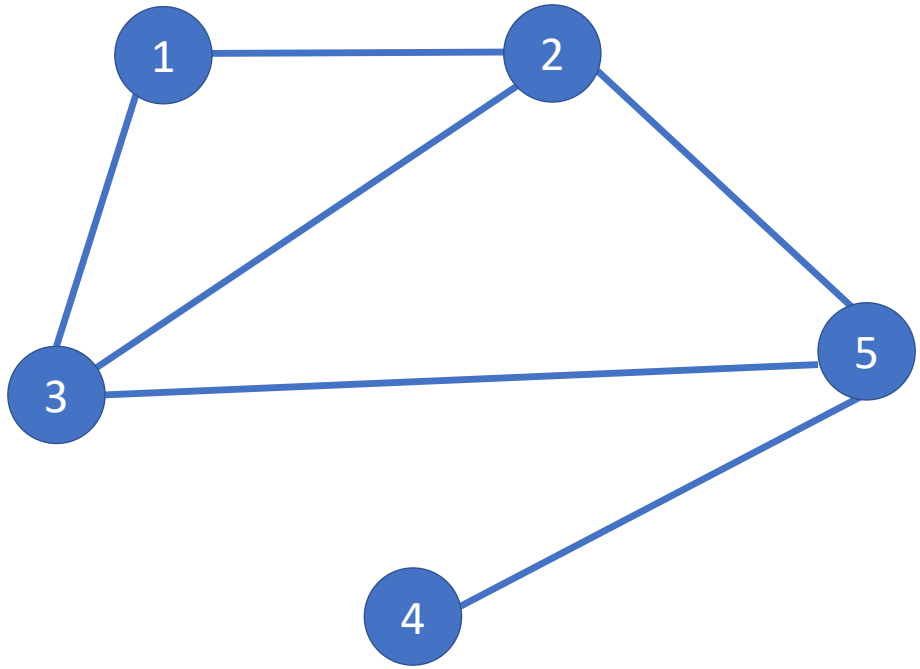
Table of contents 4

Table of contents 5

Introduction to network analysis

What are networks?

Network

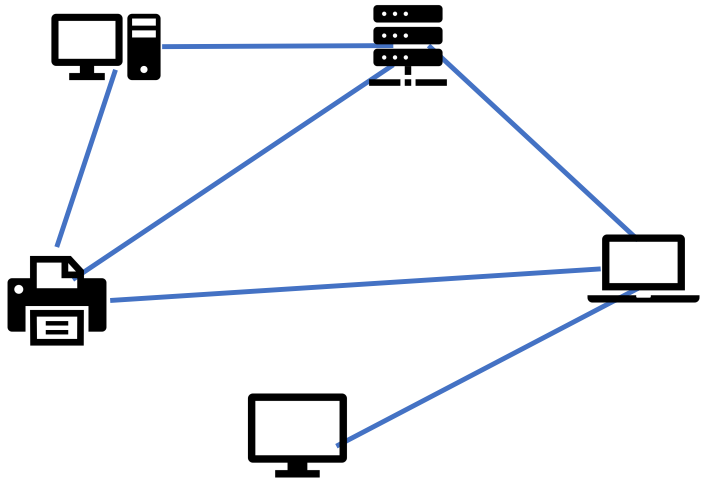


- There are some entities
  — people, countries, computers, ...
  — we will call them **nodes** or **vertices**

- Interactions occur between pairs of entities
  — friendship, emails, transactions, ...
  — we will call them **edges** or **links**

Engineering networks

- Communication networks / Internet
- Road / rail / transportation networks (TüBus, DB, ...)
- Electricity / water distribution networks

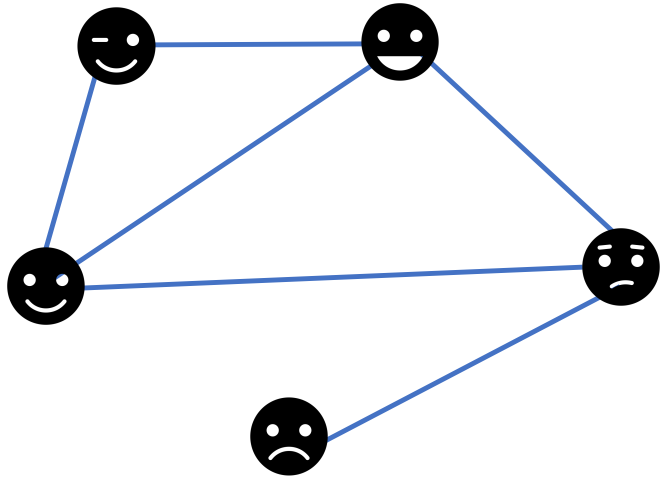


Features

- Typically connects multiple locations
- Something flows through the network
- Man-made: Usually works as planned. How to make it efficient?

Social, economic, political, . . . networks

- Social network (Facebook, Twitter, email)
- Network of friends / Society
- Trade networks among countries

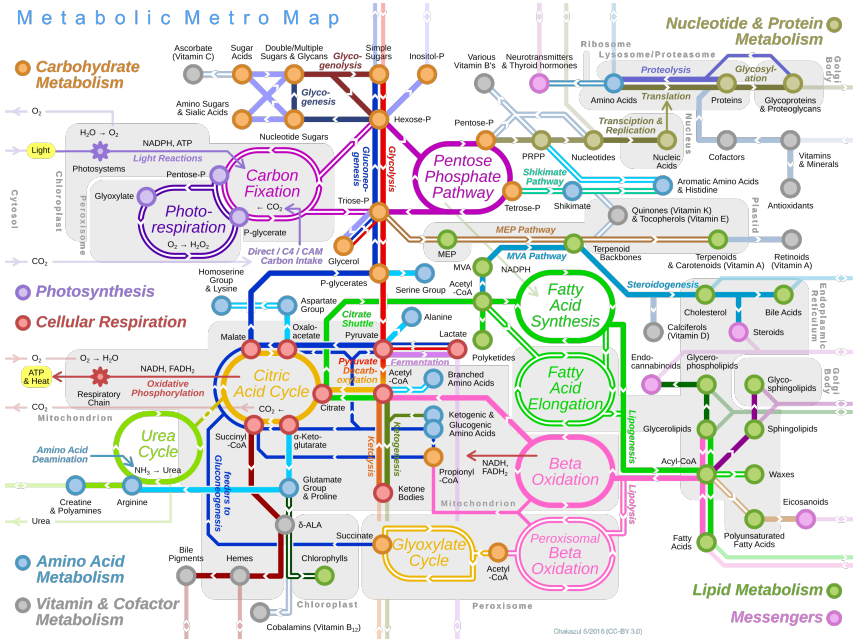


Features

- Real interactions, but not a physical network
- Behaviour of network is not predictable
  — since it involves people / companies / governments

Biological networks

- Neural network . . . the one in our brain
- Metabolic network, Gene regulatory network, . . .



[Image: Wikipedia]

Features

- Represent biological process
- Interaction denotes influence / passage of information
- Can be unpredictable because of our lack of understanding

Any many other types of networks

- World Wide Web
- Citation network
- Collaboration network
- Hierarchy in an organisation
- Sensor network
- ...

And then, we can view some data as networks:

- Movie ratings by users
- Stock correlation networks
  — correlation among stocks of different companies
- ...

What is network analysis?

Network analysis: Definitions from dictionaries

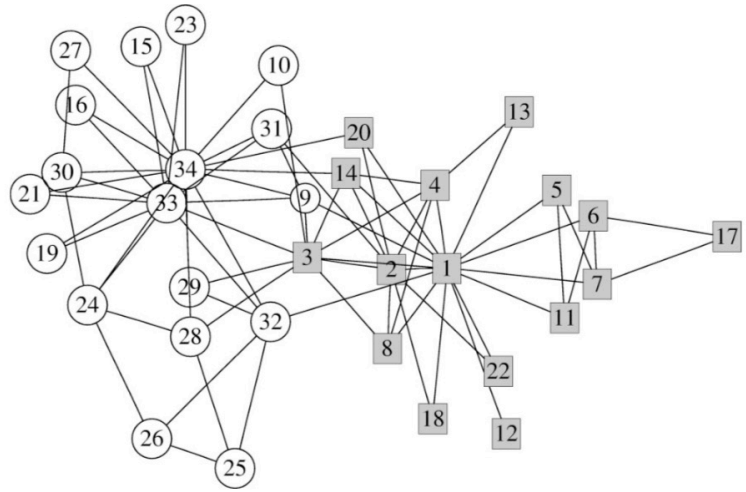
- Oxford: *The mathematical analysis of complex working procedures in terms of a network of related activities.*
- Dictionary.com: *A mathematical method of analysing complex problems, as in transportation or project scheduling, by representing the problem as a network of lines and nodes.*
- Cambridge: *The process of deciding in what order tasks need to be done in a particular project, so that it can be finished successfully in the least amount of time.*
- Longman: *Another name for Critical Path Analysis.*

Why should we care?

- Networks exist everywhere (science, business, politics, ...)
- Network analysis helps in scientific understanding, new technology
  - We will see examples soon
- Networks are not simple to understand or analyse
  - Compare networks to the standard machine learning setting where each data has d features
- Remember the variability in the definitions?
  - Each field looks at network analysis in its own way
  - Basic principles of network analysis is same in most domains
    — We need an unified view to communicate across disciplines

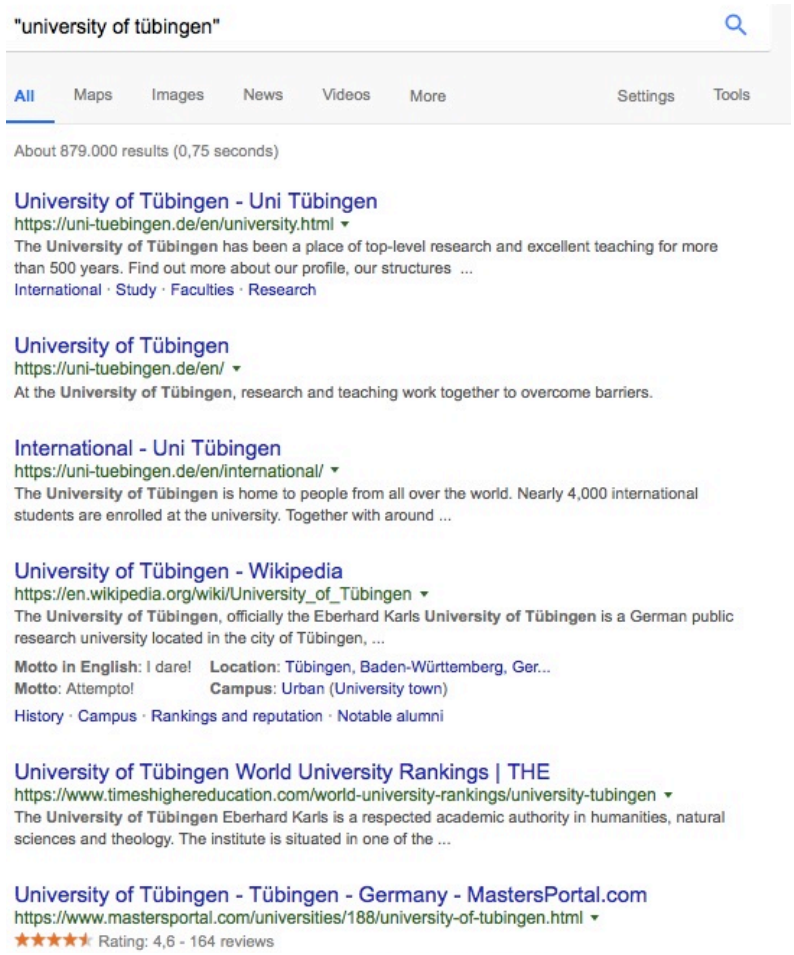
Network analysis example: Zachary's karate club

- Zachary was studying social behaviour in a karate club in 1970s
- The club had 34 members including 2 instructors;
  Conflict arose between the instructors, and the club split into two
- Zachary created a network among the members
  based on who interacted outside the club
- Zachary use a graph based algorithm
  to split the network into two parts
- Correctly predicted new group
  memberships for 33 members
- This problem is called
  community detection

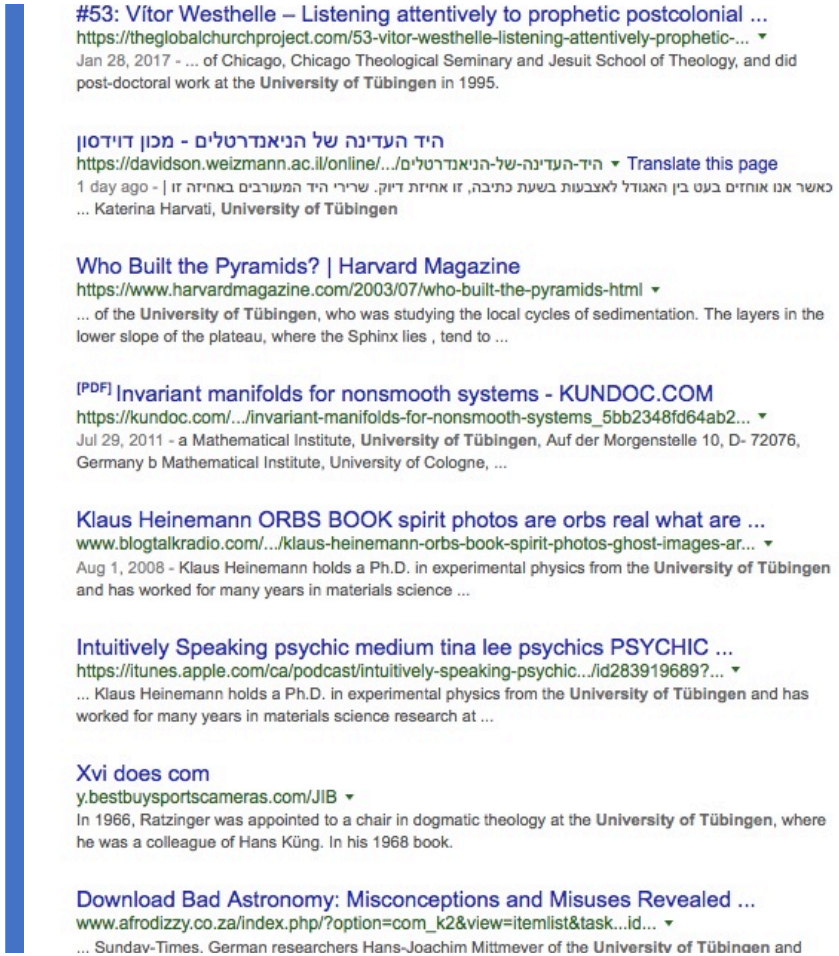


[Image: Girvan & Newman, PNAS 2002]

Network analysis example: Google's PageRank 1



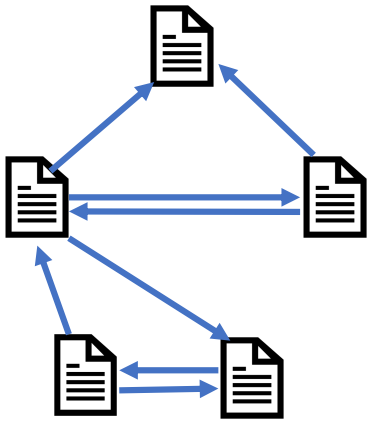
Top results by Google
(out of ~ 1 million)

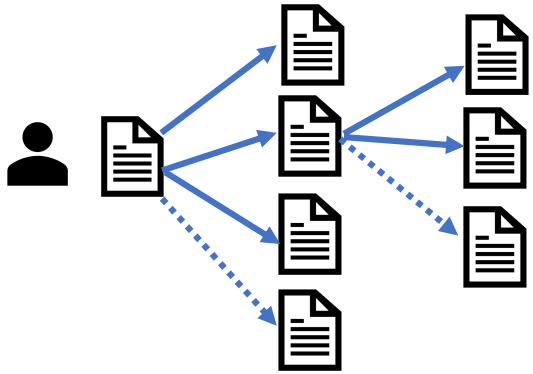


Arbitrary results

Network analysis example: Google's PageRank 2

- World Wide Web is a directed network of websites (URLs)
- Each edge $A \to B$ means website A has a hyperlink to website B





Idea

- A model for web search:
  - User starts from an arbitrary site
  - Randomly chooses one of the links, or jumps to a random site
  - Keeps doing this on every page — this is a **random walk**
- Let $p(A)$ be probability of user being in site A after infinite rounds
- Sort websites based on $p(\cdot)$ — largest $p(\cdot)$ means top result

Network analysis example: Hashtags

"In the past, if you wanted to change the world,
you had to pass a law or start a war.
Now you create a hashtag."

Hashtags do change the world

- #MeToo, #TakeAKnee, #ALSIceBucketChallenge, ...

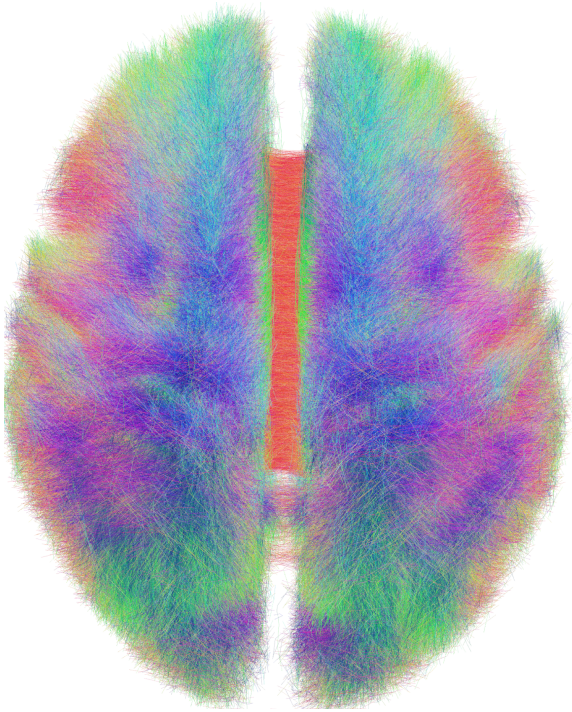
But only if you notice them

- Personalised feeds are designed by social networking companies
- You see only topics that you have been interested in the past
  — Echo chamber effect
- How can you know the world beyond your topic of interests?
  — New tools based on **information flow in networks**

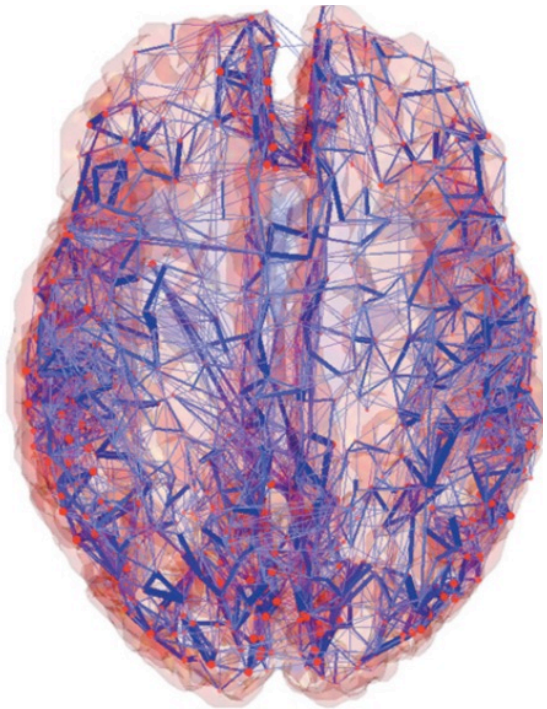
Source: Interview of Ethan Zuckerman in MIT Technology Review
"Social networks are broken. This man wants to fix them."

Network analysis example: Brain networks 1

- How does the brain work?
- Can we understand the network of neurons?
- Considerable research on combining MRI with network analysis



Connectome / Wiring diagram



Brain network

[Image: Wikipedia; MIT Technology Review]

Network analysis example: Brain networks 2

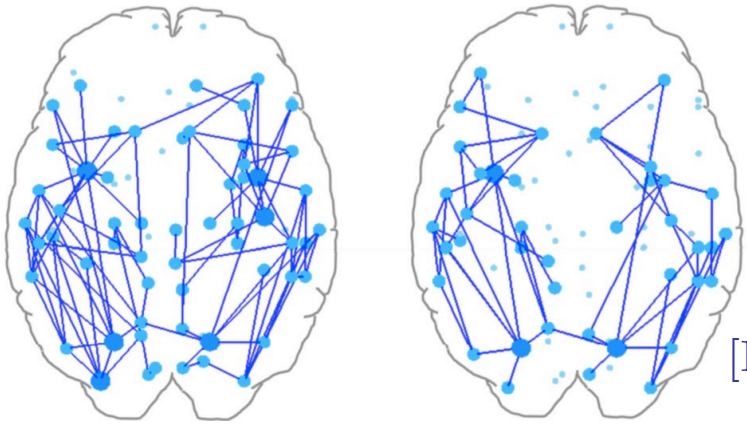
Human Connectome Project

- Difficult to collect many MRI scans at one location
- International collaboration among several universitiies
- Large collection of MRI, EEG data to understand the connectome

Understanding brain from brain network

- Brain network has **small world properties**
- Alzheimer's disease, ADHD etc. change brain network
  — problems related to **hypothesis testing**

Without Alzheimer

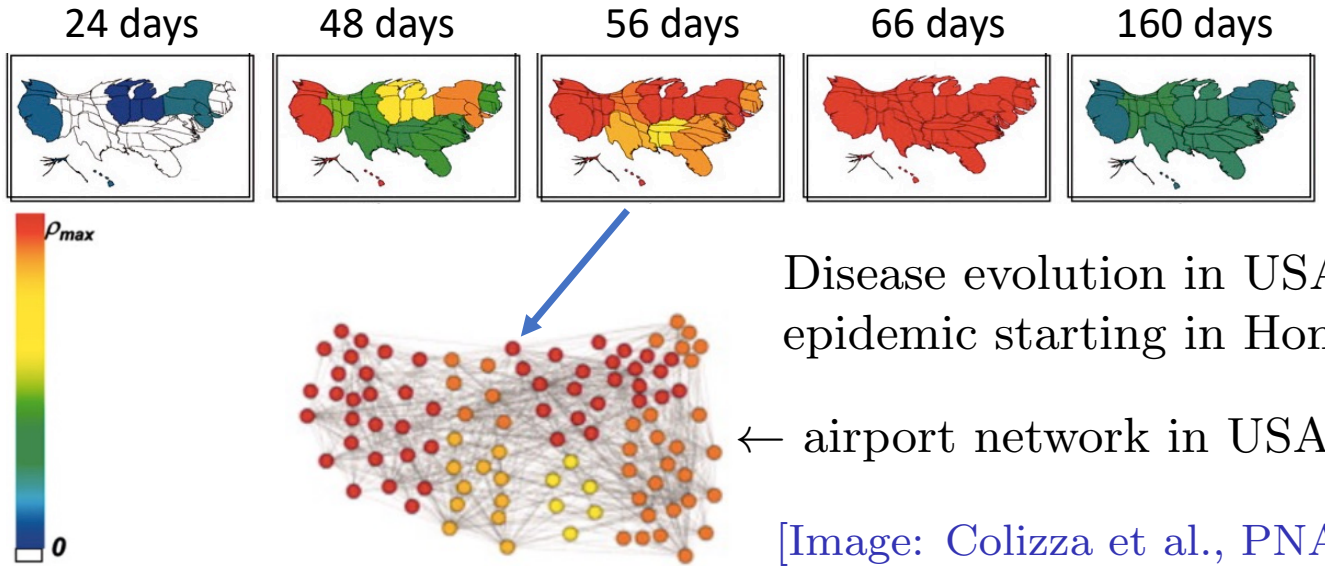


With Alzheimer

[Image: Zajac et al., Brain Sci. 2017]

Network analysis example: Epidemics and Airlines

- Remember the Ebola virus outbreak in 2013–2016?
- Epidemics cause several deaths
- Difficult to predict which region will be affected next
- Epidemics often spread through airline networks
  — involves study of **network dynamics / flow in networks**



Disease evolution in USA for an epidemic starting in Hong Kong

← airport network in USA

[Image: Colizza et al., PNAS, 2006]

Network analysis or Graph theory 1

Network analysis = Graph theory

$$+ \underbrace{\text{Domain knowledge} + \text{Mathematical modelling}}_{\text{to create meaningful networks}}$$

$$+ \underbrace{\text{Machine learning} + \text{Data analysis}}_{\text{to analyse / learn from the network}}$$

$$+ \underbrace{\text{Statistical learning theory}}_{\text{to understand how the methods work}}$$

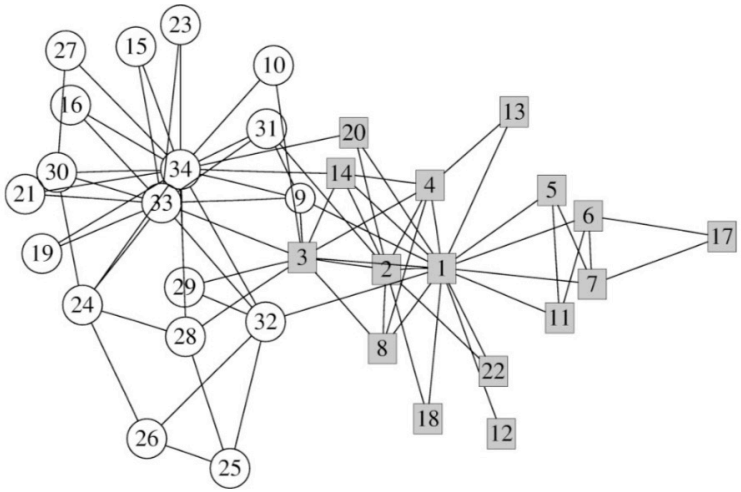
$$+ \underbrace{\text{Probability theory} + \text{Statistical physics}}_{\text{to understand the behaviour of networks}}$$

Network analysis or Graph theory 2

Example

- Recall **community detection** in the karate club network
- Underlying problem is that of **graph partitioning** — classical problem in graph theory

Focus of graph theory



- Graph partitioning: Split the vertex set into highly connected sub-groups
- Different optimization approaches: min-cut, balanced cut, . . .
- Complexity: mostly NP-hard
- Poly-time approximations: Spectral clustering, max-flow min-cut . . .

Network analysis or Graph theory 3

Components of network analysis

- Modelling : Which information / data should we use for creating the network?
- Machine learning : Formulate the mathematical problem;
  Design new algorithms;
  Efficient? Scalable to large networks?
- Learning theory : How good are these algorithms?
  — Theoretical performance guarantees
- Statistical physics : How do real networks behave? When can we find communities, patterns, . . .?

This course: Focus & logistics

Course content 1

Focus

- Machine learning + Learning theory aspects
- Main focus on mathematical principles and theory
- Bit of programming using Python and NetworkX (assignments)
- No focus on any specific application domain
- Additionally in tutorials: Mathematical preliminaries for network analysis and learning theory

Course content 2

Topics to be covered

- Network measures : We will discuss about how to describe networks in quantitative terms
- Network models : We will learn few mathematical models for networks and their properties (random graphs, geometric graphs)
- Spectral methods : We will learn the principles for spectral graph theory, some spectral algorithms and their theoretical analysis
- Network dynamics : We will discuss the principle random walk in network, and extend it to network dynamics

Course content 3

- Network visualisation : We will describe how we can meaningfully visualise networks
- More ML for networks : If time permits, we will discuss kernel methods, classification and hypothesis testing for networks
- Math tutorials : We will cover some topics like concentration inequalities, Markov chains etc.

References

- Scattered. No particular book /material
- Some reference material will be mentioned in each lecture

Course logistics 1

- Course webpage:
  `http://www.tml.cs.uni-tuebingen.de/teaching/2018_network_analysis/index.php`
  - This page contains all information
- Weekly meetings in Sand, Room F119:
  Monday 12 ct – 14 (Lecture)
  Wednesday 16 ct – 18 (Tutorial or Lecture)
  - Detailed schedule, assignment deadlines etc. will be posted regularly on course webpage
- Register on ILIAS by **October 22** (next Monday)
  - Path: Informatik / Theorie des maschinellen Lernens / Statistical Network Analysis (link on course webpage)
  - Information, assignments will be communicated through ILIAS
  - Use ILIAS forum to ask questions (no separate office hours)

Course logistics 2

Assignments

- Theory + Programming (in Python)
- Once in every two weeks
  - Assignment notification and submission through ILIAS
- Need 50% of total points in assignments to write final exam

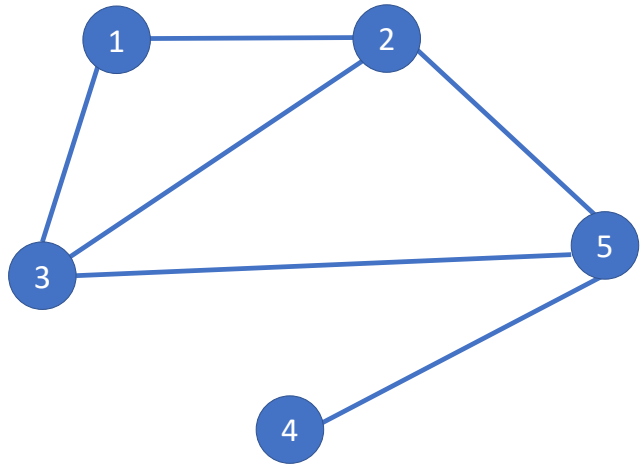
Final exam

- Grades will depend on final written exam
  - Points in assignments do not add to final grades
  - But 20% questions will be related to assignments
- Exam dates
  February 11, 2019 (Monday): 12 ct – 14 (in F119)
  April 10, 2019 (Wednesday): 10 ct – 12 (in F119)
  - Need to register for exam. Details will be announced

Network preliminaries

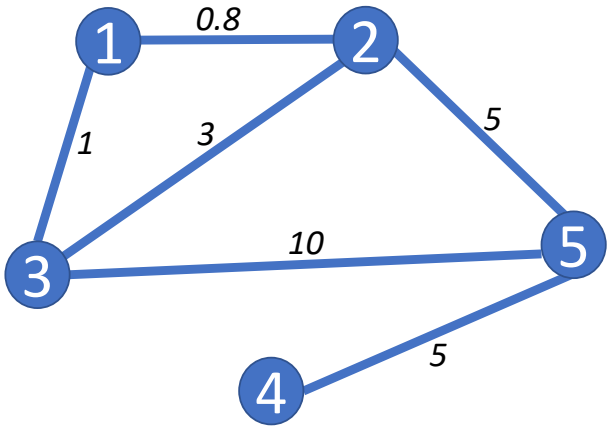
Types of networks 1

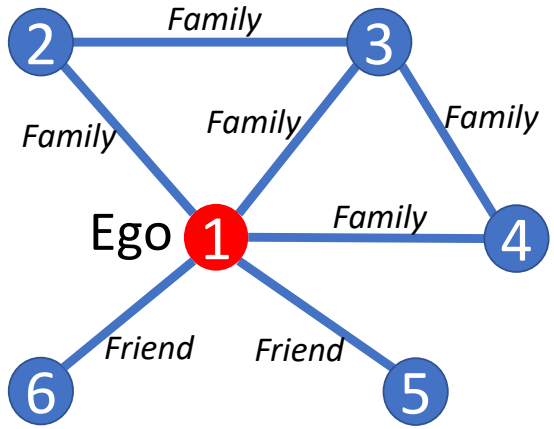
$$G = (V, E)$$



- V is a set of **nodes** or **vertices**
  - Nodes can have any name — Berlin, Informatik, John etc.
  - We will mostly write: $V = \{1, 2, 3, \ldots, n\}$
    n = number of nodes
- E is set of **edges** or **links** (interactions between pairs of nodes)
  - This is an **undirected graph**
  - Every edge $e = (u, v)$ represents a both ways connection
  - Here, edges are unweighted (we assume weight of each edge is 1)

Types of networks 2





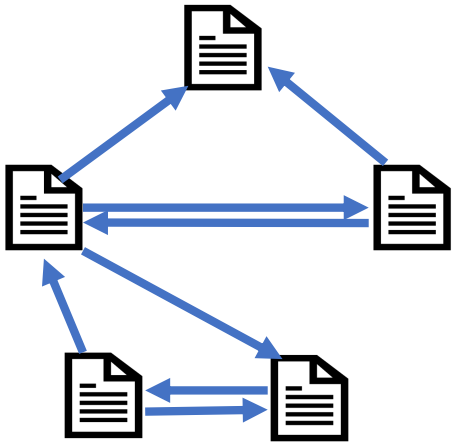
Graphs with edge weights or labels

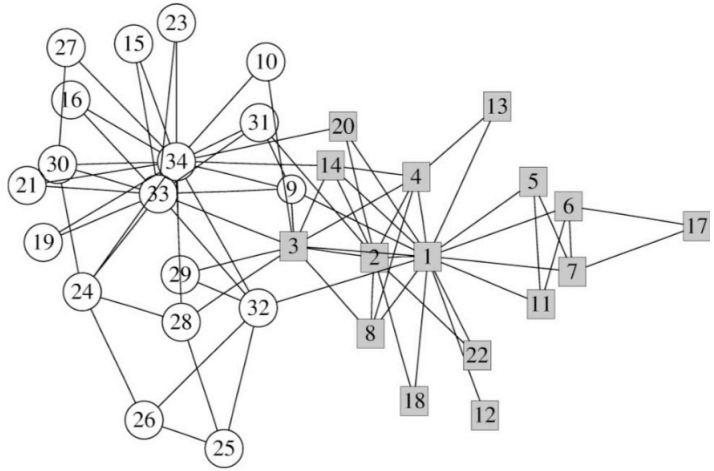
- Weighted edges: Each edge has a real-valued weight or cost
  - Network of cities (weight can be location / travel time)
- Edges with labels: Each edge can have multiple labels
  - Ego network (common in social network analysis)
  - *Ego* is a central node connected to every other node
  - Edge labels are relative, friend, colleague etc.
- Signed network: Each edge has two labels $+1$ or -1
  - Friend/foe network: $+1$ means friend, -1 means foe

Types of networks 3

Directed graphs

- Each edge has an direction from **source node** to **target node**
  - Example: World Wide Web; Transaction network
  - Edges can also have weights or labels





Graphs with node labels or attributes

- Karate network: Each node has a label depending on membership
- Often forms the basis of semi-supervised learning or classification problems in networks (we know few labels and predict others)

Types of networks 4

Many other types

- Trees, Acyclic graphs
  - Family tree, Bayesian network
  - Can be undirected, directed, weighted
- Bipartite graphs
  - Two types of nodes, and edges only go from one group to the other
  - Amazon reviews network: Users and Items are two node types, and each edge denotes an User reviewed / rated an Item

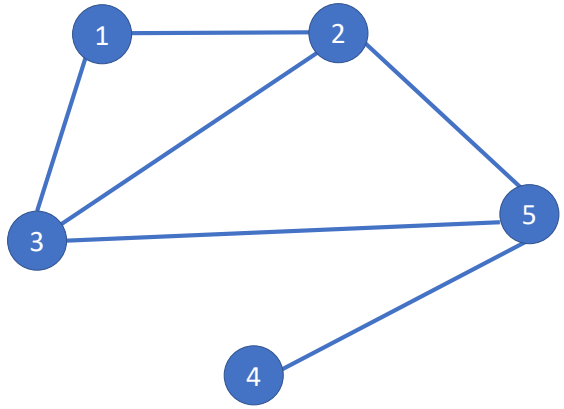
Note

- Which type of network to use?
  — Depends on problem and application
- We will mostly study undirected unweighted graphs

Representing a network 1

Adjacency matrix

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



- A is $n \times n$ matrix (n = number of nodes)
  - Unweighted graph: $A_{ij} = 1$ if $(i, j) \in E$, and 0 otherwise
  - Weighted graph: A_{ij} = weight of edge (i, j)
  - A is symmetric for undirected graphs, and asymmetric for directed
- A can be very sparse for real networks (very few non-zero entries)
  - Facebook friendship network: $n = 2.23$ billion
    #edges ≈ 173 billion, fraction of non-zero entries $\approx 7 \times 10^{-8}$
- Practically inefficient, but useful for math!

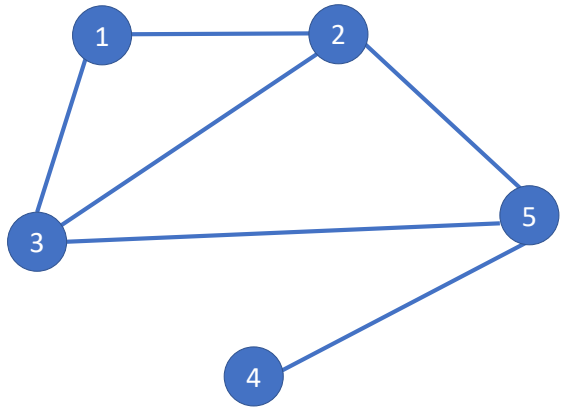
Representing a network 2

Edge list

$(1, 2)$
$(1, 3)$
$(2, 3)$
$(2, 5)$
$(3, 5)$
$(4, 5)$

Adjacency list

$1: 2, 3$
$2: 1, 3, 5$
$3: 1, 2, 5$
$4: 5$
$5: 2, 3, 4$



- Memory and computationally efficient for large, sparse graphs
- Edge list: Popular format for storing graphs
- Adjacency list: Fast retrieval of neighbours of node
- Adjacency matrix/list, edge list can be defined for directed graphs

Network measures

References

- A list of many network measures:
  M. Rubinov & O. Sporns (2010). Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52(3), pages 1059-1069
  (see Appendix A, pages 1066-1068)

- Networks used in the lecture are from
  Stanford Large Network Dataset Collection
  `http://snap.stanford.edu/data/index.htmll`

- Lecture slides and videos by Jure Leskovec
  `https://web.stanford.edu/class/cs224w/index.html`
  (see handouts for lectures 2, 5)

- Notes by Albert Barabasi on properties of real networks
  `http://barabasi.com/f/623.pdf`

Degree distribution

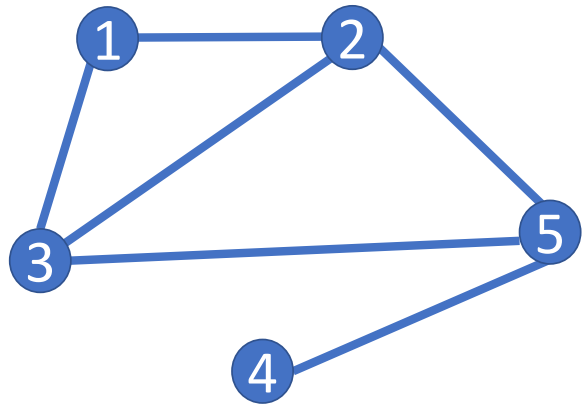
Degree of a node

Undirected graph

$$\text{degree}(i) = \text{number of neighbours of } i$$

$$= \sum_{j=1}^{n} A_{ij}$$

$\deg(2) = 3$



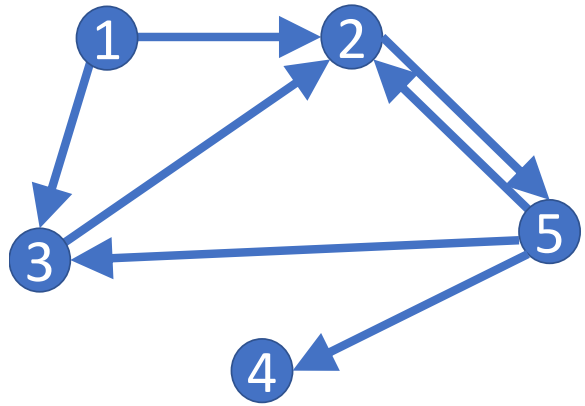
Directed graph

$$\text{out-degree}(i) = \text{number of edges from } i$$

$$= \sum_{j=1}^{n} A_{ij}$$

$$\text{in-degree}(i) = \text{number of edges to } i$$

$$= \sum_{j=1}^{n} A_{ji}$$



$\text{in-deg}(2) = 3, \ \text{out-deg}(2) = 1$

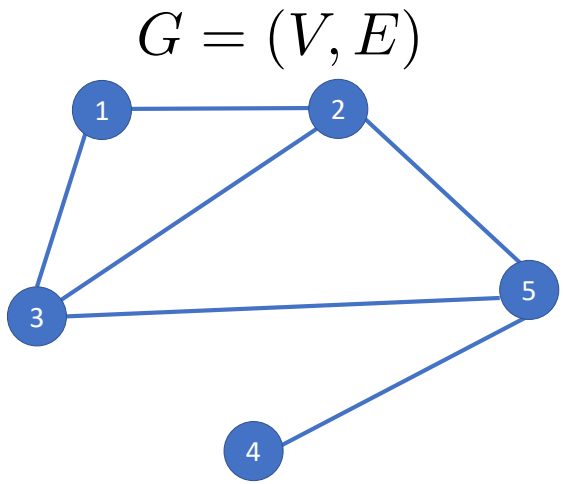
Note: A is adjacency matrix

Density, Average degree (unweighted undirected graph)

Edge density (ρ)

$$\rho = \text{fraction of possible edges in } E$$

$$= \frac{1}{\binom{n}{2}} \sum_{i<j}^{n} A_{ij} = \frac{1}{n(n-1)} \sum_{i,j=1}^{n} A_{ij}$$



Average degree ($\bar{d}$)

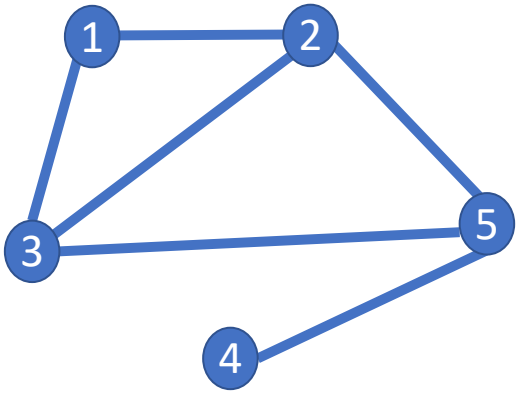
$$\bar{d} = \frac{1}{n} \sum_{i=1}^{n} \deg(i) = \frac{1}{n} \sum_{i,j=1}^{n} A_{ij}$$

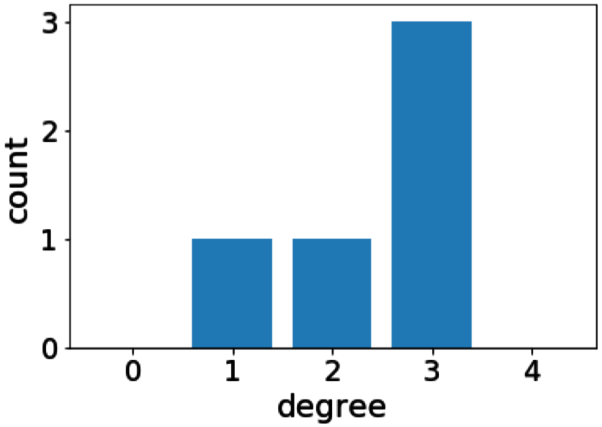
$$\rho = 0.6$$

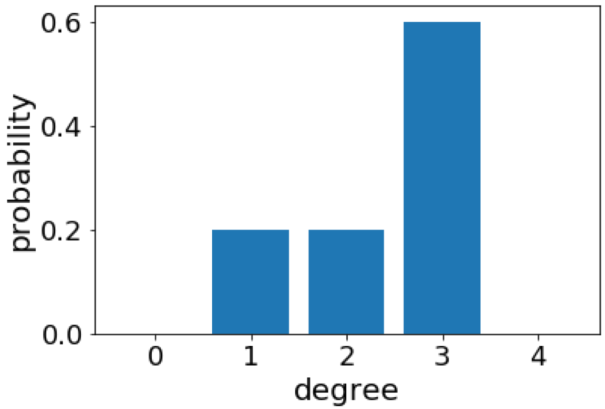
$$\bar{d} = 2.4$$

	ρ	$\bar{d}$
• Complete graph:	1	$(n-1)$
• Empty graph (graph with no edges):	0	0
• Path graph (single path on n nodes):	$\frac{2}{n}$	$2\left(1-\frac{1}{n}\right)$

Degree distribution (undirected unweighted graph)







- Number of nodes with degree d, where $d = 0, 1, \ldots, (n-1)$
- Often normalised so that total count is 1
  - Denotes proportion of nodes with specific degree
  - It is a probability mass function

$$p(d) = \frac{n_d}{n} \qquad (n_d = \text{number of nodes with degree } d)$$

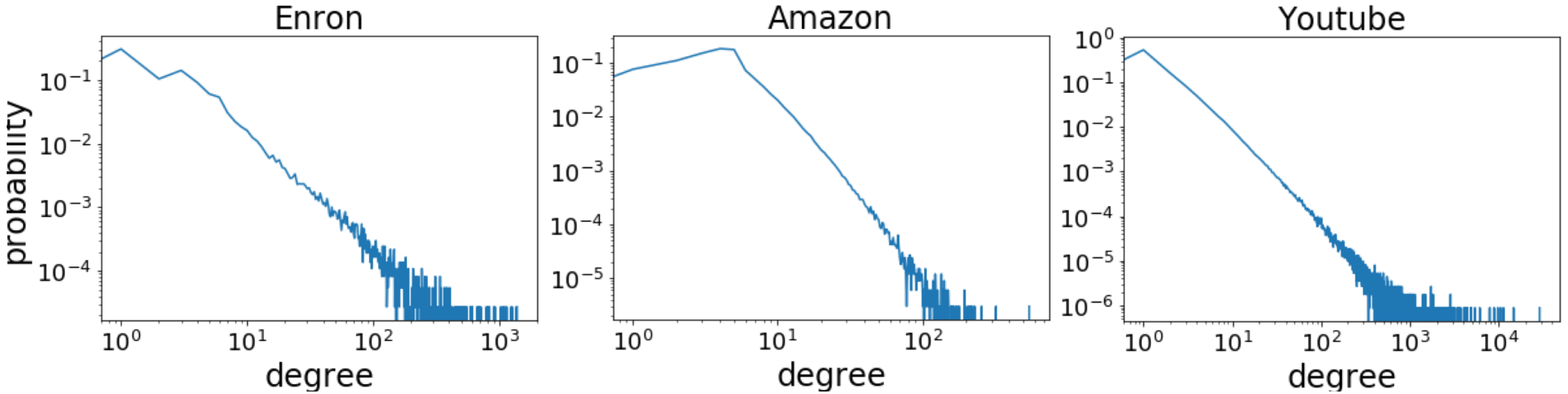
- Provides a summary of the degree of all nodes

Density and degrees in real networks 1

- Networks are sparse (low density, small average degree)

Network	n	#edges	ρ	$\bar{d}$
Email interactions (Enron)	36692	183831	2.73×10^{-4}	10.02
Item co-purchase (Amazon)	334863	925872	1.65×10^{-5}	5.53
Friendship network (Youtube)	1134890	2987624	4.64×10^{-6}	5.26

Degree distributions (normalised)



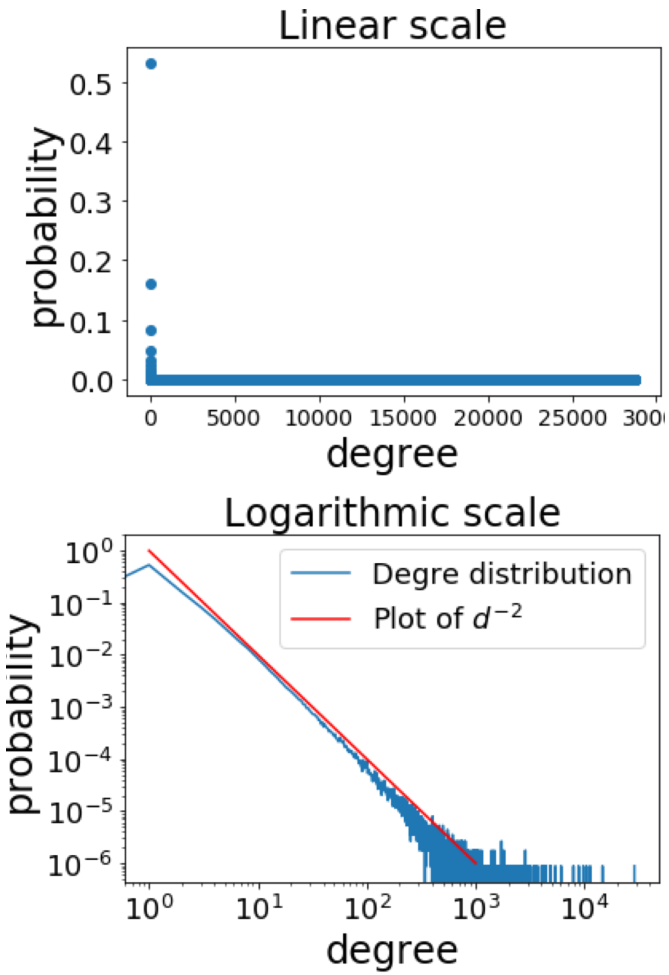
Density and degrees in real networks 2

- Beware of the scales in plot
  - Linear plots for degree distribution are meaningless for large networks
- Degree distributions typically follow **power law**
  - $p(d)$ = fraction of nodes with degree d
  - α = some positive constant

$$p(d) \propto d^{-\alpha}$$

- Degrees provide local information about the network
  - How many neighbours does a node have?
  - Degree distribution shows how **local properties** vary over the network

Degree distribution (Youtube network)

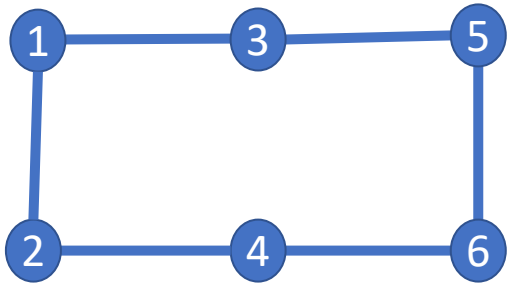


Connected components

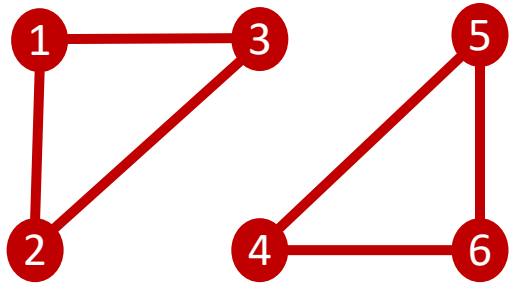
Connectness of network 1

- Is the network connected?
  - Can we reach from any node to any other node?
  - **Global property** of a graph

- No, if there are **isolated nodes** (nodes with degree zero)
  - If there is no isolated node, degree does not say anything
  - Both graphs below have same degrees but different connectivity



Connected graph



2 connected components

- **Connected components**
  - Largest possible connected subgraphs of a graph
  - Subgraph on $\{1, 2\}$ is **not** a connected component (not maximal)

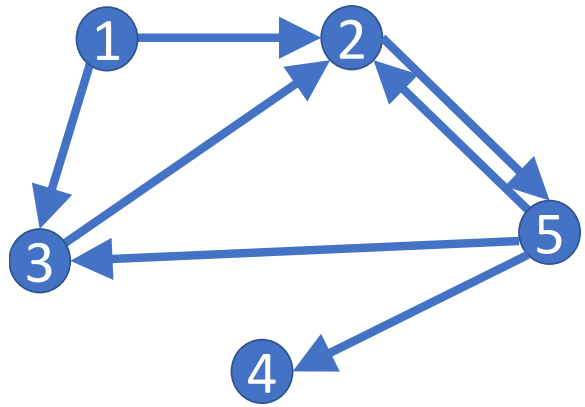
Connectness of network 2

- **Giant component**
  - Largest connected component in a graph
- How to find connected components?
  - Any search algorithm (breadth first search)
- Real networks
  - Sparse, but typically have a large giant component

	Enron	**Youtube**
Nodes n	36692	1134890
Average degree $\overline{d}$	10.02	5.26
Isolated nodes	0	0
Number of components	1065	1
Giant component size	33696 (92%)	1134890 (100%)
2^{nd} largest component	20 (0.05%)	–

Connectedness in directed graphs

- Connectivity / reachability is not bi-directional in di-graphs
- **Strongly connected graph:** If every node can be reached from every other node
- **Strongly connected component:** Maximal strongly connected subgraph
- Zero degree nodes
  - **Isolated:** in-degree = out-degree = 0
  - **Source:** in-degree = 0
  - **Sink:** out-degree = 0



$\{2, 3, 5\}$ strongly connected

1 is source, 4 is sink

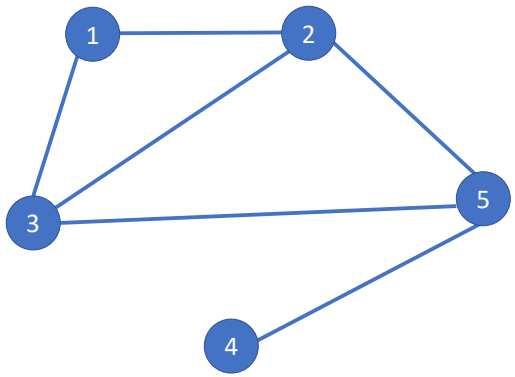
Paths in graphs

Graph paths

- **Path:** Sequence of nodes $\{u_1, u_2, \ldots, u_\ell\}$ such that $(u_i, u_{i+1}) \in E$ for every i
  - Length of path: Number of edges in path (unweighted graph)
- **Shortest path:** Path of smallest length between two vertices
  - Shortest path distance / Geodesic distance

$$\mathtt{d}_{sp}(u, v) = \begin{cases} \text{length of shortest path between } u \text{ and } v \\ \\ \infty \text{ if no path exists} \end{cases}$$

  - Symmetric for undirected graph, but not for di-graph



$\{1, 3, 2, 3, 5\}$ is a path of length 4

$\{1, 2, 5\}$ is a shortest path between 1, 5

$\mathtt{d}_{sp}(1, 5) = 2$

Path lengths in networks

- **Diameter:** Maximum shortest path distance in a graph

$$\text{diam}(G) = \max_{u,v \in V} \mathtt{d}_{sp}(u, v)$$

  - Maximum number of hops needed to reach any one

- **Average path length / Characteristic path length:**
  Average of shortest path distances between all pairs

$$L(G) = \frac{1}{n(n-1)} \sum_{u,v \in V} \mathtt{d}_{sp}(u, v)$$

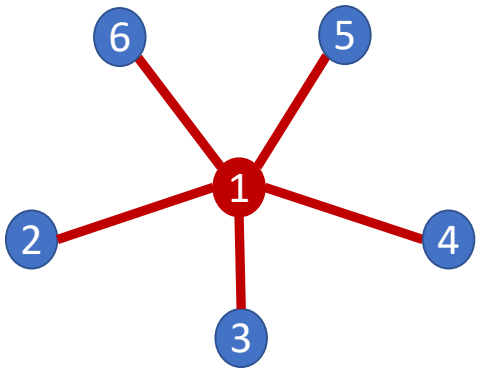
  - Typically used only for connected graphs
  - If graph is not connected, only consider connected components

- Diameter for real networks?
  - $O(n^2 \ln n + n \cdot \#\text{edges})$ time using Djikstra's algorithm
  - Takes too long for Enron (diameter = 13), Youtube etc.

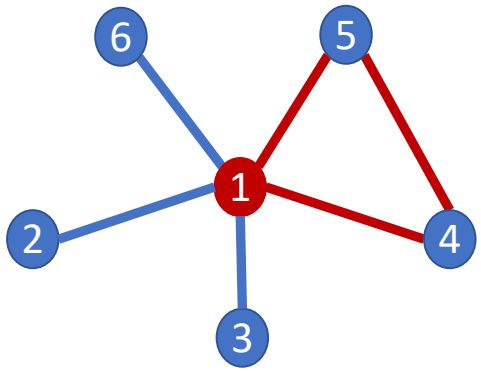
Local structures in networks

Going beyond degrees

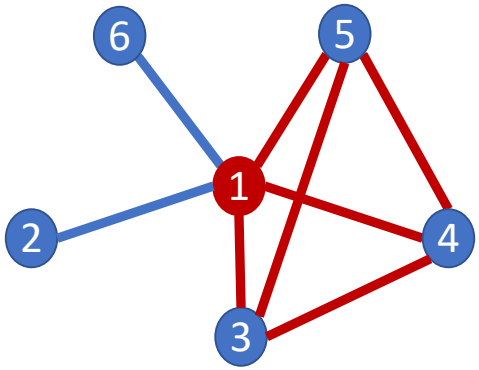
- Node degree: First level of local information
  - How many neighbours does a node have?
  - Does not say how well the neighbours are connected
- Diameter, shortest path: Long range information
  - How well is a node connected to all other nodes?
- Can we find something in between?
  - **Motifs, Graphlets:** Small patterns in a network
  - Connected subgraphs (k-cliques, path of length m, etc.)
  - Still local, but gives more information than degrees



Degree



Triangles

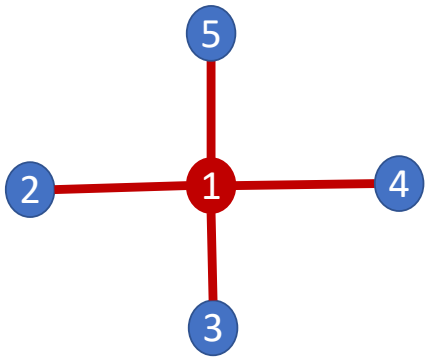


4-cliques

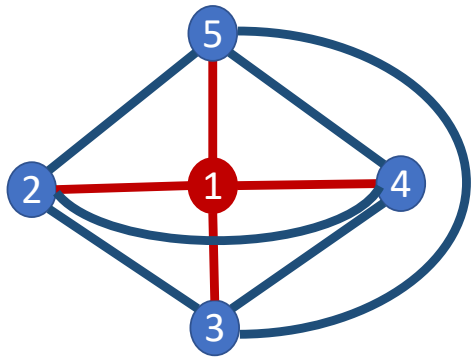
Clustering coefficient (undirected graph) 1

- Triangle: Two neighbours of a node are also connected
  - Friend of a friend is often a friend
  - Most simple motif in undirected graphs
- Number of triangles possible?
  - If node-u has degree d_u, it can be in $\frac{1}{2}d_u(d_u-1)$ possible triangles
- **Local clustering coefficient** (for a node)

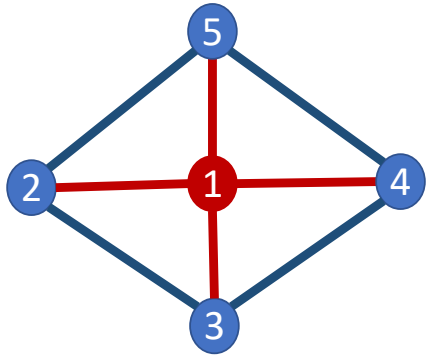
$$\mathrm{CC}(u) = \frac{\#\text{triangles containing node-}u}{\frac{1}{2}d_u(d_u-1)}$$



$\mathrm{CC}(1) = 0$



$\mathrm{CC}(1) = 1$



$\mathrm{CC}(1) = 2/3$

Clustering coefficient (undirected graph) 2

- **Average (local) clustering coefficient**

$$\mathrm{CC}_{\mathrm{local}} = \frac{1}{n}\sum_{u=1}^{n} \mathrm{CC}(u)$$

- **Global clustering coefficient**

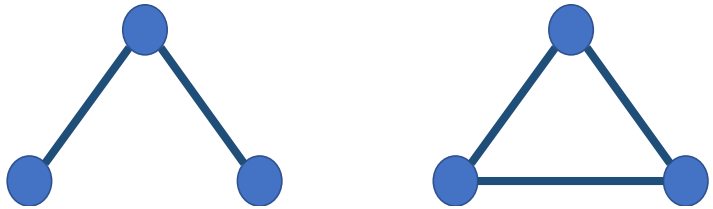
$$\mathrm{CC}_{\mathrm{global}} = \frac{3 \times \#\text{triangles in graph}}{\sum_u \frac{1}{2} d_u (d_u - 1)}$$

  - Factor of 3 counts each triangle once for every participating node
  - What fraction of triplets in the entire network form triangles?
  - Different from $\mathrm{CC}_{\mathrm{local}}$

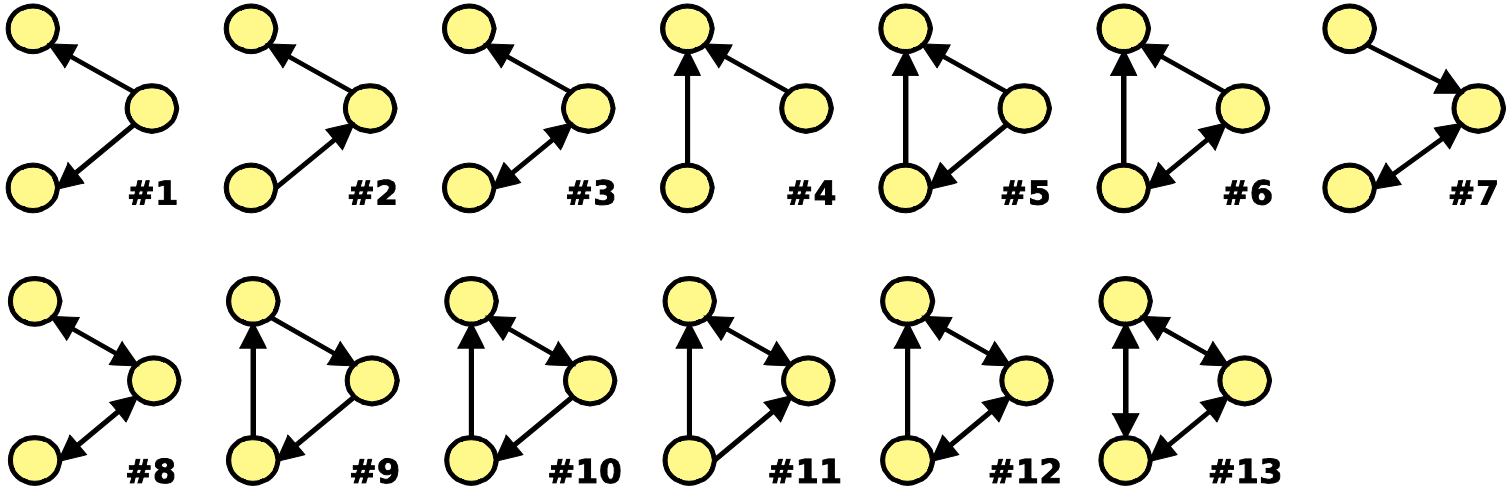
	Enron	**Youtube**
Nodes n	36692	1134890
$\mathrm{CC}_{\mathrm{local}}$	0.497	0.081
$\mathrm{CC}_{\mathrm{global}}$	0.085	0.006

Motifs 1

- Frequently occurring small connected subgraphs
- Possible 3-motifs in undirected graphs



- Possible 3-motifs in directed graphs



[Image: Leskovec lecture slides]

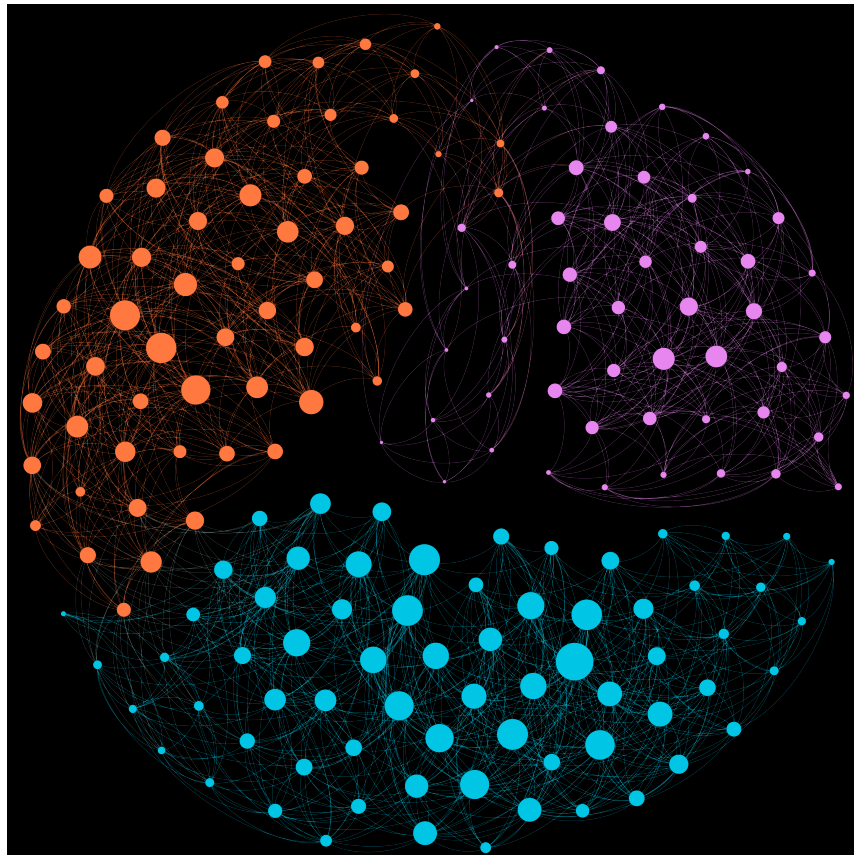
Motifs 2

- We count the number of motifs in a graph
  - Example: #edges = number of 2-motifs in a graph
- Different networks have different types of motifs in high numbers
  - How many motifs of a certain type should we expect?
    — Will discuss after we learn *network models*
- Size of motifs
  - Typically kept small (upto 5) — possibilities grow exponentially
  - Computationally expensive to find large motifs
- Further reading/viewing: Lecture-5 by Leskovec
  - `https://web.stanford.edu/class/cs224w/index.html`
  - Provides intuition for motifs and many real examples
  - Describes an algorithm to count motifs / graphlets
    (this additional material is not in the exam for our course)

Community structure

Community structure

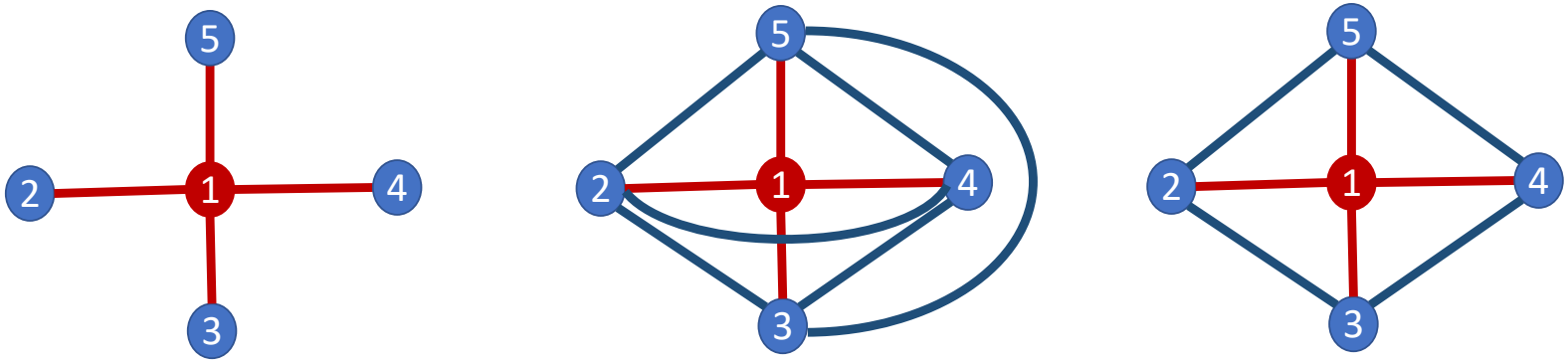
- **Community:** Group of nodes
  - Many edges among themselves
  - Few edges nodes outside community
- Does a network have communities?
  - Measured by **modularity**
- Can we find communities in network?
  - Graph partitioning
  - Spectral clustering, ...
- Will discuss after we learn
  - *Network models*
  - *Spectral graph theory*



Centrality

Centrality

- Which nodes are important in a network?

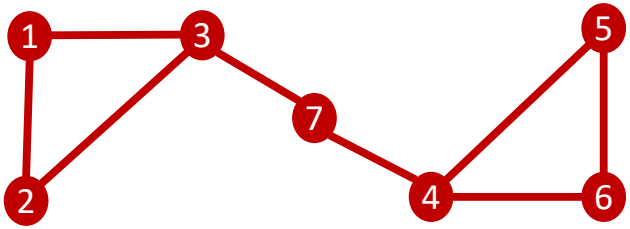


  - Left graph: No network without node-1
  - Middle graph: Other nodes well connected even if we remove node-1
  - Right graph: Is node-1 important?

- **Centrality measure**
  - Gives a score to each node to quantify its importance
  - Many different definitions — depends on application
  - Larger centrality score means node is more important / central

Centrality measures (undirected graph) 1

- **Degree centrality:** $\mathtt{C}_{\text{degree}}(u) = \text{degree}(u)$
  - More important if node has more connections



  - Not always meaningful: What happens here?

- **Closeness centrality:** $\mathtt{C}_{\text{close}}(u) = \dfrac{1}{\sum_{v \neq u} \mathtt{d}_{sp}(u, v)}$
  - $\mathtt{d}_{sp}(u, v)$ = shortest path distance between u, v
  - Node is more central if its total distance from every node is small
  - What happens if graph is disconnected?

Centrality measures (undirected graph) 2

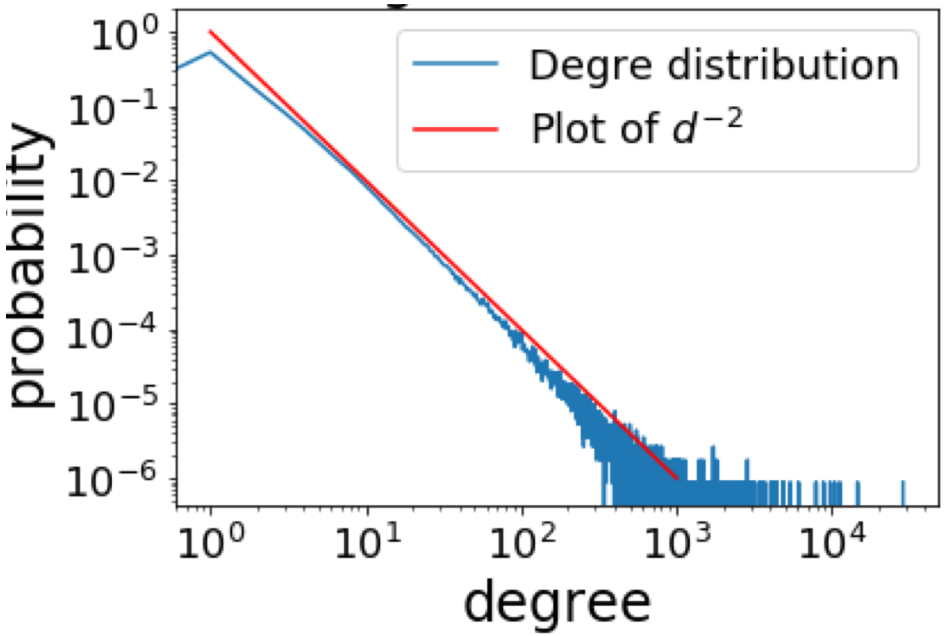
- **Harmonic centrality:** $\mathsf{C}_{\text{harmonic}}(u) = \sum_{v \neq u} \frac{1}{\mathsf{d}_{sp}(u,v)}$
  - Variant of closeness centrality
  - Meaningful even for disconnected graphs

- **Betweenness centrality:** $\mathsf{C}_{\text{between}}(u) = \sum_{\substack{s,t \neq u \\ s \neq t}} \frac{\sigma(s,t|u)}{\sigma(s,t)}$
  - $\sigma(s,t|u)$= number of shortest paths between s,t that pass through u
  - $\sigma(s,t)$ = total number of shortest paths between s,t
  - Denotes how often node-u lies between other pairs of nodes

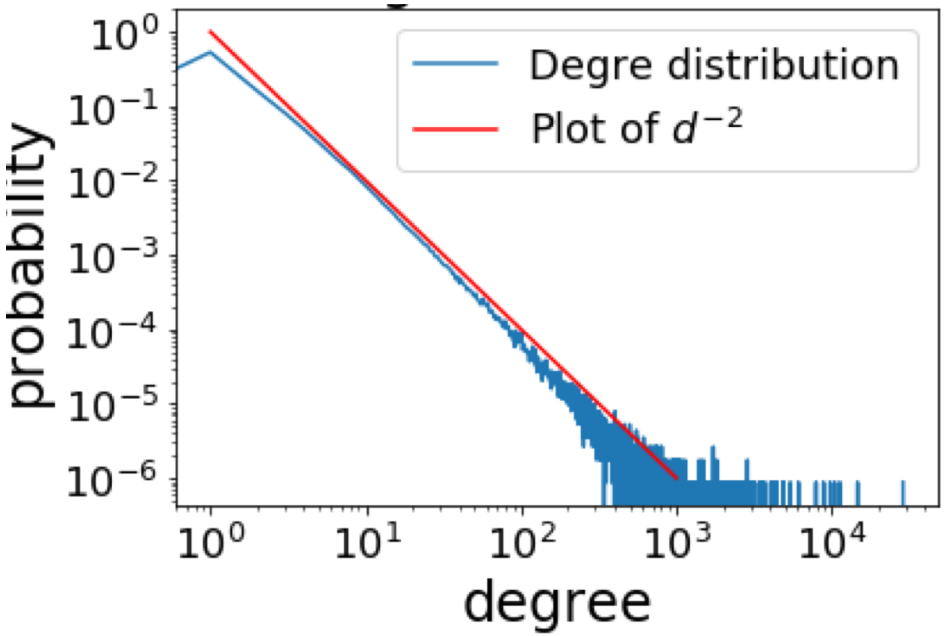
- **Eigen-centrality, Page rank:**
  - Based on eigenvectors of adjacency matrix
  - Will discuss after *spectral graph theory*

Properties of many real networks

Scale free networks



Scale free networks



- Networks with **power law degree distributions**
  - $p(d)$ = fraction of nodes with degree d (probability mass function)

$$p(d) = \frac{C}{d^{\alpha}} \qquad \text{OR} \qquad \ln(p(d)) = \ln C - \alpha \ln d$$

  - $\alpha > 1$, and C normalising constant so that $\sum_{d=0}^{\infty} p(d) = 1$

Why "scale free": First interpretation 1

- Related to notion of **scale invariance**
  - $f(\cdot)$ is scale-invariant if $\dfrac{f(ax)}{f(x)}$ does not depend on x (for large ax)
  - $f(x) = \dfrac{1}{x^r}$ is scale invariant : $\dfrac{f(ax)}{f(x)} = \dfrac{1}{a^r}$
  - $f(x) = e^{-x}$ is scaling : $\dfrac{f(ax)}{f(x)} = e^{x-ax}$
- Scale invariant distribution function
  - $F(x) = \mathbb{P}(X \leq x)$ is scale invariant distribution function if $\overline{F}(x) = 1 - F(x)$ is scale-invariant
  - $\overline{F}(x)$ is called tail probability

Theorem

$F(\cdot)$ is a scale-invariant distribution $\iff$ $\overline{F}(x) \propto x^{-r}$ for some $r > 0$

Why "scale free": First interpretation 2

- Degree distribution function is scale-invariant / scale-free
  - Let $F(d)$ = proportion of nodes with degree at most d
    $= \mathbb{P}(\text{degree} \leq d)$ . . . distribution function

$$p(d) = \frac{C}{d^{\alpha}} \quad \overset{\text{assume } d \in [1,\infty)}{\Longrightarrow} \quad F(d) = 1 - \frac{C}{(\alpha-1)d^{\alpha-1}} \qquad (\textbf{exercise})$$

$$\Longrightarrow \qquad \overline{F}(d) \propto \frac{1}{d^{\alpha-1}}$$

- High degree nodes are not *exponentially* rare
  - $\frac{1}{d^{\alpha-1}} \gg e^{-d}$ for very large d
  - Note: Degree is simply counts of neighbours
  - Compare this fact with typical laws for sums
- Power law is a special case of **heavy tailed** distribution

Why "scale free": Second interpretation

- What if degrees followed typical laws of sums?
  - Gaussian distribution, Poisson distribution, etc.
  - Let $\overline{d}$ = average degree
  - From tutorial: $\mathbb{P}\left(\text{degree} > 2\overline{d}\right)$ is exponentially small
- Networks with scale
  - Network has a scale if above happens
  - $\overline{d}$ is scale of the network
- Scale free networks
  - Network without a scale
  - $\overline{d}$ is not representative of the degrees in the network

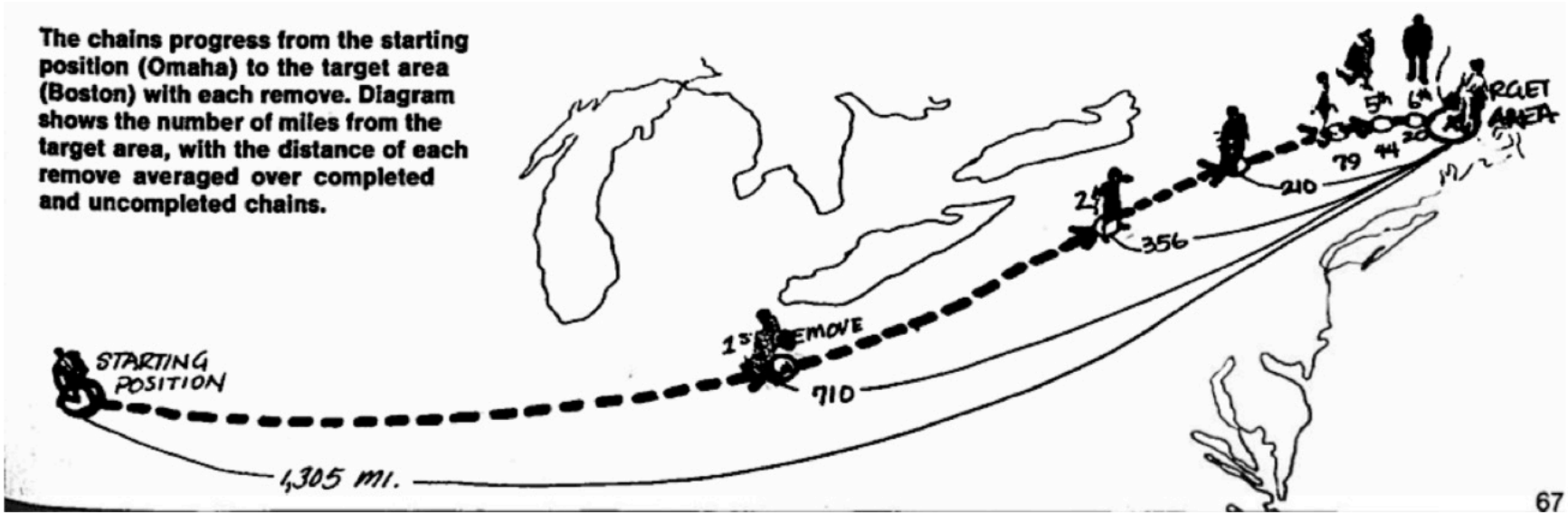
Small world networks 1

- Six degrees of separation
  - Idea started in early 1900s
  - Anyone can be connected to another person through **at most five people**
- Milgram's small world experiment (1960s)
  - Participants in Nebraska and Kansas given some letters
  - Each letter had to be delivered to a target in Massachusetts
  - Can be transfer through friends / acquaintances
- Result of Milgram's experiment
  - Only 64 out of 296 letters reached
  - Average path length for these 64 letters was between 5.5 to 6

Small world networks 2

[Image: Wikipedia]

Small world networks 3

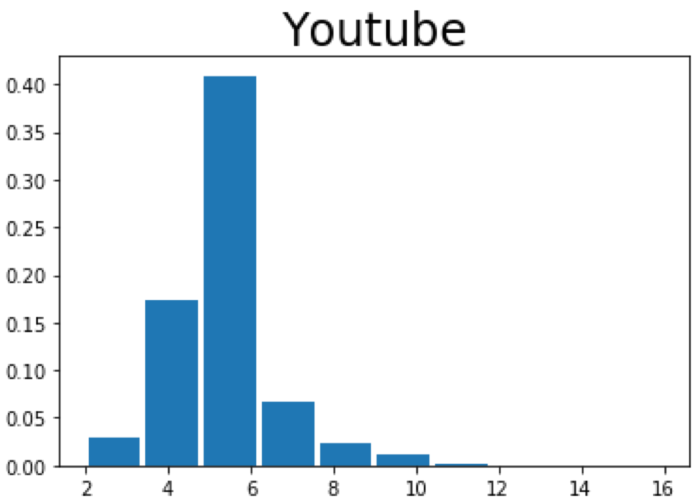


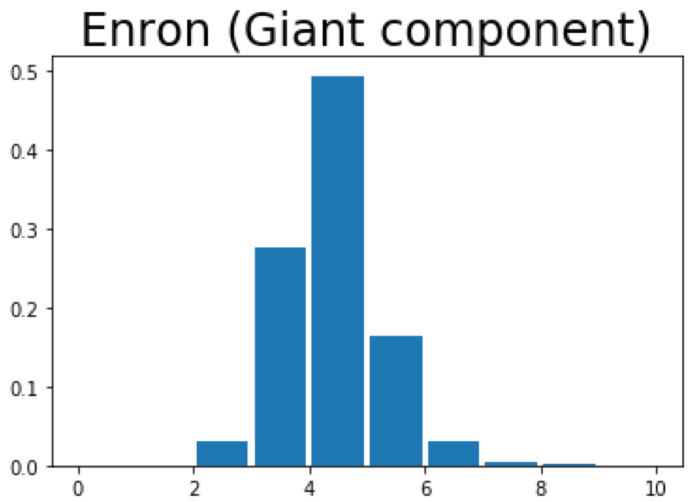
[Image: Milgram]

- **Erdős number:** Shortest paths in co-authorship network
  - How many hops from any researcher to Paul Erdős?
  - Stephen Hawking has Erdős number 4
  - Hawking — J. B. Hurtle — S. Chandrasekhar — M. Kac — Erdős

Small world networks 4

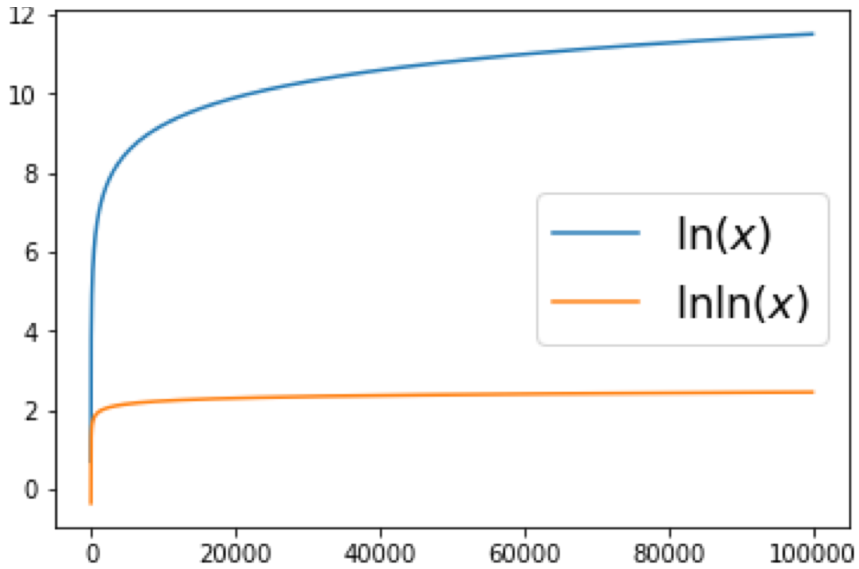
- Shortest path distances in real networks
  - Computed for 50000 random pairs
  - Average path lengths about 5 to 6
  - Do not infer diameter from these plots (**why?**)





Mathematical forms of small world property

- No precise mathematical definition
  - Some notions arise from study of network models
- Let G be a network with n nodes
  - **Small world:** $\text{diameter}(G) = O(\ln n)$
  - **Ultra-small world:** $\text{diameter}(G) = O(\ln \ln n)$
    (behaves like a constant)



Other features

Hubs

- Scale free networks have few nodes with very high degree (hubs)
- Hubs induce ultra-small world property
- Hubs have high degree centrality as well as betweenness centrality (think of a star graph)

Friendship paradox

- Scott Feld in 1991 found that:
  most people have fewer friends than their friends have on average
- Why?
  — Can be explained mathematically

Proof of friendship paradox

- Let $|V| = n$, and $d_u =$ friends of node-u
- **Average number of friends** $= \frac{1}{n} \sum_{u \in V} d_u$
- Total number of friends for everyone $= \sum_{u \in V} d_u$
- Total number of friends of friends
$$= \sum_{v \in V} \sum_{u \sim v} d_u = \sum_{u \in V} d_u^2 \qquad (u \sim v \text{ for } d_u \text{ number of } v\text{'s})$$
- **Average number of friends of friends**
$$= \text{total friends of friendstotal friends} = \frac{\sum_u d_u^2}{\sum_u d_u}$$
- $\left(\sum_u d_u \right)^2 \leq n \sum_u d_u^2$ (prove using Cauchy-Schwarz inequality)

Network models

References

- Erdős-Rényi graph: Chapter 8 of *Foundations of data science* `https://www.cs.cornell.edu/jeh/book.pdf`
- Configuration model: Aaron Clauset's notes (Lectures 11, 12) `http://tuvalu.santafe.edu/~aaronc/courses/5352/fall2013/`
- Watts-Strogatz model: Paper by Barrat and Weigt `https://arxiv.org/pdf/cond-mat/9903411.pdf`
- Preferential Attachment
  - Barabasi-Albert model: `http://barabasi.com/f/622.pdf`
  - More formal material: Chapter 3 of *Complex graphs and networks* `http://www.math.ucsd.edu/~fan/complex/`
- ER and PA (mathematical): *Random Graphs and Complex Networks (vol 1)* `http://www.win.tue.nl/~rhofstad/`

Erdős-Rényi model

The $G(n, p)$ and $G(n, m)$ models

- Two similar models for generating random undirected graphs
  - $G(n, p)$ by Edgar Gilbert (1959)
  - $G(n, m)$ by Paul Erdős and Alfred Rényi (1959)
  - $G(n, p)$ is more popular, but referred to as Erdős-Rényi model
- n, m, p are parameters
  - $n =$ number of nodes
  - $p =$ probability of an edge in graph $G(n, p)$
  - $m =$ number of edges in graph $G(n, m)$
- $G(n, p)$: For every pair of nodes i, j $(i \neq j)$
  add the edge (i, j) with probability p
- $G(n, m)$: There are $\binom{n}{2}$ pair of nodes
  Choose any m pairs randomly, and add them to edge set

Possible graphs that are generated

$G(n, m)$

- Let $\mathcal{C}_1 = \{G = (V, E) \ : \ |V| = n, |E| = m\}$
- What is the size of $\mathcal{C}_1$?
- $G(n, m) =$ Uniform distribution on $\mathcal{C}_1$

Possible graphs that are generated

$G(n, m)$

- Let $\mathcal{C}_1 = \{G = (V, E) \ : \ |V| = n, |E| = m\}$
- What is the size of $\mathcal{C}_1$?
- $G(n, m) =$ Uniform distribution on $\mathcal{C}_1$

$G(n, p)$

- Let $\mathcal{C}_2 = \{G = (V, E) \ : \ |V| = n\}$
- What is the size of $\mathcal{C}_2$?
- $G(n, p)$ can generate any graph in $\mathcal{C}_2$ — is it uniform over $\mathcal{C}_2$?

What is the nature of $G(n,p)$?

- Let $G \sim G(n,p)$

$$\mathbb{P}(G = \text{empty graph}) = (1-p)^{n(n-1)/2}$$
$$\mathbb{P}(G = \text{complete graph}) = p^{n(n-1)/2}$$

- Let $S =$ number of edges in G

$$S \sim \text{Binomial}\left(\frac{n(n-1)}{2}, p\right)$$
$$\mathbb{E}[S] = \frac{pn(n-1)}{2}, \qquad \text{Var}(S) = p(1-p)\frac{n(n-1)}{2}$$

Density and degrees of $G(n, p)$ 1

- Edge density

$$\mathbb{E}[\text{density}(G)] = p$$

- Degree (d_i) for any node-i

$$d_i \sim \text{Binomial}(n-1, p)$$
$$\mathbb{E}[d_i] = p(n-1)$$

- Average degree

$$d = \frac{1}{n} \sum_i \mathbb{E}[d_i] = p(n-1) \approx pn$$

Density and degrees of $G(n, p)$ 2

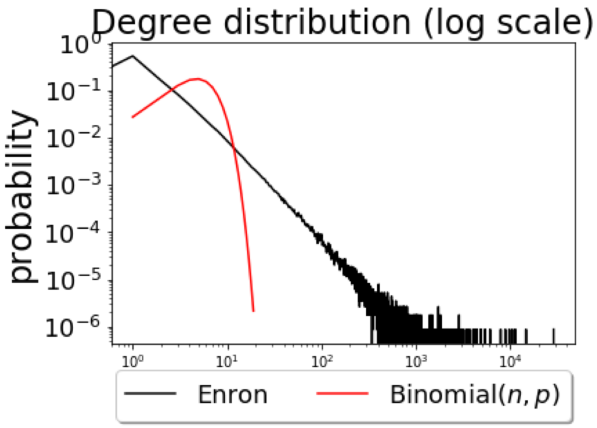
- How should we set p?
  - Real networks are sparse

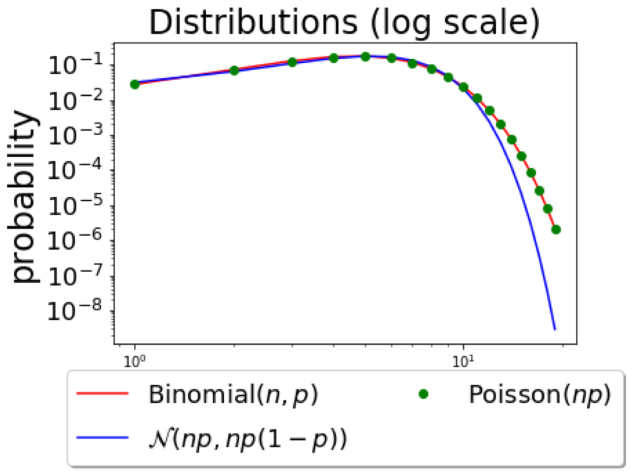
	n	#edges	density, p	d
Enron	36692	183831	2.73×10^{-4}	10.02
Amazon	334863	925872	1.65×10^{-5}	5.53
Youtube	1134890	2987624	4.64×10^{-6}	5.26

  - p decreases rapidly with n
  - $d \approx np$ behaves like a constant (or perhaps grows very slowly)
  - Possibly set $p = \dfrac{C}{n}$ or $p = \dfrac{C \ln n}{n}$

Density and degrees of $G(n, p)$ 3

- Degree distribution $\approx$ Binomial(n, p) $\qquad (n - 1 \approx n)$
- What happens when n is large?
  - Degree distribution $\approx \mathcal{N}\big(np, np(1-p)\big)$
  - If $p = \frac{C}{n}$, degree distribution $\approx$ Poisson(C)
- But in real networks, degrees follow power law distribution
  - Real network: $\mathbb{P}(\text{degree} > t) \asymp t^{-\alpha}$
  - ER model: $\mathbb{P}(\text{degree} > t) \asymp e^{-t}$





Triangles and clustering coefficient

- Let $G \sim G(3, p)$: $\quad \mathbb{P}(G \text{ is a triangle}) = p^3$
- For $G \sim G(n, p)$:
  - $\mathbb{E}[\#\text{triangles}] = p^3 \binom{n}{3}$
  - Global clustering (in expectation)

$$\mathbb{E}[\text{CC}_{\text{global}}] = \mathbb{E}\left[\frac{3 \times \#\text{triangles}}{\sum_u \frac{1}{2} d_u (d_u - 1)}\right] \approx \frac{3p^3 \binom{n}{3}}{\frac{n}{2}(np)^2} \asymp p$$

  - Average local clustering $\mathbb{E}[\text{CC}_{\text{local}}] \asymp p$

	Enron	**Youtube**
Density p	2.73×10^{-4}	4.64×10^{-6}
CC_{local}	0.497	0.081
$\text{CC}_{\text{global}}$	0.085	0.006

Connectivity: Isolated nodes 1

Theorem: Number of isolated nodes

Let $G \sim G(n,p)$, and $X = \#\text{isolated nodes in } G$.

$$\mathbb{E}[X] = n(1-p)^{n-1}$$

$$\lim_{n\to\infty} \mathbb{E}[X] = 0 \quad \text{if } p > \frac{\ln n}{n}, \quad \text{and} \quad \lim_{n\to\infty} \mathbb{E}[X] = \infty \quad \text{if } p < \frac{\ln n}{n}.$$

Proof: $\mathbb{E}[X] = \sum_i \mathbb{P}(d_i = 0) = \sum_i (1-p)^{n-1}$

Let $p = \frac{c \ln n}{n}$.

$$\lim_{n\to\infty} \mathbb{E}[X] = \lim_{n\to\infty} n\left(1 - \frac{c\ln n}{n}\right)^n$$

$$= \lim_{n\to\infty} n e^{-c\ln n} \qquad \text{since } \lim_{n\to\infty} e^{-a_n} = \lim_{n\to\infty}\left(1 - \frac{a_n}{n}\right)^n$$

$$= \lim_{n\to\infty} n^{1-c}$$

Connectivity: Isolated nodes 2

Corollary: Presence of isolated nodes

Let $G \sim G(n, p)$

$$\mathbb{P}(G \text{ contains isolated nodes}) = \begin{cases} o(1) & \text{if } p > \dfrac{\ln n}{n} \\ 1 - o(1) & \text{if } p < \dfrac{\ln n}{n} \end{cases}$$

Here, $x = o(1)$ means $\lim_{n \to \infty} x = 0$.

Proof: In tutorial.

Diameter 1

Theorem: ER graph with ultra-small world property

Let $G \sim G(n,p)$ with $p > \sqrt{\frac{2\ln n}{n}}$.

$$\mathbb{P}\big(\text{diameter}(G) \leq 2\big) = 1 - o(1)$$

Proof: In tutorial. Similar to proof for isolated nodes.

Implication

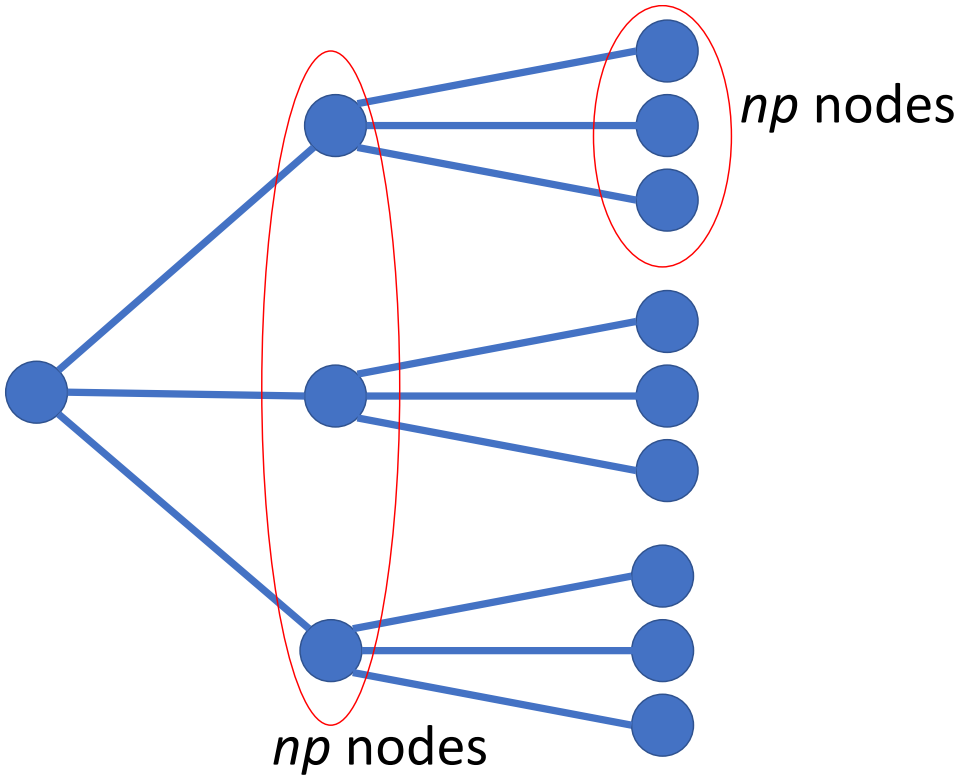
- Real networks: Diameter small due to presence of few hubs
- $G(n,p)$ do not have very high degree nodes
- Yet if $p > \sqrt{\frac{2\ln n}{n}}$, then diameter$(G) \leq 2$ **(Why?)**

Diameter 2

An informal argument (Note: This is not accurate)

$\mathrm{CC}(i) \approx p \to 0$ if $p = o(1)$. Neighbours are typically not connected

Neighbourhood of ever node is somewhat like a tree



If it is a tree
— we can reach $(np)^2$ nodes in 2 hops

If $p > \sqrt{\frac{2 \ln n}{n}}$, then
$(np)^2 > 2n \ln n$
— more than n nodes

In $G(n, p)$, there are still few triangles

Diameter 3

- $p > \sqrt{\frac{2 \ln n}{n}}$ is a relatively dense setting
- What happens for smaller p?

  Diameter > 2 for with probability $(1 - o(1))$ if $p < \sqrt{\frac{2 \ln n}{n}}$

Theorem: ER graph with small world property

Let $G \sim G(n, p)$ with $p > C_1 \frac{\ln n}{n}$.

$$\text{diameter}(G) < C_2 \ln n \text{ with probability } 1 - o(1)$$

Here C_1, C_2 are some large positive constants.

Proof: We will skip. If interested, see Theorem 8.13 in FoDS book.

Connectivity: Giant component 1

Theorem: Connectedness and presence of giant component

Each statement holds with probability $1 - o(1)$

- If $p > \dfrac{\ln n}{n}$, then G is connected

- If $p > \dfrac{\ln n}{2n}$, then G has:
  — a giant component of size $> \dfrac{n}{2}$
  — all nodes not in the giant component are isolated

Proof: We will skip. If interested, see Theorem 8.11 in FoDS book.

Connectivity: Giant component 2

Theorem: Emergence of a giant component

Each statement holds with probability $1 - o(1)$

- If $p < \frac{1}{n}$,
  — then all connected components in G are of size $< C \ln n$.

- If $p > \frac{1}{n}$,
  — then G has a giant component with $> \epsilon n$ number of nodes

Proof: We will skip. Proof based on branching processes.
Theorems 4.4 and 4.8 in Hofstadt's book OR some arguments in
`http://www.cs.yale.edu/homes/spielman/462/2010/lect5-10.pdf`

Note: $np > 1$ means average node degree > 1

Variants of ER model and related problems

Significance of ER model

- Not a good model for real networks
- Originally used for the **probabilistic method**
  - Use probability to answer deterministic questions
- Used to analyse performance of graph algorithms
  - Simple model — easier to do analysis
  - We can provide guarantees for algorithms assuming $G \sim G(n, p)$ or similar model
- Phase transitions in ER
  - **Phase transition:** Drastic changes observed if a parameter is changed a little
  - Saw this in emergence of **isolated nodes** and **giant component**
  - Has connection to problems in physics

Planted clique problem 1

Theorem: Largest clique in $G(n, \frac{1}{2})$

Let $G_0 \sim G(n, \frac{1}{2})$ and S be the largest clique in G.

$$\mathbb{P}\big(|S| < 2\log_2 n\big) = 1 - o(1)$$

Planted k-clique

- Let $G_0 \sim G(n, \frac{1}{2})$
- Choose a random subset of nodes S of size k
- Add all edges between nodes in S, and call the new graph G
- G is a random graph with a **planted clique** S

Planted clique problem 2

- **Planted clique problem:** Let $k \gg 2\log_2 n$.
  Can we find S?

Theorem: Finding large planted clique (Kucera, 1995)

Let G has a planted k-clique with $k > \sqrt{n \ln n}$.

Let $S =$ set of k nodes with highest degrees

S is the planted clique with probability $1 - o(1)$

- Better algorithms till date can find planted cliques if $k > \epsilon\sqrt{n}$
- What happens when $2\log_2 n \ll k \ll \sqrt{n}$?
- **Planted clique conjecture:**
  No polynomial time algorithm can find planted clique of size $k \ll \sqrt{n}$

Stochastic block model / Planted partition

- Let $G_1, \ldots, G_k \sim G(s, p)$
  — If p is large, each graph is connected

- G' is a graph on $n = sk$ nodes such that $G' = G_1 \cup \ldots \cup G_k$
  — G' has k connected components
  — G' has k communities with no interaction across communities

- For every pair of nodes from two different communities:
  — Add edge with probability $q < p$
  — Call this new graph G

- G is called stochastic block model
  — G is a random graph, but has a hidden partition of nodes

- Can analyse performance of graph partitioning algorithms on G

Configuration model

Revisiting ER model

Modelling reality with $G(n,p)$ or $G(n,m)$

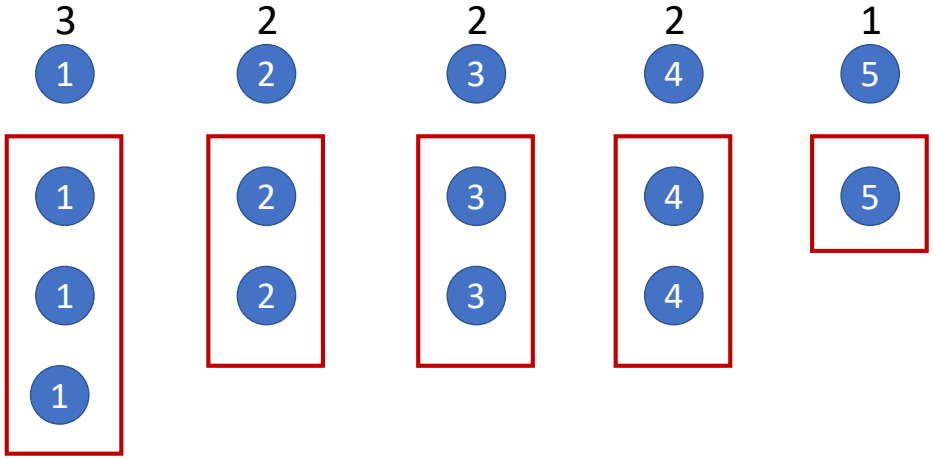
- Real networks are sparse: $p = \dfrac{C}{n}$
- Degree distribution: Binomial (asymp Poisson) . . . not scale free
- Clustering coefficient $\approx p = \dfrac{C}{n}$ . . . very low clustering
- Diameter is $O(\ln n)$ if $p > \dfrac{C \ln n}{n}$
- Giant component if $p > \dfrac{1}{n}$
- **What about $G(n,m)$?** — Nearly similar to $G(n,p)$ for $p = \dfrac{m}{\binom{n}{2}}$

Configuration model 1

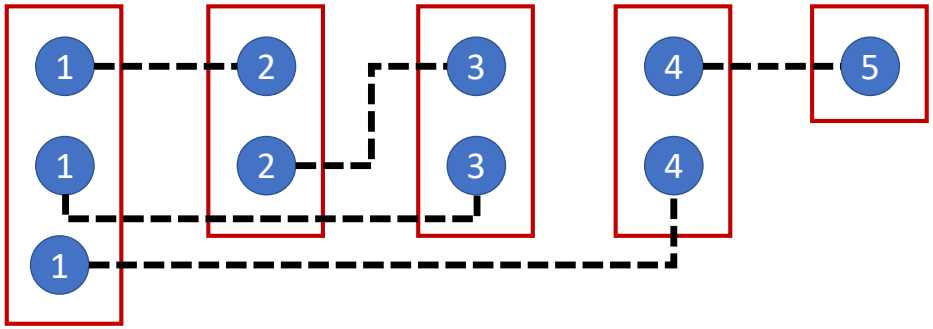
- Generalisation of $G(n, m)$
- Allows specification of node degrees
- **Generation process**
  - Given degree sequence $(d_1, d_2, \ldots, d_n)$ such that $\sum_i d_i$ is even
  - Create d_i copies of node-i
  - Randomly pair any two node copies (each copy paired only once)
  - Merge all copies of same node

Configuration model 2

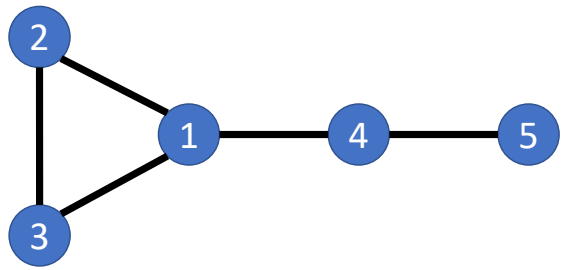
Example: Given degrees (3,2,2,2,1). Make copies of nodes



Randomly pair nodes

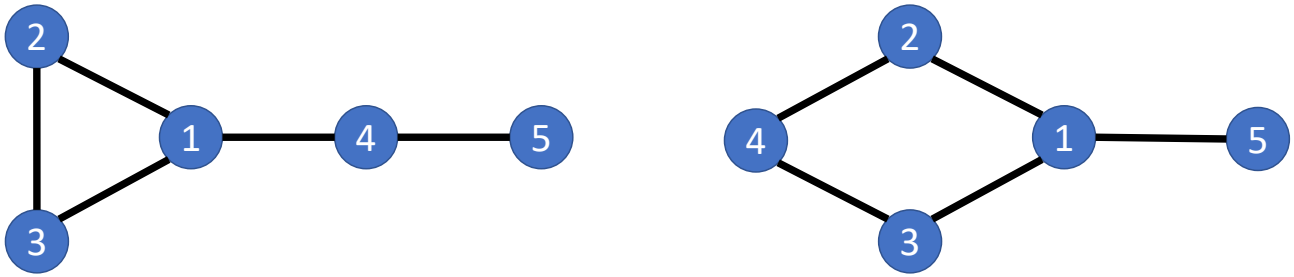


Merge copies of same node

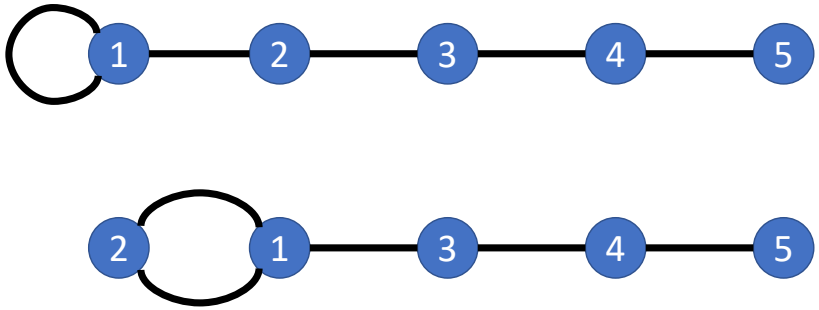


Configuration model 3

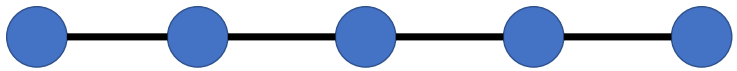
- Generates different graphs with same degree sequence



- Can also lead to self-loop or multi-edge



- **Solution:** Collapse such edges . . . what happens if we do this?



Properties of configuration model

Edge probability

Let $G = (V, E)$ be generated from CM with degrees $(d_1, \ldots, d_n)$,

$$\mathbb{P}\big((i,j) \in E\big) = \frac{d_i d_j}{2m-1} \approx \frac{d_i d_j}{2m} \qquad \text{where } m = \frac{1}{2}\sum_i d_i$$

Proof: Node-i has d_i copies.

Each copy can form an edge with node-j with probability $\dfrac{d_j}{2m-1}$.

Local properties

- Degree distribution: Any specification ... can choose power-law
- Global clustering coefficient $\asymp \dfrac{C}{n}$ for sparse graphs ... too small

Small world models

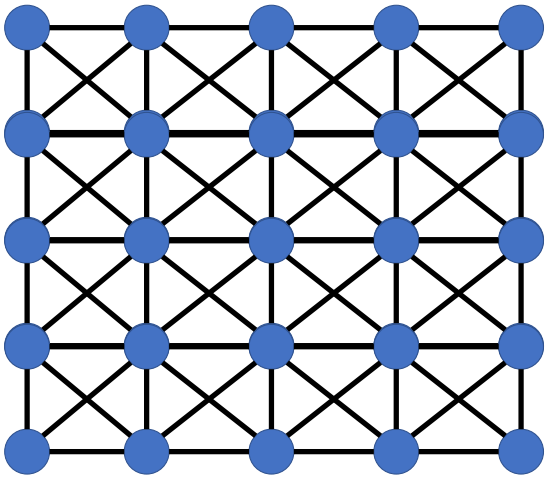
Small world network

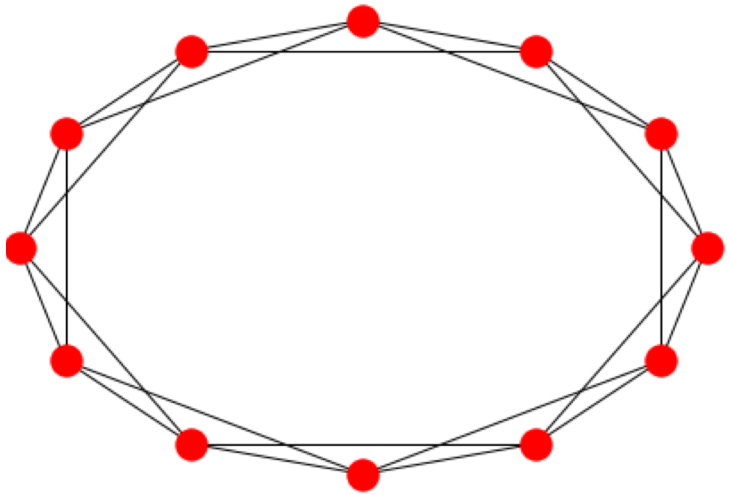
A small world network has:

- Short distances among all nodes — $O(\ln n)$ average path length
- Neighbours likely to be connected — High clustering coefficient

Contradictory features?

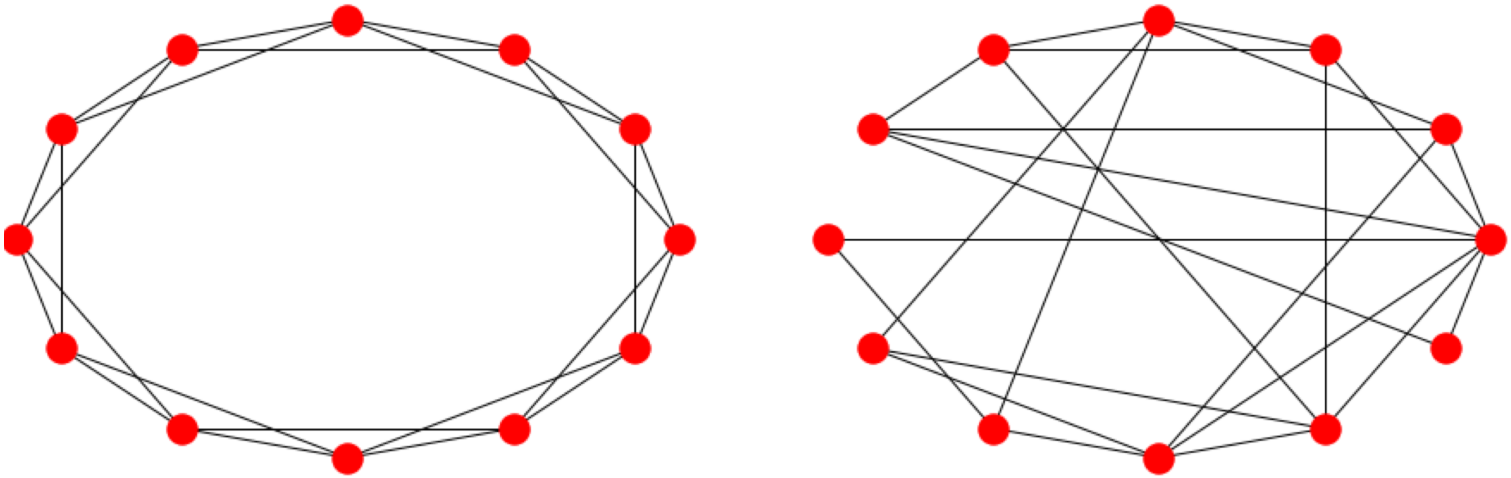
- ER and Configuration: Short distances, but low clustering
- Lattice graphs: High clustering, but large distances





Watts-Strogatz model

- Parameters: n, k, β
- Start with n nodes arranged in circle
- Connect each node to $2k$ nearest nodes
- Rewiring:
  - For every edge (i, j) in original graph (with i on left of j) with probability β, detach from j and connect to a random node



Properties of Watts-Strogatz graph 1

Let $G_\beta \sim \text{WSM}(n, k, \beta)$

- Average degree $= 2k$ (irrespective of β)
- Consider $\beta = 0$
  - G_0 only has connection with nearby nodes
  - Diameter $\approx \dfrac{n}{2k}$
  - Average path length $\approx \dfrac{n}{4k}$
  - Number of triangles containing node-$i = \frac{3}{2}k(k-1)$
  - Clustering coefficient (global / local) $= \dfrac{3(k-1)}{2(2k-1)}$

(**Exercise:** Verify these properties)

Properties of Watts-Strogatz graph 2

Clustering coefficient of G_β

$$\mathrm{CC}_{\text{local}}(G_\beta) \approx \mathrm{CC}_{\text{global}}(G_\beta) \approx \frac{3(k-1)}{2(2k-1)}(1-\beta)^3$$

Proof idea: For every triangle,

- each of the 3 sides are not changed with probability $(1-\beta)$
- all three sides not modified with probability $(1-\beta)^3$

Average path length, $L(G_\beta)$

$$L(G_\beta) = \begin{cases} O(n) & \text{if } \beta = \dfrac{c}{n} \\ O\left(\dfrac{\ln n}{\ln(2k-1)}\right) & \text{if } \beta \to 1 \end{cases}$$

Other small world models

- Newman-Watts-Strogatz model
  - Instead of rewiring, add more edges with some probability

- Kleinberg model
  - Start with a grid in $\mathbb{R}^2$
  - Add edges between non-adjacent u, v with probability $p_{uv} = \frac{1}{\|u - v\|^r}$

- Small world models produce networks with
  - average path length $\leq C \ln n$
  - clustering coefficient $> \epsilon$
  - But, degree distributions are not power law **(Why?)**

Preferential attachment

Preferential attachment

- Previous models directly generate large graphs
- Many real networks grow over time
- Preferential attachment process
  - Models how a network grows over time
  - Principle: The rich get richer
- *Rich get richer* — in networks
  - New nodes in a network connect more with high degree nodes
- Why should we consider preferential attachment?
  - Preferential attachment typically leads to power law distributions

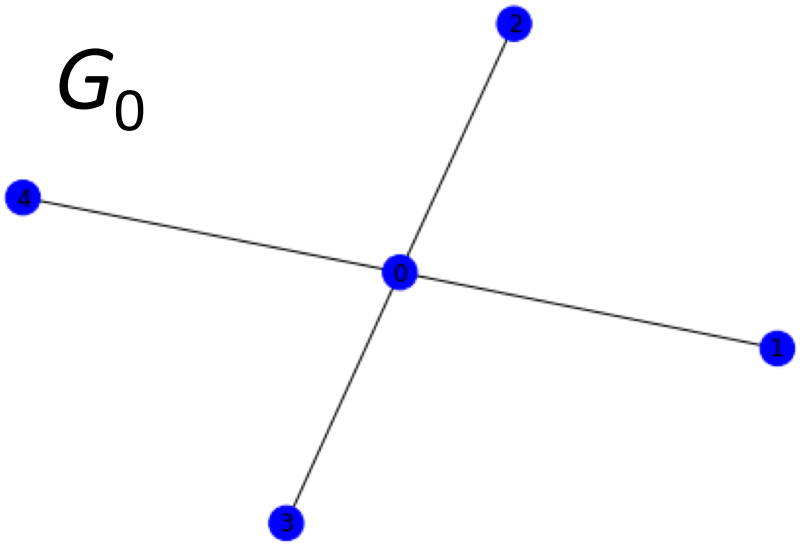
Barabasi-Albert model 1

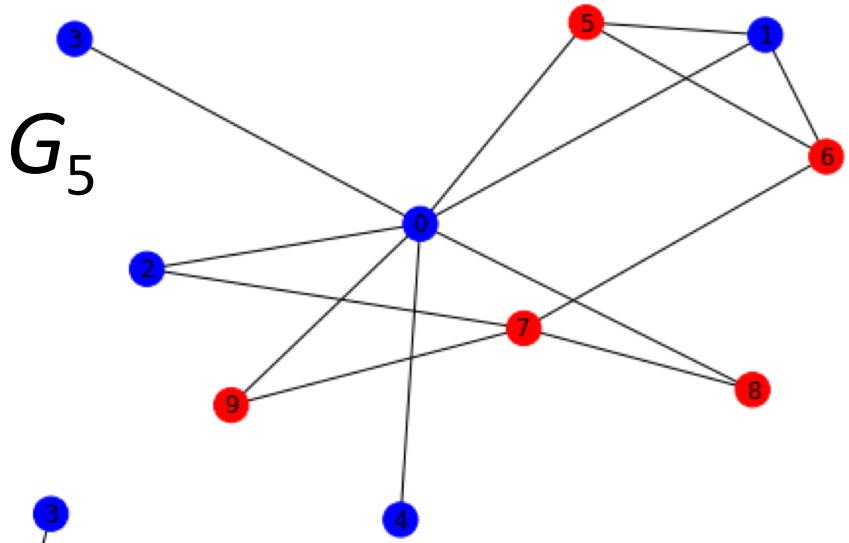
- One of many models for preferential attachment (most popular)
- Parameters
  - Initial graph $G_0 = (V_0, E_0)$
  - Integer $m \leq |V_0|$
- At each $t = 1, 2, \ldots,$ graph G_t is as follows
  - Add a new node v
  - Add edges between v and m random nodes in V_{t-1}
  - Probability of choosing node $u \in V_{t-1}$

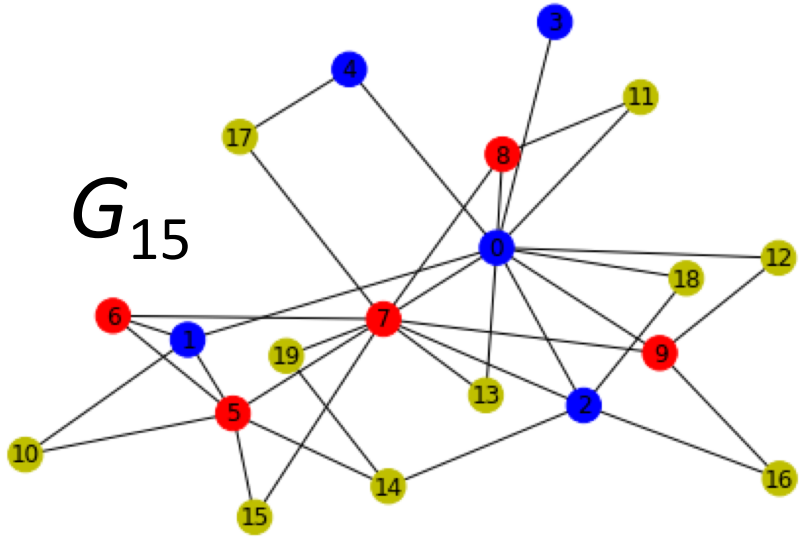
$$p_u = \frac{d_u}{\sum_{i \in V_{t-1}} d_i}$$

- NOTE: If d_u is large, u is more likely to have new connections

Barabasi-Albert model 2







$m = 2$

Properties of BA model

- Let G_0 be a small graph and t be large.
  We study properties of $G_t = (V_t, E_t)$
- $|V_t| = \big(|V_0| + t\big) \approx t, \qquad \text{and} \qquad |E_t| = \big(|E_0| + mt\big) \approx mt$
- Average degree, $\bar{d} = \dfrac{2|E_t|}{|V_t|} \approx 2m$
- Degree distribution, $p(d) \asymp d^{-3}$ (power law)
- Average path length, $L(G_t) \asymp \dfrac{\ln n}{\ln \ln n}$
- Clustering coefficient $\asymp \dfrac{(\ln n)^2}{n}$

Geometric graphs

Neighbourhood graphs 1

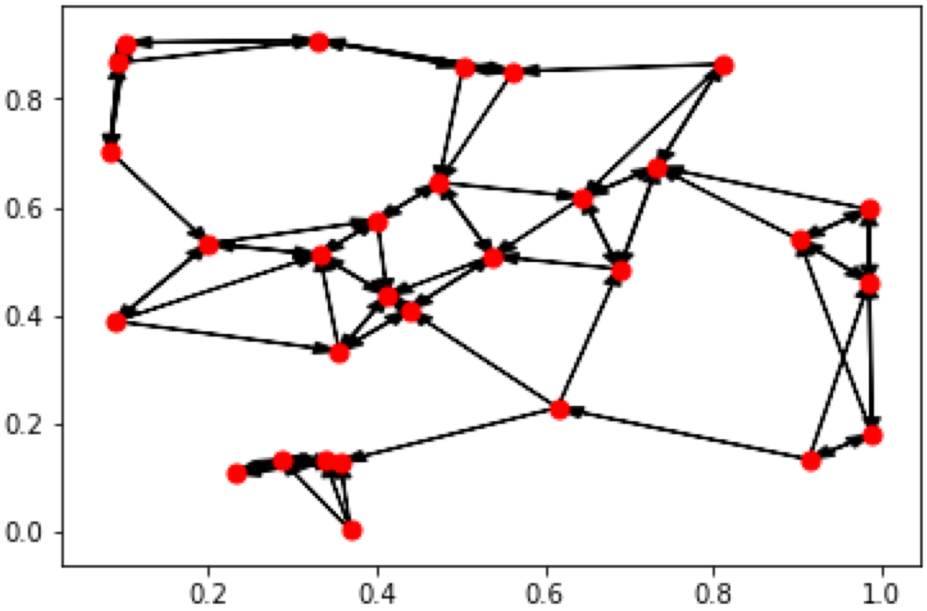
- Let $v_1, v_2, \ldots, v_n \in \mathbb{R}^d$ (can be some other metric space)
- Neighbourhood graph $G = (V, E)$:
  - $V = \{v_1, v_2, \ldots, v_n\}$
  - $(v_i, v_j) \in E$ if the two points are close
- Directed k-nearest neighbour graph:
  - Directed edge $(v_i, v_j) \in E$ if
    $$\|v_i - v_j\| > \|v_i - u\| \text{ for at most } k-1 \text{ other } u \in V \backslash \{v_i\}$$
- Undirected k-NN graphs:
  - standard k-NN: $(v_i, v_j) \in E$ if
    $$v_j \in kNN(v_i) \text{ or } v_i \in k\text{NN}(v_j)$$
  - mutual k-NN: $(v_i, v_j) \in E$ if
    $$v_j \in kNN(v_i) \text{ and } v_i \in k\text{NN}(v_j)$$

Neighbourhood graphs 2

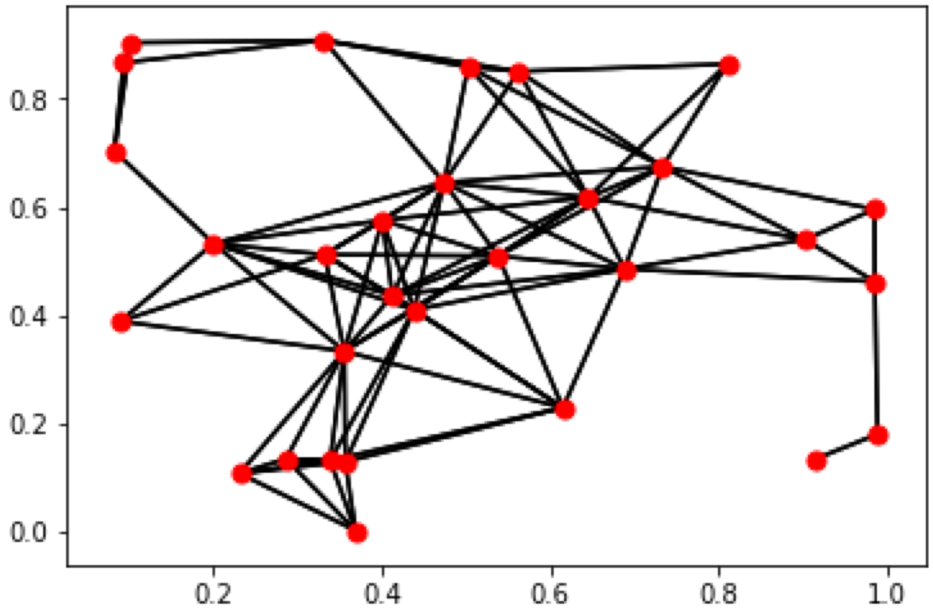
- ϵ-neighbourhood graph:
  - Undirected edge $(v_i, v_j) \in E$ if
$$\|v_i - v_j\| \leq \epsilon$$

- Example: $n = 30$ points in $[0, 1]^2$

Directed k-NN, $k = 3$



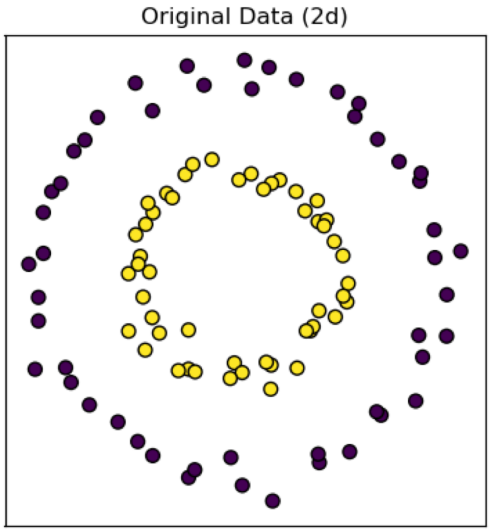
ϵ-neighbourhood, $\epsilon = 0.3$

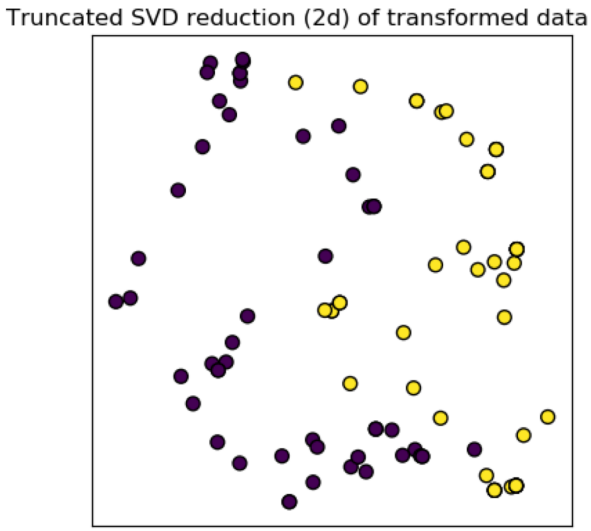


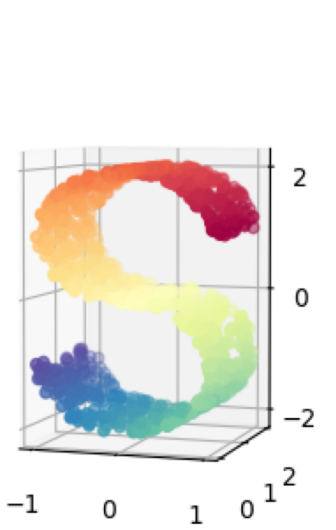
Motivation for geometric graphs 1

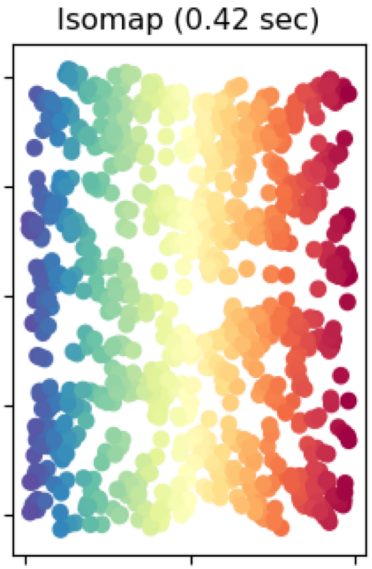
- Not a typical model for social networks
- Can model wireless or sensor networks
  - Two wireless devices communicate if they are close
- Useful for standard data analysis (not related to network data)
  - Big complex data often lie on manifolds
  - Does not span whole of $\mathbb{R}^d$
  - Example: Think of all possible 800×600 RGB images of cats — Can they be any arbitrary image in $[0, 255]^{800 \times 600 \times 3}$?
- It is often difficult to find / formally define these manifolds
  - Can we directly apply machine learning on the manifold?

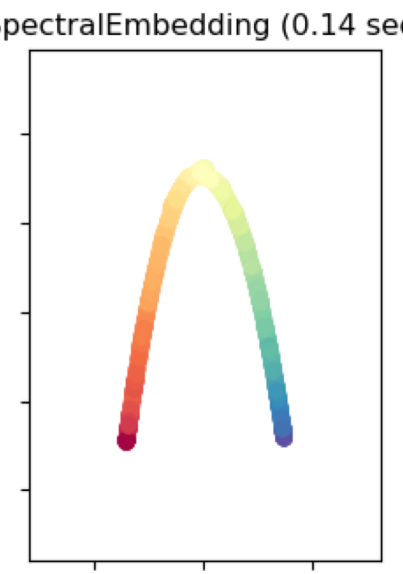
Motivation for geometric graphs 2











- Data lies in a specific region in $\mathbb{R}^d$ (manifold)
  - Circles in $\mathbb{R}^2$
  - S-curve in $\mathbb{R}^3$
- Machine learning
  - Choose features
  - Kernel trick (choose kernel)
- Graph based
  - Learn manifold from neighbourhood graph
  - Spectral embedding, Isomap

Random geometric graphs

Random geometric graphs (RGG)

- Various models for random graphs
  - Main component: Nodes are points in some space

- First type of RGGs
  - $f(\cdot)$ is a probability density on some space
    — example: Uniform distribuiton on unit cube / ball in $\mathbb{R}^d$
  - $v_1, \ldots, v_n \sim_{iid} f(\cdot)$
  - G is k-NN or ϵ-graph on $v_1, \ldots, v_n$

- Second type of RGGs
  - $V = \{v_1, \ldots, v_n\}$ lie in some space (may not be random)
  - $G = (V, E)$ undirected with edges being independent
  - $\mathbb{P}\big((v_i, v_j) \in E\big) \propto \exp\left(-\dfrac{\|v_i - v_j\|^2}{\sigma^2}\right)$ or $\dfrac{1}{\|v_i - v_j\|^\alpha}$
    (similar to Kleinberg's model)

Analysis of RGG 1

- Motivation for studying RGG
  - Partly mathematical interest
    — when are RGGs connected, have giant component etc.
  - Provides guidance for choice of parameters

Connectedness of undirected k-NN

Let $v_1, \ldots, v_n \sim f$, and G be undirected k-NN on them.
Under some conditions on f, with high probability

- G is connected if $k \gg \ln n$
- G is not connected if $k \ll \ln n$

Proof: Skipped.

Analysis of RGG 2

Implication of above result

- Fast NN search
  - n is large, and $V = \{v_1, \ldots, v_n\}$ entries in a database
  - Given query point v, find its nearest neighbour in V
  - Can we do in less than $O(n)$ time?
- k-NN graph based approximate NN search
  - 1. Start with a random $u \in V$
  - 2. If $\|v - u\| < \|v - x\|$ for all $x \in \mathrm{NN}(u)$
    — return u,
    — else repeat step-2 with $u^* = \underset{x \in \mathrm{NN}(u)}{\arg\min} \|v - x\|$
- Above algorithm cannot be accurate if graph is disconnected
  - Need to set $k \gg \ln n$

Analysis of RGG 3

Density estimation using ϵ-neighbourhood graph

Let $v_1, \ldots, v_n \sim_{iid} f$ on $\mathbb{R}^d$. Under some conditions on f and ϵ,

$$\mathbb{E}[\text{degree}(v_i)] \approx f(v_i) \cdot nC_d\epsilon^d$$

for large n, where C_d volume of unit ball in $\mathbb{R}^d$.

Proof idea: $\mathbb{E}[\text{degree}(v_i)] = \sum_{j \neq i} \mathbb{P}\big((v_i, v_j) \in E\big) = \sum_{j \neq i} \mathbb{P}\big(v_j \in B_\epsilon(v_i)\big)$

where $B_\epsilon(x) = \{y : \|x - y\| \leq \epsilon\}$

$$\mathbb{P}\big(v \in B_\epsilon(v_i)\big) = \int_{B_\epsilon(v_i)} f(v)\mathrm{d}v \approx f(v_i)\mathrm{Vol}\big(B_\epsilon(v_i)\big) \qquad \text{if } \epsilon \text{ is very small}$$

$$= f(v_i) \cdot C_d\epsilon^d$$

Spectral graph theory

References

- F. Chung. *Spectral graph theory*. Chapters 1, 2.
  `http://www.math.ucsd.edu/~fan/research/revised.html`

- U. von Luxburg. *A tutorial on spectral clustering*.
  `https://arxiv.org/pdf/0711.0189.pdf`

Spectra of graph Laplacians

Graph as a matrix

- Graphs can be represented by matrices
  - Adjacency matrix, Incidence matrix etc.
- Matrix spectral theory
  - Many properties of matrices depend on eigenvalues and eigenvectors
- Fan Chung writes:
  - *Roughly speaking, half of the main problems of spectral theory lie in deriving bounds on the distributions of eigenvalues. The other half concern the impact and consequences of the eigenvalue bounds as well as their applications.*
- Graph Laplacians
  - Other matrices defined from adjacency matrix
  - Spectra of Laplacians more useful than that of adjacency matrix

Unnormalised graph Laplacian

- For undirected graph $G = (V, E)$, let
  - $A \in \{0,1\}^{n \times n}$ (symmetric) is adjacency matrix
  - $D \in \mathbb{R}^{n \times n}$ (diagonal) is degree matrix, $D_{ii} = \text{degree}(i)$

- Unnormalised graph Laplacian, $L \in \mathbb{R}^{n \times n}$

$$L = D - A$$

- **Exercise:** For any vector $f \in \mathbb{R}^n$

$$(Lf)_i = \sum_{j \neq i} A_{ij}(f_i - f_j) \quad \text{and} \quad f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} A_{ij}(f_i - f_j)^2$$

- Think of f as a function $f : V \to \mathbb{R}$ that is, $f_i = f(v_i)$
  - $(f_i - f_j)$ = how much f changes across edge (i, j) ... derivative!!!

Why call it Laplacian?

- Laplace operator: Let $f : \mathbb{R}^2 \to \mathbb{R}$

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Computing Δf at a point $v = (x, y)$

$$\Delta f(v) = \frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y)$$

$$\approx \frac{f(x+h, y) + f(x-h, y) - 2f(x, y)}{h^2} + \frac{f(x, y+h) + f(x, y-h) - 2f(x, y)}{h^2}$$

(finite difference)

- Set $h = 1$, and think of $v = (x, y)$ as node in a grid graph
  - $(x \pm 1, y)$ and $(x, y \pm 1)$ are neighbours of v
  - L_{g} be Laplacian of grid graph
  - $\Delta f(v) \approx -L_{\mathrm{g}} f(v)$

Properties of unnormalised Laplacian 1

Properties of L

Let L be unnormalised Laplacian of undirected graph G

- L is symmetric
- L is positive semi-definite
- Smallest eigenvalue of L is 0, with corresponding eigenvector $\mathbf{1}_n = (1, 1, \ldots, 1)^T$

Proof:

1. A is symmetric, and D is diagonal. So $L = D - A$ is symmetric.

2. $f^T L f = \frac{1}{2} \sum_{i,j} A_{ij}(f_i - f_j)^2 \geq 0$ for all f

3. $(Lf)_i = \sum_{j \neq i} A_{ij}(f_i - f_j)$ for every i, and so $L\mathbf{1}_n = \mathbf{0} = 0 \cdot \mathbf{1}_n$

Properties of unnormalised Laplacian 2

Node relabelling does not change eigenvalues

Let G' be obtained from G by permuting node labels.
Let L be Laplacian of G, and L' or G'.

- Eigenvalues of L and L' are same.

Let $P \in \{0,1\}^{n \times n}$ be the permutation matrix for node relabelling, that is, $P_{i\pi_i} = 1$ if node-i in G is relabelled to node-π_i in G'

- $L' = P^T L P$
- If (λ, v) is eigenpair for $L \implies (\lambda, P^T v)$ is eigenpair for L'

Proof: Exercise.
Start with proof of $L' = P^T L P$. Everything follow from there.
Note: For permutation matrix P, $P^T P = P P^T = I$

Properties of unnormalised Laplacian 3

Eigenvalues of L and connectivity

Let L be unnormalised Laplacian of undirected graph G.
Let $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the n eigenvalues of L.

$$\lambda_2 > 0 \quad \text{if and only if} \quad G \text{ is connected}$$

Proof: Part 1 — Assume G is connected.

We show $\lambda_2 > 0$.
OR We show there is exactly one eigenvalue $= 0$
OR We show if $(0, f)$ is an eigenpair, then $f = c\mathbf{1}_n$

$(0, f)$ is eigenpair $\implies Lf = \mathbf{0}$

$$0 = f^T L f = \frac{1}{2} \sum_{i,j} A_{ij}(f_i - f_j)^2$$

Each term in sum is non-negative. So sum is zero if each term is zero.

Properties of unnormalised Laplacian 4

For every pair i, j, any one of these should hold:

- $A_{ij} = 0$, that is, $(i, j) \notin E$
- $f_i = f_j$

If there is a path $i_1, i_2, \ldots, i_\ell$

- $f_{i_1} = f_{i_2} = \ldots = f_{i_\ell}$

Since G is connected

- there is a path between any two nodes
- $f_i = f_j$ for all $i, j \implies f = c\mathbf{1}_n$

Properties of unnormalised Laplacian 5

Part 2 — Assume G is disconnected

We can split V into two disjoints sets V_1, V_2 so that there is no edge between V_1 and V_2

Let $\mathbf{1}_{V_1} \in \{0,1\}^n$ with i^{th} coordinate 1 if $i \in V_1$
and define $\mathbf{1}_{V_2}$ similar for V_2

Observe:

- $\mathbf{1}_{V_1}^T \mathbf{1}_{V_2} = 0$ (we also write as $\mathbf{1}_{V_1} \perp \mathbf{1}_{V_2}$)
- $L\mathbf{1}_{V_1} = \mathbf{0}$ and $L\mathbf{1}_{V_2} = \mathbf{0}$

There are two orthogonal eigenvectors for the eigenvalue 0
Hence, eigenvalue 0 has multiplicity at least two $\implies \lambda_2 = 0$.

Properties of unnormalised Laplacian 6

Eigenvalues of L and connected components

Let graph G has k connected components $V_1, \ldots, V_k$.

- L has exactly k zero eigenvalues
- The eigenspace of the eigenvalue 0 is spanned by $\mathbf{1}_{V_1}, \ldots, \mathbf{1}_{V_k}$

Let $G_1 = (V_1, E_1), \ldots, G_k = (V_k, E_k)$ be the k connected subgraphs. L_i be the Laplacian for G_i.

- Spectrum of L is union of the spectrum of $L_1, \ldots, L_k$

Note: For symmetric matrix M

- Spectrum of M = set of all eigenvalues of M
- Eigenspace of $\lambda = \{x : Mx = \lambda x\}$

Properties of unnormalised Laplacian 7

Proof: Start with last part. After reordering the nodes, we can write

$$L = \begin{pmatrix} L_1 & 0 & \cdots & 0 \\ 0 & L_2 & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & L_k \end{pmatrix}$$

Verify the following

- Let (λ, v) be an eigenpair for L_ℓ
- Define $\widetilde{v}$ as $\widetilde{v}_i = v_i$ if $i \in V_\ell$, and 0 otherwise
- Then $(\lambda, \widetilde{v})$ is eigenpair for L

Doing this for every eigenvalue of every L_ℓ proves last part.

First two statements follow from above since

- Each L_ℓ has exactly one eigenvalue 0
- Corresponding eigenvector is constant on V_ℓ

Properties of unnormalised Laplacian 8

Algebraic connectivity, λ_2

G is a connected graph, and λ_2 is smallest non-zero eigenvalue of L.

$$\lambda_2 = \min_{f \perp \mathbf{1}_n} \frac{\sum\limits_{i<j} A_{ij}(f_i - f_j)^2}{\sum\limits_i f_i^2}$$

Proof: Note $f^T L f = \frac{1}{2}\sum\limits_{i,j} A_{ij}(f_i - f_j)^2 = \sum\limits_{i<j} A_{ij}(f_i - f_j)^2$

So we have to show $\lambda_2 = \min\limits_{f \perp \mathbf{1}_n} \frac{f^T L f}{f^T f}$

To prove this, we need Rayleigh's principle (next slide).

The result follows by combining Rayleigh's principle with the fact $\lambda_1 = 0$ with corresponding eigenvector $\mathbf{1}_n$.

Properties of unnormalised Laplacian 9

Rayleigh's principle: (follows from spectral decomposition)

- Characterises eigenpairs as solution for optimisation problems
- For symmetric matrix $M \in \mathbb{R}^{n \times n}$:
  - Let eigenvalues be $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$
  - v_i be an eigenvector corresponding to λ_i
- $\lambda_1 = \min_{x \neq \mathbf{0}} \frac{x^T M x}{x^T x}$ and $\lambda_n = \max_{x \neq \mathbf{0}} \frac{x^T M x}{x^T x}$
- $\lambda_k = \min_{\substack{x \neq \mathbf{0} \\ x \perp v_1, \ldots, v_{k-1}}} \frac{x^T M x}{x^T x} = \max_{\substack{x \neq \mathbf{0} \\ x \perp v_{k+1}, \ldots, v_n}} \frac{x^T M x}{x^T x}$
- $v_k = \arg\min_{\substack{x \neq \mathbf{0} \\ x \perp v_1, \ldots, v_{k-1}}} \frac{x^T M x}{x^T x} = \arg\max_{\substack{x \neq \mathbf{0} \\ x \perp v_{k+1}, \ldots, v_n}} \frac{x^T M x}{x^T x}$

Normalised Laplacians

Two versions:

- Symmetric normalised graph Laplacian

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}$$

  - Spectrum of L_{sym} related to important graph properties
  - Key property:

$$f^T L_{sym} f = \frac{1}{2} \sum_{i,j} A_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 = \sum_{i<j} A_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

- Random walk graph Laplacian

$$L_{rw} = D^{-1} L = I - D^{-1} A$$

  - L_{rw} connected to random walks on graphs (will discuss later)

L_{sym} vs L_{rw}

Eigenvalues of L_{sym} and L_{rw}

1. L_{sym} is symmetric and positive semi-definite
2. L_{rw} may not be symmetric, but all eigenvalues are non-negative
3. (λ, x) eigenpair for $L_{rw} \iff (\lambda, D^{1/2}x)$ eigenpair for L_{sym}
4. $(0, \mathbf{1}_n)$ is an eigenpair for L_{rw}
5. $(0, D^{1/2}\mathbf{1}_n)$ is eigenpair for L_{sym}

Proof: Exercise.
1 — similar to L
3, 4, 5 — use definitions of L_{rw} and L_{sym}, and compute
2 — follows from 1 and 3

L_{sym} and connected components

L_{sym} and connected components

Let graph G has k connected components $V_1, \ldots, V_k$.

- L_{sym} has exactly k zero eigenvalues
- Eigenspace of the eigenvalue 0 is spanned by $D^{1/2}\mathbf{1}_{V_1}, \ldots, D^{1/2}\mathbf{1}_{V_k}$

Proof: Exercise. Similar to the unnormalised case.

Remark:
Many results, like above, for L, L_{sym}, L_{rw} also hold for **weighted** undirected graphs.
But all edge weights must be non-negative **(Why?)**

Eigenvalues of L_{sym} 1

Largest eigenvalue of L_{sym}

Let $\widetilde{\lambda}_n$ be the largest eigenvalue of L_{sym}

$$\|L_{sym}\|_2 = \widetilde{\lambda}_n \leq 2$$

Proof: Rayleigh's principle: $\widetilde{\lambda}_n = \max_{x \neq \mathbf{0}} \frac{x^T L_{sym} x}{x^T x} = \|L_{sym}\|_2 \quad$ (since $x^T L_{sym} x \geq 0$)

$$x^T L_{sym} x = \frac{1}{2} \sum_{i,j} A_{ij} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2$$

$$\leq \sum_{i,j} A_{ij} \cdot \left(\frac{x_i^2}{d_i} + \frac{x_j^2}{d_j} \right) \qquad \text{note: } (a+b)^2 \leq 2(a^2+b^2)$$

$$= \sum_i \sum_j A_{ij} \cdot 2 \frac{x_i^2}{d_i}$$

$$= \sum_i 2x_i^2 \quad = 2x^T x$$

Eigenvalues of L_{sym} 2

Smallest non-zero eigenvalue of L_{sym}

Let G be a connected graph.
Let $\widetilde{\lambda}_2$ be the smallest non-zero eigenvalue of L_{sym}

$$\widetilde{\lambda}_2 = \min_{f \perp D\mathbf{1}_n} \frac{\sum_{i<j} A_{ij}(f_i - f_j)^2}{\sum_i f_i^2 d_i} = \min_{f \perp D\mathbf{1}_n} \frac{f^T L f}{f^T D f}$$

Proof: Recall $\widetilde{\lambda}_1 = 0$ with eigenvector $D^{1/2}\mathbf{1}_n$

Rayleigh's principle:

$$\widetilde{\lambda}_2 = \min_{\substack{x \neq \mathbf{0} \\ x \perp D^{1/2}\mathbf{1}_n}} \frac{x^T L x}{x^T x} = \min_{\substack{x \neq \mathbf{0} \\ x \perp D^{1/2}\mathbf{1}_n}} \frac{\sum_{i<j} A_{ij} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2}{\sum_i x_i^2}$$

Replace x by f where $f_i = \frac{x_i}{\sqrt{d_i}}$. Check $\sum_i f_i d_i = 0$ $(f \perp D\mathbf{1}_n)$

Eigenvalues of L_{sym} 3

A key quantity in spectral graph theory:

- $\widetilde{\lambda}_2$ related to many interesting properties of graph

Bound on graph diameter

Let diam = diameter of unweighted graph $G = (V, E)$.

$$\text{diam} \geq \frac{1}{2|E| \cdot \widetilde{\lambda}_2}$$

If G is weighted, replace $2|E|$ by $\sum_i d_i$.

Eigenvalues of L_{sym} 4

Proof: Bound holds for disconnected graph.

Assume G is connected.

Recall $\widetilde{\lambda}_2 = \min_{f \perp D\mathbf{1}_n} \dfrac{\sum_{i<j} A_{ij}(f_i - f_j)^2}{\sum_i f_i^2 d_i}$

Let f achieves the minimum in above

- $f \perp D\mathbf{1}_n \Longrightarrow \sum_i f_i d_i = 0$
- Let $v = \arg\max_{i \in V} |f_i|$
- There is $u \in V$ such that $f_u f_v < 0$ (f_u, f_v have opposite signs)
- Let $P = (i_0, i_2, \ldots, i_\ell)$ be shortest path between $i_0 = u$ and $i_\ell = v$ — note: length of path $\ell \leq$ diam

Eigenvalues of L_{sym} 5

Note: $f_v - f_u = \sum_{k=0}^{\ell-1} f_{i_{k+1}} - f_{i_k} \implies (f_v - f_u)^2 \leq \ell \sum_{k=0}^{\ell-1} (f_{i_{k+1}} - f_{i_k})^2$

(using Cauchy-Schwarz)

Now
$$\widetilde{\lambda}_2 = \frac{\sum_{i<j} A_{ij}(f_i - f_j)^2}{\sum_i f_i^2 d_i}$$
$$\geq \frac{\sum_{i<j} A_{ij}(f_i - f_j)^2}{f_v^2 \sum_i d_i}$$
note: $\sum_i d_i = 2|E|$
$$\geq \frac{\sum_{(i,j)\in P} (f_i - f_j)^2}{f_v^2 \cdot 2|E|}$$
summing only over P, not all edges
$$\geq \frac{\frac{1}{\ell}(f_v - f_u)^2}{f_v^2 \cdot 2|E|} \geq \frac{1}{\ell \cdot 2|E|}$$
note: $(f_v - f_u)^2 > f_v^2$

$$\text{diam} \geq \ell \geq \frac{1}{2|E| \cdot \widetilde{\lambda}_2}$$

Cheeger constant 1

$G = (V, E)$ is an undirected graph

- unweighted or edges have non-negative weights A_{ij}

Let $S \subset V$ be a subset of nodes, and $\bar{S} = V \backslash S$

- Volume of a set: $\text{vol}(S) = \sum_{i \in S} d_i$
- Cut value: $\text{cut}(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} A_{ij}$
  - If G is unweighted, $\text{cut}(S, \bar{S})$ = #edges between S and$\bar{S}$
- Cheeger constant

$$h_G = \min_{S \subset V} h(S), \quad \text{where } h(S) = \frac{\text{cut}(S, \bar{S})}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}$$

Cheeger constant 2

h_G shows how well-connected a graph is

Examples (unweighted graphs):

- G is not connected $\implies h_G = 0$
- G is complete graph $\implies h_G \approx \frac{1}{2}$
- G is barbell graph $\implies h_G \approx \frac{4}{n^2}$

Cheeger cut (cut that achieves h_G) has:

- both sets $S, \bar{S}$ are large
  — high $\text{vol}(S)$ and $\text{vol}(\bar{S})$
- few connection between $S, \bar{S}$
  — small $\text{cut}(S, \bar{S})$

Cheeger constant 3

Cheeger inequality for graphs

$$\frac{\widetilde{\lambda}_2}{2} \leq h_G \leq \sqrt{2\widetilde{\lambda}_2}$$

Proof: We prove only $\frac{1}{2}\widetilde{\lambda}_2 \leq h_G$

Let $S, \bar{S}$ be a Cheeger cut, and define $f \in \mathbb{R}^n$ as

$$f_i = \frac{1}{\mathrm{vol}(S)} \text{ for } i \in S, \quad \text{and} \quad f_i = -\frac{1}{\mathrm{vol}(\bar{S})} \text{ for } i \in \bar{S}$$

$f \perp D\mathbf{1}_n$, and so $\widetilde{\lambda}_2 \leq \dfrac{f^T L f}{f^T D f} = \left(\dfrac{\mathrm{cut}(S,\bar{S})}{\mathrm{vol}(S)} + \dfrac{\mathrm{cut}(S,\bar{S})}{\mathrm{vol}(\bar{S})}\right) \leq 2h_G$

(exercise)

Will skip other part. If interested, see Chung's book (Theorem 2.2) or ML lecture slides.

Communities in networks

References

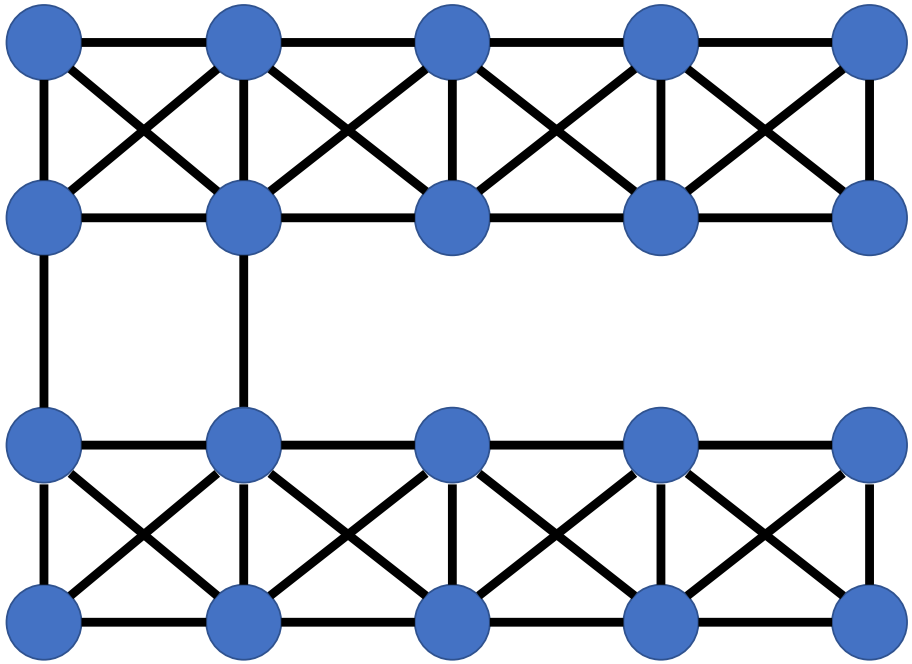
- U. von Luxburg. *A tutorial on spectral clustering.* `https://arxiv.org/pdf/0711.0189.pdf`

- M. E. J. Newman. *Spectral methods for network community detection and graph partitioning.* `https://arxiv.org/pdf/1307.7729.pdf`

Spectral clustering

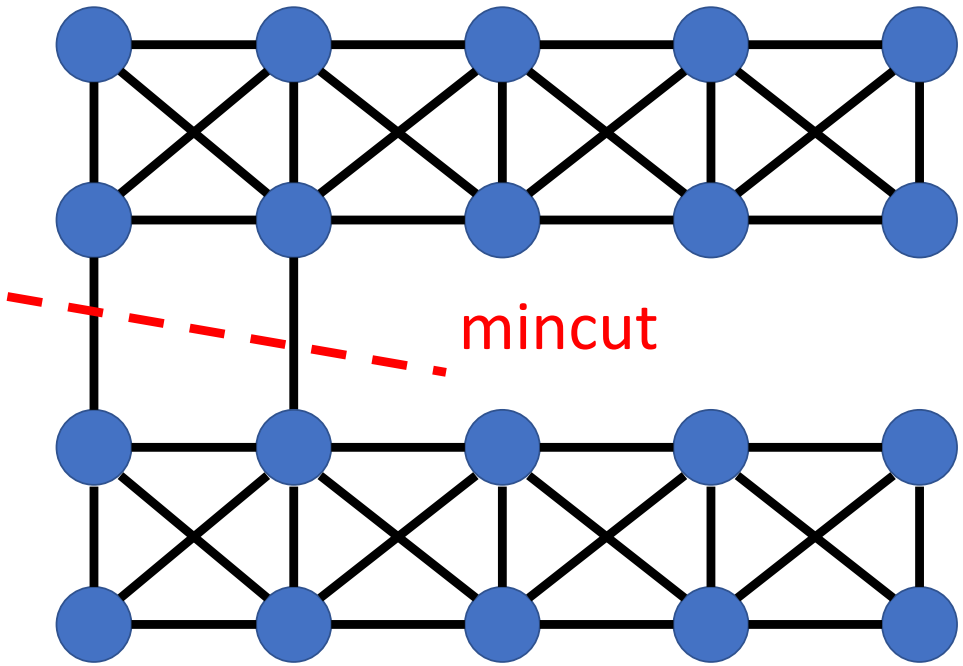
How do we define communities?

- Many edges within each community
- Few edges between two communities
- Which are the communities here?
  — How do we find them (algorithmically)?



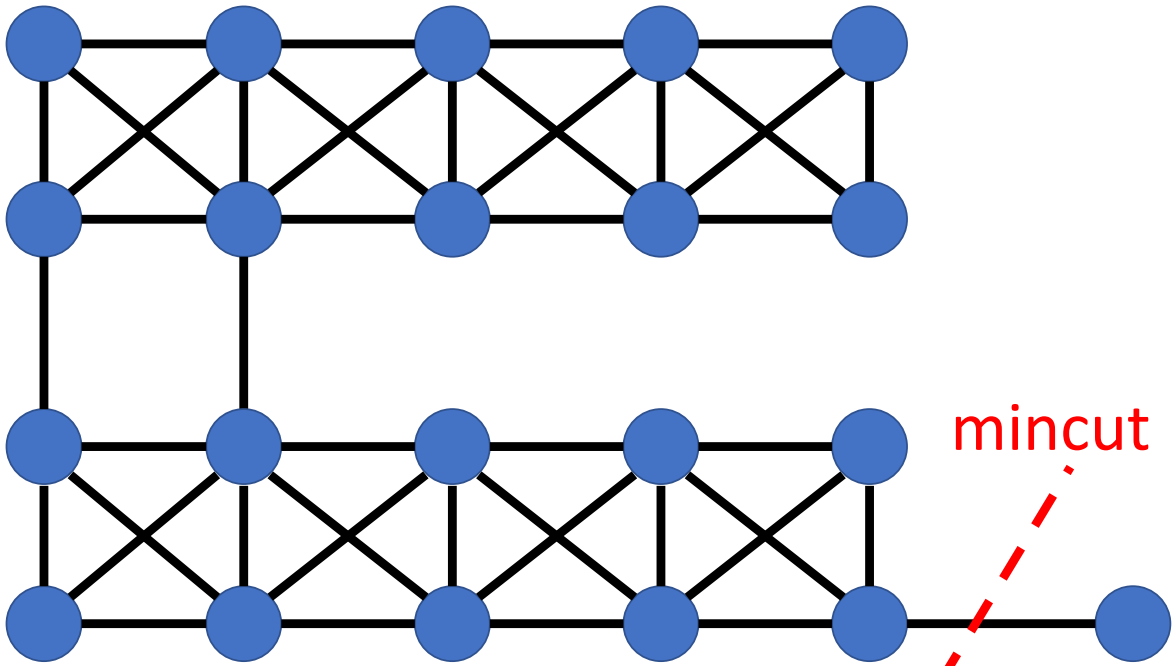
The minimum cut approach 1

- $\text{cut}(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} A_{ij}$
- Find $S = \arg\min_{S \subset V} \text{cut}(S, \bar{S})$
  - Meaning:
    Remove the minimum number of edges so that graph is disconnected



The minimum cut approach 2

- Mincut can be solved in polynomial time
- May not produce balanced partition



- Balanced partition helps when we want to split the network
  - for storage
  - for easier / faster network analysis

Balanced graph partitioning 1

- **Approach 1:** Add constraints to make sets nearly equal

  Balanced mincut:

$$\min_{S \subset V} \ \text{cut}(S, \bar{S})$$
$$\text{s.t. } |S| \leq (1+\epsilon)\frac{|V|}{2}$$
$$|\bar{S}| \leq (1+\epsilon)\frac{|V|}{2} \qquad \big(\text{note: } |S| = \ \#\text{nodes in S}\big)$$

- Balanced mincut is a NP-hard problem

Balanced graph partitioning 2

- **Approach 2:** Modify objective to induce balancing

  Cheeger cut: $h(S) = \dfrac{\text{cut}(S, \bar{S})}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}$

  Normalised cut: $\text{N-Cut}(S, \bar{S}) = \text{cut}(S, \bar{S}) \left(\dfrac{1}{\text{vol}(S)} + \dfrac{1}{\text{vol}(\bar{S})} \right)$

  Ratio cut: $\text{R-Cut}(S, \bar{S}) = \text{cut}(S, \bar{S}) \left(\dfrac{1}{|S|} + \dfrac{1}{|\bar{S}|} \right)$

- The terms $|S|$ or $\text{vol}(S)$ make partition more balanced ... **Why?**
- Minimising these objectives are also NP-hard
  - But, we can relax the optimisation problem (for N-Cut, R-Cut)

Spectral relaxation of Ratio Cut 1

- Min R-Cut problem:

$$\min_{S \subset V} \text{ R-Cut}(S, \bar{S})$$

- Re-writing the objective: Let $f_s \in \mathbb{R}^n$ such that

$$(f_s)_i = \begin{cases} \sqrt{\dfrac{|\bar{S}|}{|V| \cdot |S|}} & \text{if } i \in S \\ -\sqrt{\dfrac{|S|}{|V| \cdot |\bar{S}|}} & \text{if } i \in \bar{S} \end{cases}$$

Exercise: Show that $\text{R-Cut}(S, \bar{S}) = f_s^T L f_s$

- Min R-Cut problem (rephrased):

$$\min_{S \subset V} \ f_s^T L f_s$$

Spectral relaxation of Ratio Cut 2

- **Exercise:** Verify that $\|f_s\|_2 = 1$ and $f_s \perp \mathbf{1}_n$ for any S
- Relaxation:
  - f need not have the exact structure of f_s for some S
  - We still impose the constraints $\|f\|_2 = 1$ and $f \perp \mathbf{1}_n$
- Relaxed R-Cut problem:

$$\begin{aligned} \min_{f \in \mathbb{R}^n} \quad & f^T L f \\ \text{s.t.} \quad & f \perp \mathbf{1}_n \\ & \|f\|_2 = 1 \end{aligned}$$

- Spectral connection: What is the optimal f for above problem?

Spectral relaxation of Ratio Cut 3

Unnormalised Spectral Clustering: . . . for bi-partitioning

1. Compute f = eigenvector for second smallest eigenvalue of L
2. Let $S = \{i : f_i \geq 0\}$, and $\bar{S} = V \backslash S$
   . . . split based on intuition from f_s

Remark:

- If graph has 2 connected components, the algorithm returns them
- What happens if graph has more than 2 connected components?

k-way partitioning 1

- R-Cut for bi-partitioning:
  Let $S_1 = S$ and $S_2 = \bar{S} = \bar{S}_1$

$$\text{R-Cut}(S_1, S_2) = \frac{\text{cut}(S_1, S_2)}{|S_1|} + \frac{\text{cut}(S_1, S_2)}{|S_2|}$$

- R-Cut for k-way partitioning:
  Let $V = S_1 \cup S_2 \cup \ldots \cup S_k$, where $S_j \cap S_\ell = \emptyset$

$$\text{R-Cut}(S_1, \ldots, S_k) = \sum_{\ell=1}^{k} \frac{\text{cut}(S_\ell, \bar{S}_\ell)}{|S_\ell|}$$

- How do we write R-Cut$(S_1, \ldots, S_k)$ in terms of L?

k-way partitioning 2

- Define $f_1, \ldots, f_k \in \mathbb{R}^n$ such that

$$f_\ell = \frac{\mathbf{1}_{S_\ell}}{\sqrt{|S_\ell|}} \qquad \text{that is} \qquad (f_\ell)_i = \begin{cases} \sqrt{\dfrac{1}{|S_\ell|}} & \text{if } i \in S_\ell \\ 0 & \text{otherwise} \end{cases}$$

- **Exercise:** Let $F = [f_1, \ldots, f_k] \in \mathbb{R}^{n \times k}$. Show that
  - $\|f_\ell\|_2 = 1$ and $f_\ell \perp f_j$ for $\ell \neq j$, that is, $F^T F = I$
  - $f_\ell^T L f_\ell = \dfrac{\text{cut}(S_\ell, \bar{S}_\ell)}{|S_\ell|}$
  - $\text{R-Cut}(S_1, \ldots, S_k) = \sum_{\ell=1}^{k} f_\ell^T L f_\ell = \text{Trace}(F^T L F)$

k-way partitioning 3

- Min R-cut problem:

$$\min_{S_1,\ldots,S_k} \text{R-Cut}(S_1,\ldots,S_k)$$

OR

$$\min_{F\in\mathbb{R}^{n\times k}} \text{Trace}(F^T L F)$$

$$\text{s.t. } F = [f_1,\ldots,f_k] \text{ has above structure}$$

- Relaxed R-cut problem:

$$\min_{F\in\mathbb{R}^{n\times k}} \text{Trace}(F^T L F)$$

$$\text{s.t. } F^T F = I$$

- Solution: F = matrix of k leading orthonormal eigenvectors of L
  . . . corresponding to k smallest eigenvalues

k-way partitioning 4

Notation: Let $F_{i\bullet}$ be i^{th} row of matrix F

Unnormalised Spectral Clustering: . . . for k-way partitioning

1. Compute F = matrix of k leading orthonormal eigenvectors of L
2. Normalise each row of F, that is, let $\widetilde{F} \in \mathbb{R}^{n \times k}$
$$\widetilde{F}_{i\bullet} = \frac{F_{i\bullet}}{\|F_{i\bullet}\|_2}$$
3. Think of $\widetilde{F}_{1\bullet}, \ldots, \widetilde{F}_{n\bullet}$ as n points in $\mathbb{R}^k$
— Use k-means clustering to group them into k clusters
4. Let $S_\ell = \{i : \widetilde{F}_{i\bullet}$ grouped into ℓ^{th} cluster$\}$

k-way partitioning 5

Intuition for row normalisation and clustering:

- Let $F = [f_1, \ldots, f_k]$, where $f_\ell = \dfrac{\mathbf{1}_{S_\ell}}{\sqrt{|S_\ell|}}$
- After normalisation: $\widetilde{F}_{i\ell} = \mathbf{1}\{i \in S_\ell\}$

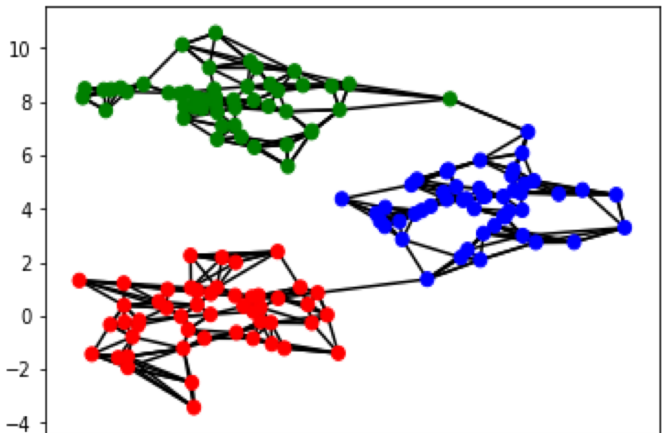
  $\widetilde{F} = [\mathbf{1}_{S_1}, \ldots, \mathbf{1}_{S_k}] \in \{0,1\}^{n \times k}$ . . . cluster assignment matrix
- If we cluster rows of above $\widetilde{F}$, the clusters correspond to $S_1, \ldots, S_k$

Performance of unnormalised spectral clustering 1

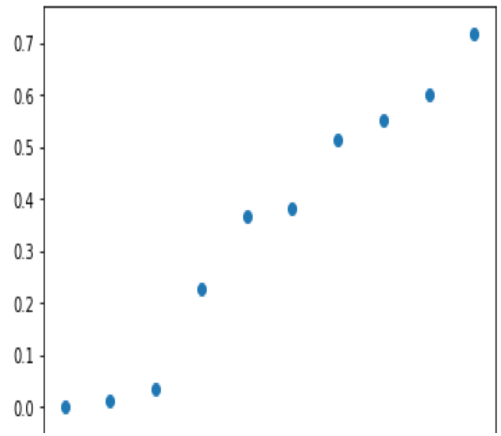
Example:

- Points sampled from mixture of 3 Gaussians
- 3-way partitioning of k-NN graph
- Note: Smallest three eigenvalues of L are close to 0
- 3-D plot of $\widetilde{F}_{1\bullet}, \ldots, \widetilde{F}_{n\bullet}$ shows the 3 groups

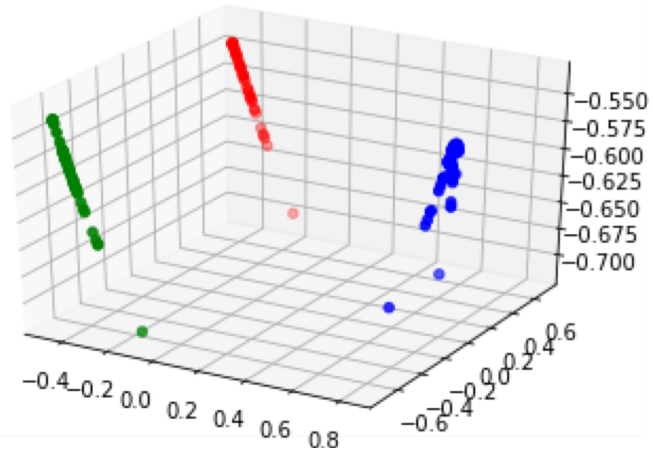
5-NN graph



$\lambda_1, \ldots, \lambda_{10}$



$\widetilde{F}_{1\bullet}, \ldots, \widetilde{F}_{n\bullet}$

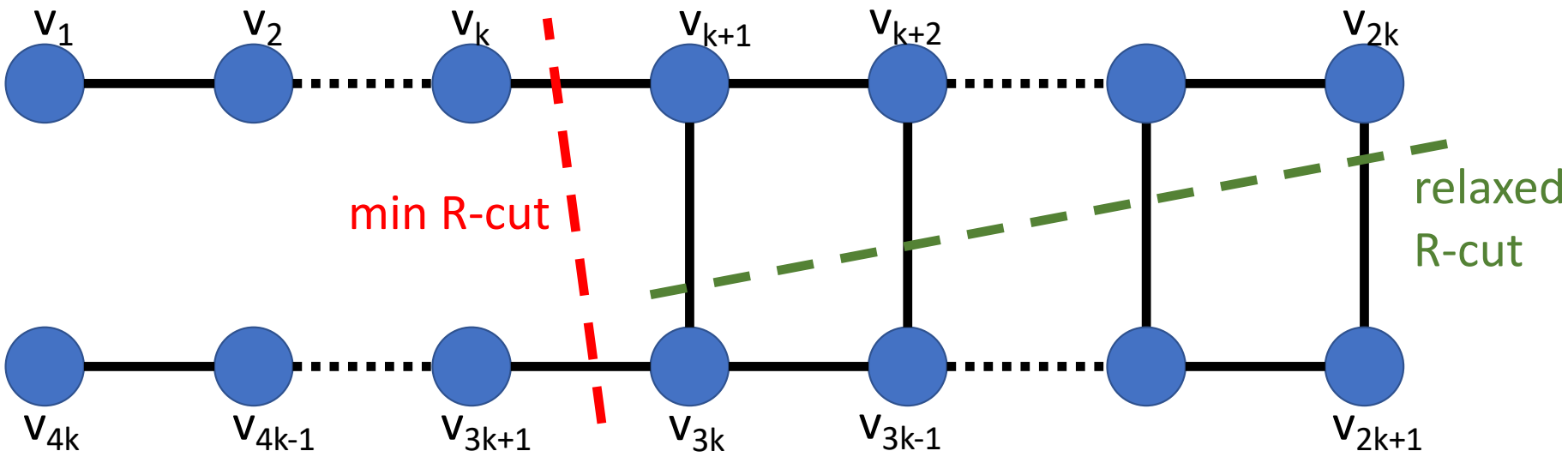


Performance of unnormalised spectral clustering 2

- In general, no guarantee that

  solution of spectral relaxation = optimal R-Cut

- Example: Cockroach graph on $n = 4k$ nodes
  - optimal R-Cut value $= \frac{2}{k}$
  - R-Cut value for spectral solution $= 1$



Performance of unnormalised spectral clustering 3

- Guarantees hold for graphs that have some community structure
  [Peng, Sun & Zanetti, COLT-2015; Rohe, Chatterjee & Yu, Ann. Stat.-2011]

- Below is a result for stochastic block model

  $G \sim \mathrm{SBM}(s, k, p, q)$ if:
  - $V = S_1 \cup S_2 \cup \ldots \cup S_k$ with $|S_\ell| = s$
  - All edges are independent with

$$\mathbb{P}\big((i,j) \in E\big) = \begin{cases} p & \text{if } i,j \in S_\ell \text{ for some } \ell \\ q & \text{if } i,j \text{ belong to different groups} \end{cases}$$

Consistency of spectral clustering

Let $p, q \in (0,1)$ with $p > q$. Let $G \sim \mathrm{SBM}(\frac{n}{2}, 2, p, q)$.

Unnormalised spectral clustering outputs the underlying split with probability $1 - o(1)$ as $n \to \infty$.

Normalised cut 1

- N-Cut for bi-partitioning:

$$\text{N-Cut}(S, \bar{S}) = \frac{\text{cut}(S, \bar{S})}{\text{vol}(S)} + \frac{\text{cut}(S, \bar{S})}{\text{vol}(\bar{S})}$$

- N-Cut for k-way partitioning:
  Let $V = S_1 \cup S_2 \cup \ldots \cup S_k$, where $S_j \cap S_\ell = \emptyset$

$$\text{N-Cut}(S_1, \ldots, S_k) = \sum_{\ell=1}^{k} \frac{\text{cut}(S_\ell, \bar{S}_\ell)}{\text{vol}(S_\ell)}$$

$$= \sum_{\ell=1}^{k} f_\ell^T L f_\ell \qquad \text{where } f_\ell = \frac{\mathbf{1}_{S_\ell}}{\sqrt{\text{vol}(S_\ell)}}$$

- Note: $f_\ell^T D f_\ell = 1$ and $f_j^T D f_\ell = 0$ for $j \neq \ell$

Normalised cut 2

- Let $F = [f_1, \ldots, f_k]$, and $U = D^{1/2}F$

- Relaxed N-cut problem:

$$\min_{F \in \mathbb{R}^{n \times k}} \ \text{Trace}(F^T L F)$$
$$\text{s.t. } F^T D F = I$$

OR:

$$\min_{U \in \mathbb{R}^{n \times k}} \ \text{Trace}(U^T L_{sym} U)$$
$$\text{s.t. } U^T U = I$$

- Spectral solution:
  U = matrix of k leading orthonormal eigenvectors of L_{sym}
  . . . corresponding to k smallest eigenvalues

Normalised cut 3

Normalised Spectral Clustering: . . . for k-way partitioning

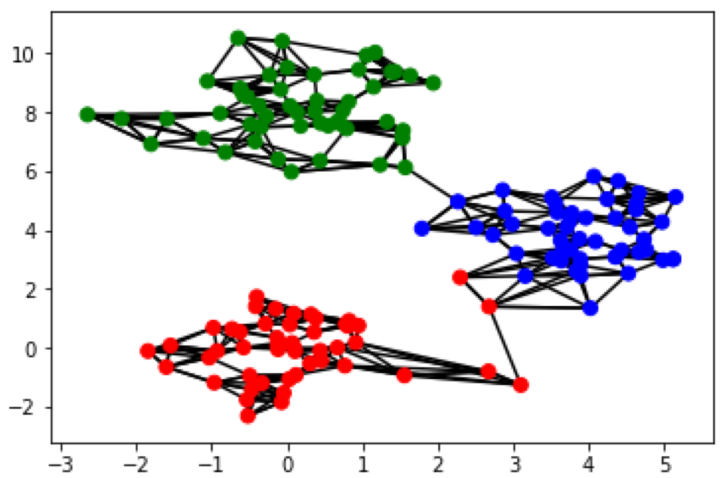
1. U = matrix of k leading orthonormal eigenvectors of L_{sym}
2. Normalise each row of U, that is, let $\widetilde{U} \in \mathbb{R}^{n \times k}$
$$\widetilde{U}_{i\bullet} = \frac{U_{i\bullet}}{\|U_{i\bullet}\|_2}$$
3. Use k-means clustering to group $\widetilde{U}_{1\bullet}, \ldots, \widetilde{U}_{n\bullet} \in \mathbb{R}^k$ into k clusters
4. Let $S_\ell = \{i : \widetilde{U}_{i\bullet}$ grouped into ℓ^{th} cluster$\}$

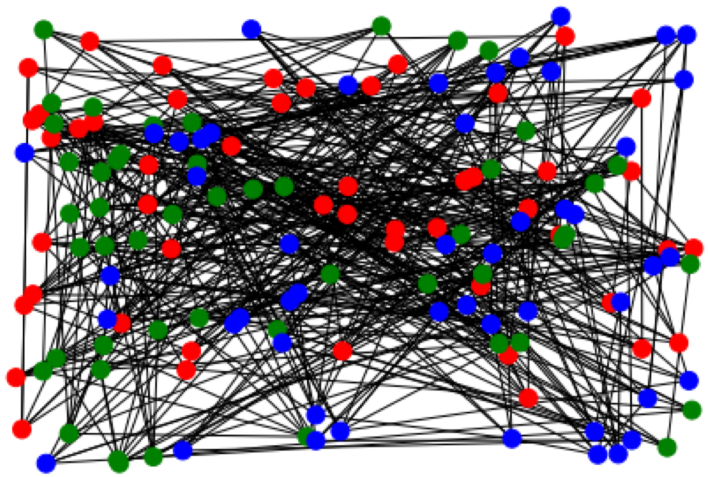
Remark:

- Spectral clustering is one way to relax R-Cut / N-Cut problem
- Another popular relaxation: Semi-definite programming

Modularity

Does a network have communities?





Spectral graph theory: $0 = \widetilde{\lambda}_1 \leq \widetilde{\lambda}_2 \leq \ldots \leq \widetilde{\lambda}_n$ eigenvalues of L_{sym}

- $\widetilde{\lambda}_k = 0 \implies G$ has at least k disjoint communities
- Cheeger's inequality: $h_G \leq \sqrt{2\widetilde{\lambda}_2}$

  Small $\widetilde{\lambda}_2 \implies G$ has ≥ 2 sparsely connected communities
- Higher-order Cheeger inequality: [Lee, Gharan & Trevisan, 2011]

  Small $\widetilde{\lambda}_k \implies G$ has $\geq k$ sparsely connected communities

Modularity 1

- Statistical approach for quantifying *community*-ness of network
- Community: Sub-group of nodes
  — More connection among themselves than outside community
- Which of the following have communities?
  - $G \sim G(n, p)$
  - $G \sim \mathrm{SBM}(s, k, p, q)$ for $p > q$
  - $G \sim \mathrm{CRM}(d_1, \ldots, d_n)$
- Recall: For $G \sim \mathrm{CRM}(d_1, \ldots, d_n)$

$$\mathbb{P}\big((i,j) \in E\big) \approx \frac{d_i d_j}{2m}$$

Modularity 2

- Let $S \subset V$, and $G_S = (S, E_S)$ be the sub-graph on S

- Under Configuration model

$$\mathbb{E}\big[|E_S|\big] = \sum_{\substack{i,j \in S \\ i<j}} \mathbb{P}\big((i,j) \in E\big) \approx \sum_{\substack{i,j \in S \\ i<j}} \frac{d_i d_j}{2m}$$

  Call S a community if $|E_S| \gg \mathbb{E}\big[|E_S|\big]$

- Let $S_1, \ldots, S_k$ be partition of V
  — $\psi : V \to \{S_1, \ldots, S_k\}$ is cluster assignment function

$$\text{Modularity}(S_1, \ldots, S_k) = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \mathbf{1}\{\psi(i) = \psi(j)\}$$

$$\approx \frac{1}{m} \sum_{\ell=1}^{k} |E_{S_\ell}| - \mathbb{E}\big[|E_S|\big]$$

Modularity maximisation 1

- What is the maximum modularity for a k-way partitioning?
- Modularity matrix, $B \in \mathbb{R}^{n \times n}$

$$B_{ij} = A_{ij} - \frac{d_i d_j}{2m} \qquad \text{or} \qquad B = A - \frac{dd^T}{2m} \qquad d = (d_1, \ldots, d_n)^T$$

- $\text{Modularity}(S_1, \ldots, S_k) = \dfrac{1}{2m} \displaystyle\sum_{\ell=1}^{k} \mathbf{1}_{S_\ell}^T B \mathbf{1}_{S_\ell}$ **(Verify)**
- Modularity maximisation:

$$\max_{F \in \mathbb{R}^{n \times k}} \frac{1}{2m} \text{Trace}(F^T B F)$$
$$\text{s.t. } F = [\mathbf{1}_{S_1} \ldots \mathbf{1}_{S_k}] \qquad \ldots \text{again NP-hard}$$

Modularity maximisation 2

- We relax the problem . . . will look at $k = 2$ case only
- For $S \subset V$, let $s \in \{-1, +1\}^n$

$$s_i = \begin{cases} +1 & \text{if } i \in S \\ -1 & \text{if } i \in \bar{S} \end{cases}$$

$$\text{Modularity}(S, \bar{S}) = \frac{1}{2m} \sum_{i,j} B_{ij} \frac{s_i s_j + 1}{2}$$

$$\text{note: } \frac{s_i s_j + 1}{2} = \mathbf{1}\{\psi(i) = \psi(j)\}$$

- Maximum modularity split:

$$s^* = \underset{s \in \{-1,+1\}^n}{\arg\max} \; s^T B s$$

Observe: $\|s\|_2 = \sqrt{n}$

Modularity maximisation 3

- Relaxing the hard constraint:

$$\widehat{s} = \underset{\|s\|_2 = \sqrt{n}}{\arg\max}\ s^T B s$$

- Spectral solution:
  $\widehat{s}$ = eigenvector of B corresponding to largest eigenvalue

Spectral modularity maximisation: . . . for bi-partitioning

1. Compute $\widehat{s}$ = eigenvector of B corresponding to largest eigenvalue
2. Let $S = \{i : \widehat{s}_i \geq 0\}$ and $\bar{S} = V \backslash S$

Graph embedding and visualisation

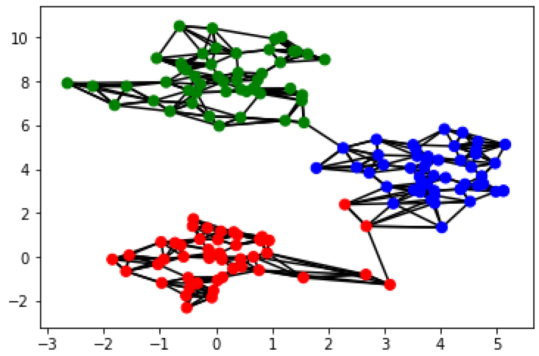
References

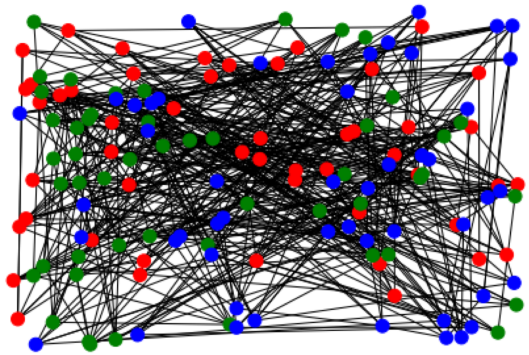
- Surveys for graph embedding algorithms:
  https://arxiv.org/pdf/1705.02801.pdf
  https://arxiv.org/pdf/1709.07604.pdf
- Fruchterman-Reingold method for visualisation:
  http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.8444&rep=rep1&type=pdf
- MDS, Isomap and more:
  https://www.math.uwaterloo.ca/~aghodsib/courses/f06stat890/readings/tutorial_stat890.pdf

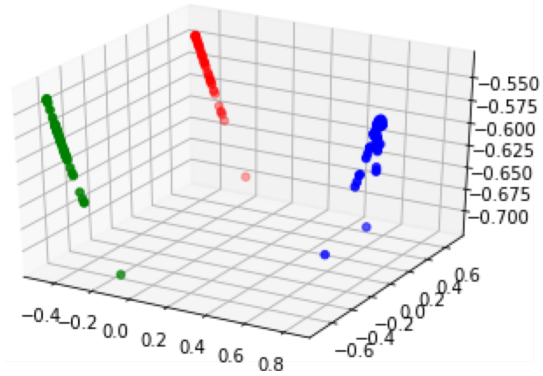
Graph embedding (spectral methods)

Graph embedding

- Which representation says something about the network structure?







- Graph embedding:
  - Find points $x_1, \ldots, x_n \in \mathbb{R}^p$, where x_i = location of node-i
  - Representation should reflect graph structure
  - Example: $\widetilde{F}_{1\bullet}, \ldots, \widetilde{F}_{n\bullet}$ in spectral clustering

- Graph drawing / visualisation:
  - Embed graph in $\mathbb{R}^2$ or $\mathbb{R}^3$ . . . and a bit more

Laplacian embedding

- Idea: x_i and x_j should be close if $(i, j) \in E$
- An optimisation problem

$$\min_{x_1,\ldots,x_n \in \mathbb{R}^d} \sum_{i<j} A_{i,j} \|x_i - x_j\|^2$$

- $\sum_{i<j} A_{ij} \|x_i - x_j\|^2 = \frac{1}{2} \text{Trace}\left(X^T L X\right)$, where $X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$
- Without constraint, we get trivial solution $X = \mathbf{0}$
  — Add constraint, $X^T X = I$ or $X^T D X = I$
- Solution:
  Leading p eigenvectors of L or L_{sym} (depending on constraint)

Local linear embedding

- Idea: Draw every node at the centre of its neighbours

$$x_i \approx \frac{1}{d_i} \sum_{j \sim i} x_j$$

- May not be achieved, and so, optimise

$$\min_{x_1,\ldots,x_n \in \mathbb{R}^d} \sum_{i=1}^{n} \left\| x_i - \sum_{j \neq i} A_{ij} x_j \right\|^2 = \left\| X - D^{-1} A X \right\|_F^2$$

- Can be re-written as

$$\min_{X \in \mathbb{R}^{n \times d}} \text{Trace}\left(X^T L_{rw}^T L_{rw} X \right)$$

$$\text{s.t. } X^T X = I \qquad \text{(avoids trivial solution)}$$

- Solution: Leading p eigenvectors of $L_{rw}^T L_{rw}$

Applications of graph embedding

- Visualising a network
- Apply standard machine learning tools on networks
- Community detection = clustering Laplacian embedding of nodes
- Anomalous nodes = outlier detection in embedded nodes
- Semi-supervised learning:
  Given labels of few nodes, infer those of other nodes
- Big picture: Graph embedding = feature learning for nodes

Force-based algorithms

Graph visualisation

- Similar to embedding into $\mathbb{R}^2$ or $\mathbb{R}^3$
  - For $p > 3$, we cannot obviously visualise
- The different layout methods in NetworkX
- We may use LLE or Laplacian embedding
  - May not be good for visualisation
  - Let G = union of 2 disjoint cliques
    — What is its embedding in $\mathbb{R}^2$?
- Requirements of a good visualisation
  - Nodes should not overlap, and well spread
  - Adjacent nodes close, non-adjacent nodes far
  - Densely connected communities clearly visible

Fruchterman-Reingold algorithm 1

- Force based drawing
  - Place adjacent nodes close, but not too close
  - Based on a physical laws of attraction and repulsion
- Let $x_1, \ldots, x_n \in \mathbb{R}^2$ be the locations of the nodes
- Every pair of nodes repel each other

$$f_r(u, v) = \frac{k^2}{\|x_u - x_v\|}$$

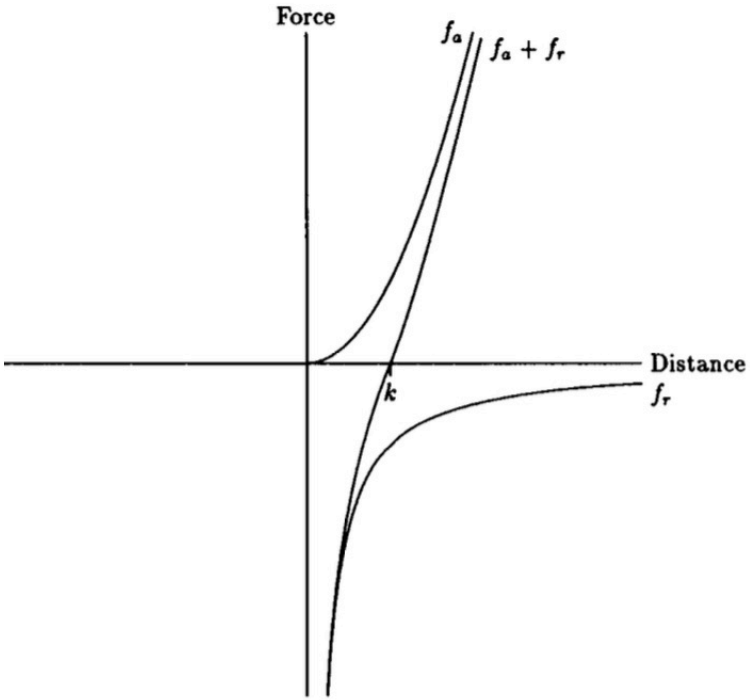
  - $f_r(u, v) = \infty$ if $x_u = x_v$
  - Nodes cannot overlap, and well spread

Fruchterman-Reingold algorithm 2

- Adjacent nodes also pull each other closer

$$f_a(u, v) = \frac{\|x_u - x_v\|^2}{k}$$

  - $f_a(u, v)$ large if $(u, v) \in E$, but $\|x_u - x_v\|$ large
  - Adjacent nodes tend to be close



[Image: Fruchterman & Reingold]

Fruchterman-Reingold algorithm 3

- Forces that act on node-u
  - For every v such that $(u, v) \notin E$

$$F_u(v) = f_r(u, v)\overrightarrow{x_v x_u}$$

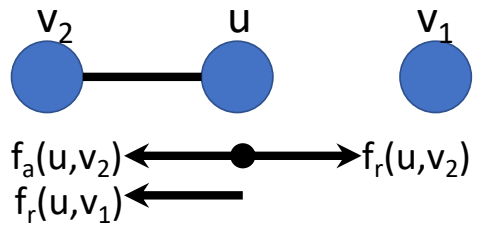
($\overrightarrow{x_v x_u}$ = unit vector along x_v to x_u)

  - For every v such that $(u, v) \in E$

$$F_u(v) = f_r(u, v)\overrightarrow{x_v x_u} + f_a(u, v)\overrightarrow{x_u x_v}$$

  - All forces on u must cancel each other (at equilibrium)

$$\sum_{v \neq u} F_u(v) = 0$$



  - Solving this for every u provides the location $x_1, \ldots, x_n$
    - Algorithm skipped (see reference if interested)

Isomap

Metric multi-dimensional scaling (metric MDS)

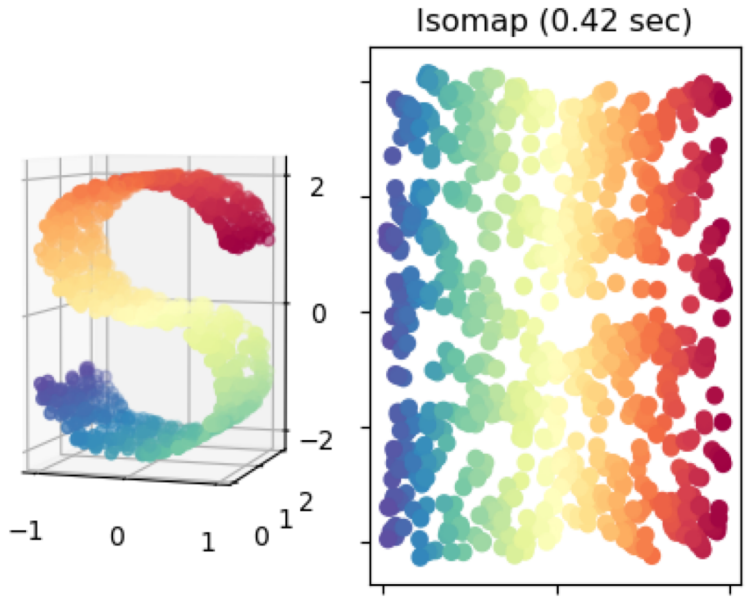
- General technique for embedding data
- Let $v_1, \ldots, v_n$ be points in a metric space $(V, \mathtt{d})$
  - We do not observe the points
  - But we know the pairwise distances $\mathtt{d}(v_i, v_j)$ for all i, j
- Metric MDS problem:
  - Find points $x_1, \ldots, x_n \in \mathbb{R}^p$ that optimize
$$\min_{x_1,\ldots,x_n} \sum_{i<j} \left(\mathtt{d}(v_i, v_j) - \|x_i - x_j\|\right)^2$$
  - Can replace $\|\cdot\|$ by another metric to embed in a different space
- Can we embed graphs using MDS?

Metric multi-dimensional scaling (metric MDS)

- General technique for embedding data
- Let $v_1, \ldots, v_n$ be points in a metric space $(V, \mathtt{d})$
  - We do not observe the points
  - But we know the pairwise distances $\mathtt{d}(v_i, v_j)$ for all i, j
- Metric MDS problem:
  - Find points $x_1, \ldots, x_n \in \mathbb{R}^p$ that optimize
$$\min_{x_1,\ldots,x_n} \sum_{i<j} \left(\mathtt{d}(v_i, v_j) - \|x_i - x_j\|\right)^2$$
  - Can replace $\|\cdot\|$ by another metric to embed in a different space
- Can we embed graphs using MDS?
  - Let V be the vertex set, and $\mathtt{d}$ = shortest path distance
  - Kamada-Kawai layout: variant of this approach

Isomap 1

- Isomap estimates "*intrinsic geometry of a data manifold*"
- Example:
  - There is a S-curve in $\mathbb{R}^3$
  - Points inside the S-curve are uniformly distributed
  - How can we verify uniformity given the points in 3-dim?
    OR Apply ML on this data?



- Here, points lie on a low-dimensional manifold
- Isomap shows how the points are distributed on this manifold

Isomap 2

Method

1. Generate k-NN graph from the points
2. Compute $\mathtt{d}_{sp}(u, v)$ for every u, v
3. Embed the data in a lower dimensional space using metric MDS

Remarks: We use graph . . .

- as intermediate step to embed data into low-dimensional space
- more generally, as a tool for manifold learning

Random walks on graphs

References

- L. Lovasz. *Random walks on graphs: A survey*
  `http://web.cs.elte.hu/~lovasz/erdos.pdf`

- F. Chung and W. Zhao. *PageRank and random walks on graphs*
  `http://www.math.ucsd.edu/~fan/wp/lov.pdf`

Basics of random walk

Random walk on graph

- $G = (V, E)$ is an undirected unweighted graph
- We start from node v_0 at time $t = 0$
- At time $t = 1$, randomly pick a neighbour of X_0 and move there. Call it X_1
- At time t, randomly pick a neighbour of X_{t-1} and move there. Call it X_t
- $(X_0, X_1, X_2, \ldots)$ is a random walk on G

Markov chain and random walk 1

- Let $\{X_t\}$ be a collection of random variables, indexed by parameter t
  — Think of t as time: $t \in \mathbb{R}$ or $t = 0, 1, 2, \ldots$

- X_t takes values in $\mathcal{S}$ (called states of chain)

- Independent trial process:
  For every $t_1, \ldots, t_k$ distinct, $X_{t_1}, \ldots, X_{t_k}$ are mutually independent

- Markov chain (first-order):
  The value of X_t depends only on previous time instant

  Example (walk on G):
  $X_{t-1} = i \implies X_t$ is a neighbour of i (no influence of X_{t-2})

Markov chain and random walk 2

- Markov chain is
  - finite if $\mathcal{S}$ is finite
  - discrete / continuous state if $\mathcal{S}$ is discrete / continuous
  - discrete / continuous time if t takes discrete / continuous values
- Examples:
  - Random walk on graph: $\mathcal{S} = V$ finite, discrete time
  - Brownian motion of molecule: $\mathcal{S} = \mathbb{R}^3$ continuous state and time
- Transition probability in discrete time Markov chain:
  For $i, j \in \mathcal{S}$,

$$M_t(i,j) = \mathbb{P}(X_t = j | X_{t-1} = i) \qquad \text{can depend on } t$$

Markov chain and random walk 3

- Time-homogeneous Markov chain:
  - Transition probabilities do not depend on t
  - Characterised by single transition kernel $M(\cdot, \cdot)$
  - For finite Markov chain, M represented by a matrix
    $$M_{ij} = \mathbb{P}(X_t = j | X_{t-1} = i)$$

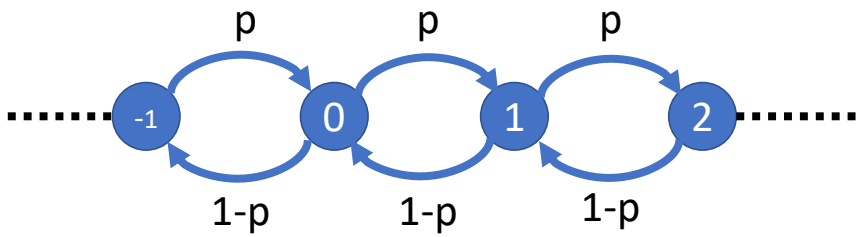
- **Exercise:** M is a row stochastic matrix, i.e., $\sum_j M_{ij} = 1$

- k-step transition probabilities:
  Let $M^{(k)}$ be a matrix with $M^{(k)}_{ij} = \mathbb{P}(X_{t+k} = j | X_t = i)$

  Exercise: Show that $M^{(k)} = M^k$
  Hint: $M^{(2)}_{ij} = \sum_\ell M_{i\ell} M_{\ell j}$

Random walk on $\mathbb{R}$

- A gambler goes to a casino, and bets 1 euro in each round
- The gambler gets 2 euros on winning, else loses the 1 euro. Win probability p
- X_t = net gain of gambler after t rounds



- Many interesting probability problems based on this
- **Example:** Gambler starts with N euros, and keeps playing. With probability 1, the gambler will get broke eventually (skipped)

Random walk on graph — formally 1

- $\mathcal{S} = V$ (set of vertices)
- Transition probability: For any t,

$$\mathbb{P}(v_t = j | v_{t-1} = i) = \begin{cases} \frac{1}{d_i} & \text{if } (i,j) \in E \\ 0 & \text{if } (i,j) \notin E \end{cases}$$

- Transition probability matrix $M = D^{-1}A = I - L_{rw}$

$$M_{ij} = \mathbb{P}(v_t = j | v_{t-1} = i)$$

- $M = D^{-1}A$ also for weighted / directed graphs

Random walk on graph — formally 2

- Let v_0 be sampled from probability mass function $p^{(0)}$
- Let $p^{(t)}$ = p.m.f for v_t. View $p^{(t)}$ as a n-dim row vector

$$p_i^{(t)} = \mathbb{P}(v_t = i)$$

- **Exercise:**
  - $p^{(1)} = p^{(0)} M, \qquad p^{(t)} = p^{(t-1)} M$
  - $p^{(t+k)} = p^{(t)} M^k$ for any $k = 0, 1, 2, \ldots$
  - Let $M^{(k)} \in \mathbb{R}^{n \times n}$ with $M_{ij}^{(k)} = \mathbb{P}(v_{t+k} = j | v_t = i)$

    Recall $M^{(k)} = M^k$
- What happens to the walk as $t \to \infty$?

Stationary distribution 1

- π is stationary distribution of the random walk if:

$$p^{(0)} = \pi \qquad \Longrightarrow \qquad p^{(t)} = \pi \text{ for every } t$$

- Spectral connection:

  — π satisfies $\pi = \pi M$

  — π is a left eigenvector of M corresponding to eigenvalue 1

- Does there exists such a π?

  — Set $\pi_i = \dfrac{d_i}{2m}$

  — Verify that $\pi = \pi M$ and π is a p.m.f.

- How many stationary distributions can a graph have?

Stationary distribution 2

- What are the eigenvalues of M?

$$M = I - L_{rw} \quad \Longrightarrow \quad \lambda(M) = 1 - \lambda(L_{rw}) = 1 - \lambda(L_{sym})$$

- **Exercise:**
  — All eigenvalues of M are real if G is undirected
  — All eigenvalues of M lie between $[-1, 1]$

- G is connected $\Longrightarrow$ M has exactly one eigenvalue equal to 1

- The eigenvalue 1 has:
  — $\pi = \left(\frac{d_1}{2m}, \ldots, \frac{d_n}{2m}\right)$ as left eigenvector, $\pi = \pi M$
  — $\mathbf{1}_n$ as right eigenvector, $M\mathbf{1}_n = \mathbf{1}_n$

Long term behaviour 1

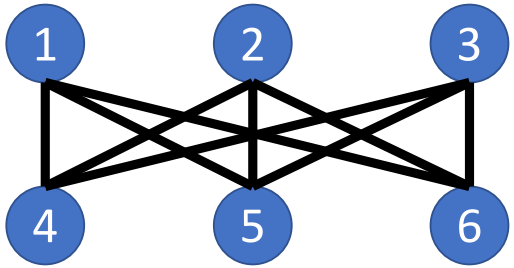
- If $p^{(0)} = \pi$ (stationary distribution), then $p^{(t)} = \pi$ for all t
- What happens if $p^{(0)}$ is arbitrary?
- Assume G is connected:
  — G is not bipartite $\implies \lim_{t\to\infty} p^{(t)} = \pi$
  — G is bipartite
  $\implies p^{(t)}$ may oscillate between two p.m.f. for odd and even t
- If G is not connected:
  — $\lim_{t\to\infty} p^{(t)}$ depends on which connected component we start

Long term behaviour 2

Bipartite case — Example

- Consider a complete bipartite graph

  Assume that walk starts on node-1, $p^{(0)} = (1, 0, \ldots, 0)$



- $p^{(1)}$ = uniform on 4, 5, 6
- $p^{(2)}$ = uniform on 1, 2, 3
- This oscillation goes on between every odd and even t

Long term behaviour 3

Random walk on connected non-bipartite graph

Let G be a connected non-bipartite graph.
Consider a random walk on G with initial distribution $p^{(0)} = p$.

$$\lim_{t\to\infty} p^{(t)} = \lim_{t\to\infty} pM^t = \pi, \qquad \text{where } \pi_i = \frac{d_i}{2m}$$

Proof:

- Need to analyse M^t
  — Cannot use eigen decomposition as $M = D^{-1}A$ is asymmetric
- Write $M = D^{-1/2}ND^{1/2}$, where $N = D^{-1/2}AD^{-1/2} = I - L_{sym}$
- Let $1 \geq \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq -1$ be eigenvalues of N
  (Why the upper and lower bounds?)

Long term behaviour 4

- Facts about N **(exercise)**
  - $\lambda_1 = 1$ with eigenvector $v_1 = \frac{1}{\sqrt{2m}} D^{1/2} \mathbf{1}_n$
  - $\lambda_2 < 1$ if G is connected
  - $\lambda_n > -1$ if G is non-bipartite (skip this proof)
  - $M^t = D^{-1/2} N^t D^{1/2}$, and $N^t = \sum_i \lambda_i^t v_i v_i^T$

- We now analyse pM^t

$$pM^t = \sum_{i=1}^{n} \lambda_i^t p D^{-1/2} v_i v_i^T D^{1/2}$$

$$= \underbrace{\frac{1}{2m} p \mathbf{1}_n \mathbf{1}_n^T D}_{=\pi} + \sum_{i=2}^{n} \underbrace{\lambda_i^t}_{\to 0} p D^{-1/2} v_i v_i^T D^{1/2}$$

Hitting and commute times 1

- Shortest path distance
  — Distance between two nodes if we take shortest route

- How far are two nodes if we follow a random walk?

Hitting time / Access time

- Assume $v_0 = i$

- Let $T_{ij} = \min\{t \geq 0 : v_t = j\}$ (smallest #steps to reach j from i)

- Hitting time, $H_{ij} = \mathbb{E}[T_{ij}|v_0 = i]$ (expected time to reach j from i)

Hitting and commute times 2

Commute time / Commute distance

- In general, $H_{ij} \neq H_{ji}$
- Commute time, $C_{ij} = H_{ij} + H_{ji}$

(expected time to go from i to j and back)

Computing hitting and commute times

Let $L_{sym}^{\dagger} \in \mathbb{R}^{n \times n}$ be the pseudo-inverse of L_{sym}

$$H_{ij} = 2m \left(\frac{\left(L_{sym}^{\dagger}\right)_{jj}}{d_j} - \frac{\left(L_{sym}^{\dagger}\right)_{ji}}{\sqrt{d_i d_j}} \right)$$

Hitting and commute times 3

Proof: Will skip the full proof, but discuss some key points. Complete proof in Luxburg, Radl & Hein, *JMLR*, 2014.

Steps discussed here:
— What is pseudo-inverse?
— Basic idea for computing H_{ij}

Pseudo-inverse:

- Let $B \in \mathbb{R}^{n \times n}$ symmetric with spectral decomposition $\sum_{k=1}^{n} \mu_k u_k u_k^T$
- $B^{-1} = \sum_{k=1}^{n} \frac{1}{\mu_k} u_k u_k^T$ exists if all eigenvalues are non-zero
- $B^{\dagger} = \sum_{k:\mu_k \neq 0} \frac{1}{\mu_k} u_k u_k^T$ is pseudo-inverse

Hitting and commute times 4

Computing H_{ij}:

- $H_{ii} = 0$
- For $i \neq j$,

$$H_{ij} = 1 + \frac{1}{d_i} \sum_{\ell \sim i} H_{\ell j} = 1 + \left(D^{-1} A H\right)_{ij}$$

  — $T_{ij} = 1 + T_{\ell j}$ for any ℓ neighbour of i

  — After one step, we reach any neighbour of i with probability $\frac{1}{d_i}$

- Can be re-written as $(L_{rw} H)_{ij} = 1$ for $i \neq j$
- Solving the set of equations for $i = j$ and $i \neq j$ gives the result

Modified random walks

Lazy random walk

- If graph has no self loop,
  — walk always moves away from current location

- Lazy walk: Move to a neighbour with probability $\alpha \in (0, 1)$, else stay at current position

- Assume there is no self loop

$$M_{ij} = \mathbb{P}(v_{t+1} = j | v_t = i) = \begin{cases} 1 - \alpha & \text{if } j = i \\ \alpha / d_i & \text{if } j \neq i, \text{ but } (i, j) \in E \end{cases}$$

- More generally, $M = (1 - \alpha) I + \alpha W$

  — $W = D^{-1} A =$ transition matrix for the standard random walk

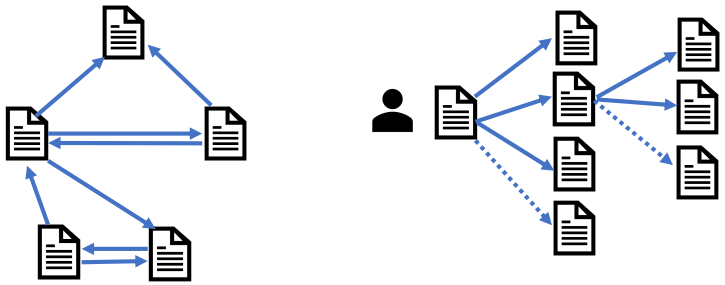
Random walk with restart

- Let s be seed node from which walk starts
- Restart: With probability $1-\alpha$, we start afresh from s
- $M_{ij} = \mathbb{P}(v_{t+1} = j | v_t = i) = \begin{cases} 1-\alpha & \text{if } j = s \\ \alpha/d_i & \text{if } (i,j) \in E \end{cases}$
- $M = (1-\alpha) e_s e_s^T + \alpha W$

  — $e_s = s^{th}$ standard basis vector
- Alternatively, one can write in terms of the distribution

  $p^{(t+1)} = (1-\alpha) e_s^T + \alpha p^{(t)} W$

Random surfer (walk behind PageRank)



- With probability $1-\alpha$, move to a random node

  — Helps to reach different components in disconnected graphs

- $M_{ij} = \mathbb{P}(v_{t+1} = j | v_t = i) = \dfrac{1-\alpha}{n} + \alpha \dfrac{A_{ij}}{d_i}$

  $$\implies \qquad M = \frac{1-\alpha}{n} \mathbf{1}_n \mathbf{1}_n^T + \alpha W$$

- In terms of the distribution

  $$p^{(t+1)} = \frac{1-\alpha}{n} \mathbf{1}_n^T + \alpha p^{(t)} W$$

Personalised PageRank (walk)

- Restart: Fixes a specific seed node for restarting

  $\implies \quad p^{(t+1)} = (1-\alpha)e_s^T + \alpha p^{(t)} W$

- Random surfer: Restarts from a uniformly random node

  $\implies \quad p^{(t+1)} = \frac{1-\alpha}{n}\mathbf{1}_n^T + \alpha p^{(t)} W$

- Personalised PageRank: A generalisation of both

- Let q be a given distribution over the nodes

  — Restart by randomly choosing a node according to q

  $\implies \quad p^{(t+1)} = (1-\alpha)q + \alpha p^{(t)} W$

Random walk on weighted graph

- $A =$ weighted adjacency of graph, and $d_i = \sum_j A_{ij}$
- $M_{ij} = \mathbb{P}(v_{t+1} = j | v_t = i) = \dfrac{A_{ij}}{d_i} \quad \Longrightarrow \quad M = D^{-1}A$
- **Exercise:**
  Can we write each of the above walks in terms of walks on some weighted graph?

PageRank and Eigen-centrality

PageRank for undirected graph

- Under random surfer model with $\alpha \in (0, 1)$

$$p^{(t+1)} = \frac{1-\alpha}{n}\mathbf{1}_n^T + \alpha p^{(t)} W$$

— $p_i^{(t)}$ = probability that walk is at node-i at step-t

- What happens as $t \to \infty$?
— Convergence to a stationary distribution π_{pr}

$$\pi_{pr} = \frac{1-\alpha}{n}\mathbf{1}_n^T + \alpha \pi_{pr} W$$

- PageRank vector is the unique stationary distribution in this case

$$\pi_{pr} = \frac{1-\alpha}{n}\mathbf{1}_n^T \big(I - \alpha W\big)^{-1}$$

- Why don't we need to assume connected and non-bipartite?

More centrality measures

- High $(\pi_{pr})_i$ = node-i more likely to be frequently visited
  — Measures importance of node

- Instead of random surfer, consider standard random walk
  — $\pi = \left(\frac{d_1}{2m}, \ldots, \frac{d_n}{2m}\right)$, steady-state distribution of walk
  — Does π correspond to a centrality measure?

Eigen centrality:

- π = left eigenvector of $D^{-1}A$ corresponding to largest eigenvalue
- Instead, simply consider eigenvectors of A
  — $Av = \lambda v$ where λ is largest eigenvalue of A
- v_i = eigen centrality of node-i

Semi-supervised learning

Information propagation in graph

Random walk to information propagation

- Standard random walk: $p^{(t+1)} = p^{(t)} W$

$$p^{(0)} = e_i \qquad \Longrightarrow \qquad p_j^{(1)} = \frac{1}{d_i} \text{ for } j \sim i$$

— probability mass at i gets distributed to its neighbours

- Personalised PageRank (PPR): $p^{(t+1)} = (1-\alpha)q + \alpha p^{(t)} W$

$$p^{(0)} = q = e_i \qquad \Longrightarrow \qquad p_i^{(1)} = 1-\alpha, \;\; p_j^{(1)} = \frac{\alpha}{d_i} \text{ for } j \sim i$$

— node-i retains α mass, and rest is distributed among neighbours

- Same occurs if $p^{(t)}$ is replaced by an arbitrary vector $f^{(t)} \in \mathbb{R}^n$

— In each step, node-i shares its information $f_i^{(t)}$ with neighbours

SSL: Problem and algorithm

Semi-supervised learning problem

Reference:

- Zhou et al. *Learning with Local and Global Consistency*, NIPS-2004.
  `https://papers.nips.cc/paper/2506-learning-with-local-and-global-consistency.pdf`

Given:

- Data points, $x_1, \ldots, x_\ell, x_{\ell+1}, \ldots, x_n \in \mathbb{R}^d$
- Labels, $y_1, \ldots, y_\ell \in \{-1, 1\}$
- Similarity matrix $A \in \mathbb{R}^{n \times n}$
  — A_{ij} = similarity score between x_i and x_j

Problem: Infer labels of $x_{\ell+1}, \ldots, x_n$

Label propagation 1

- Consider graph with (weighted) adjacency matrix A
- Define a row vector $q \in \mathbb{R}^n$

$$q_i = \begin{cases} y_i & \text{for } i \leq \ell \\ \\ 0 & \text{for } i > \ell \end{cases}$$

- Perform PPR starting with $f^{(0)} = q$ and some $\alpha \in (0, 1)$

$$f^{(t+1)} = (1 - \alpha)q + \alpha f^{(t)} D^{-1} A$$

- Do we need to run this for $t \to \infty$?
  — No, we can compute steady-state vector

$$\pi_{ppr} = (1 - \alpha)q \left(I - \alpha W\right)^{-1}$$

Label propagation 2

- Predict labels $y_{\ell+1}, \ldots, y_n$ as follows

$$y_i = \begin{cases} +1 & \text{if } (\pi_{ppr})_i > 0 \\ \\ -1 & \text{if } (\pi_{ppr})_i < 0 \end{cases}$$

  Choose arbitrarily if $(\pi_{ppr})_i = 0$

- **Remarks:**
  - Often q and π_{ppr} defined as column vectors. Then

$$\pi_{ppr} = (1-\alpha)\left(I - \alpha A D^{-1}\right)^{-1} q$$

  - Label propagation is not formally a random walk
    — We can replace $W = D^{-1}A$ by other matrices, for instance,

$$\pi = (1-\alpha)\left(I - \alpha D^{-1/2} A D^{-1/2}\right)^{-1} q$$

  - We can drop $(1-\alpha)$-factor as it does not affect final result

Label propagation 3

Label propagation algorithm

1. Define $q \in \{-1, 0, +1\}^n$ as

$$q_i = \begin{cases} y_i & \text{if label of node-}i\text{ is known} \\ \\ 0 & \text{otherwise} \end{cases}$$

2. Compute either of following:

$$\pi = \begin{cases} (1-\alpha)\left(I - \alpha A D^{-1}\right)^{-1} q & \text{for PPR} \\ \\ (1-\alpha)\left(I - \alpha D^{-1/2} A D^{-1/2}\right)^{-1} q & \text{symmetric case} \end{cases}$$

3. Predict labels as

$$y_i = \begin{cases} +1 & \text{if } \pi_i \geq 0 \\ \\ -1 & \text{if } \pi_i < 0 \end{cases}$$

Label propagation 4

Label propagation for k-class

1. Define $Q \in \{0,1\}^{n \times k}$ as

$$Q_{ij} = 1 \text{ if } y_i \text{ is known and } y_i = j$$

2. Compute $\Pi \in \mathbb{R}^{n \times k}$

$$\Pi = (1-\alpha)\left(I - \alpha D^{-1/2} A D^{-1/2}\right)^{-1} Q$$

3. Predict unknown labels as

$$y_i = \underset{j \in \{1,\ldots,k\}}{\arg\max} \ \Pi_{ij}$$

A regularisation framework

- Consider the binary setting, and define $q \in \{-1, 0, +1\}^n$ as before
- Minimise the cost

$$J(f) = \underbrace{\sum_{i=1}^{n} (f_i - q_i)^2}_{\text{fitting constraint}} + \underbrace{\lambda \sum_{i,j=1}^{n} A_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2}_{\text{smoothing constraint}}$$

- For optimal $f = f^*$ (**exercise**)

$$\left. \frac{\partial J}{\partial f_i} \right|_{f=f^*} = 0 \text{ for all } i \implies \left((1+\lambda) I - \lambda D^{-1/2} A D^{-1/2} \right) f = q$$

- $f^* = \pi$ for $\alpha = \dfrac{\lambda}{1+\lambda}$

Network dynamics

References

- Lectures 10-13 of Leskovec's course
- Epidemics:
  - Martcheva, *Introduction to Epidemic Modeling*, 2015
  - Ganesh, Massoulie and Towsley, *The effect of network topology on the spread of epidemics*, INFOCOM 2005
- Network cascade and influence maximisation
  - Kempe, Kleinberg and Tardos, *Maximizing the Spread of Influence through a Social Network*, Theory of Computing, 2015
  - Castillo, Chen and Lakshmanan *Information and influence spread in social networks*, KDD 2012 Tutorial
    `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/07/kdd12-tutorial-inf-part-iii_notes.pdf`

Epidemics in networks

SIR model 1

- Models how a disease spreads and it is cured (proposed in 1927)
- Population of n people
- Three types of states for each person (varies over time)

$$\textbf{Susceptible} \longrightarrow \textbf{Infected} \longrightarrow \textbf{Recovered}$$

  - Susceptible individuals get disease from infected people
  - Infected people are gradually cured
  - Recovered individuals cannot be further infected (model for smallpox etc.)

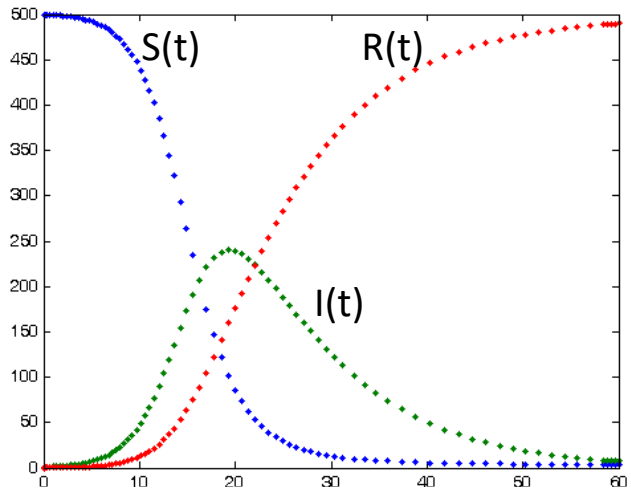
SIR model 2

- Originally not associated as a network
  Alternative view: Everyone interacts (complete graph)

- Mathematical model:
  $S(t), I(t), R(t)$ — number of people in each state at time t

$$S(t) + I(t) + R(t) = n$$

$$\frac{dS}{dt} = -\beta SI$$
$$\frac{dI}{dt} = \beta SI - \alpha I$$
$$\frac{dR}{dt} = \alpha I$$



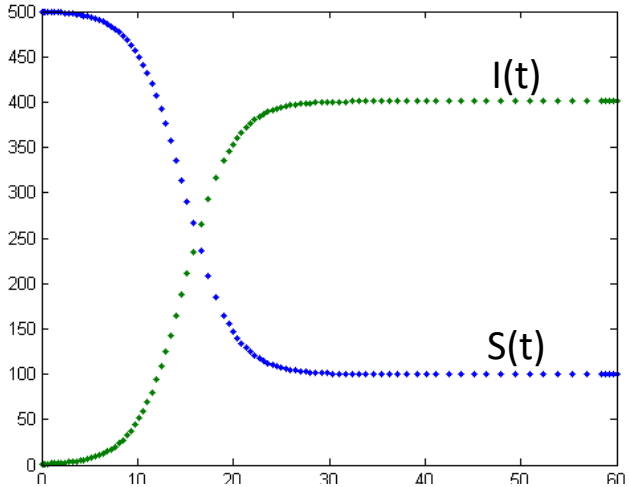
[Image: Wikipedia]

- Note: SI = total #interactions between infected and susceptibles

SIS model 1

- SIR cannot model diseases like flu
- SIS: Each individual switches between 2 states
  - Susceptible individuals get disease from infected people
  - Infected people are gradually cured, but are susceptible
- Mathematical model:
  $S(t), I(t)$ — number of people in each state at time t

$$\frac{dI}{dt} = \beta SI - \alpha I$$
$$\frac{dS}{dt} = -\beta SI + \alpha I$$
$$n = S(t) + I(t)$$



[Image: Wikipedia]

SIS model 2

- Simplifying the equations

$$\begin{aligned}\frac{dI}{dt} &= \beta I(n-I) - \alpha I \\ &= rI\left(1 - \frac{\beta}{r}I\right) \qquad\qquad (r = \beta n - \alpha)\end{aligned}$$

- Key quantity: Basic reproduction number, $R_0 = \dfrac{\beta n}{\alpha}$
  — how fast the virus reproduces

- Case 1: $R_0 < 1 \Longrightarrow r < 0$

$$\frac{dI}{dt} \leq rI \qquad \Longrightarrow \qquad I(t) \leq I(0)e^{rt} \qquad \Longrightarrow \qquad \lim_{t\to\infty} I(t) = 0$$

  Disease is eventually cured completely

SIS model 3

- Case 2: $R_0 > 1 \implies r > 0$

$$I(t) = \frac{\frac{r}{\beta}}{1 + \left(\frac{r}{\beta I(0)} - 1\right) e^{-rt}} \qquad \implies \qquad \lim_{t \to \infty} I(t) = \frac{r}{\beta} = n - \frac{\alpha}{\beta}$$

  Disease becomes endemic (always exists)

- Case 3: What happens for $R_0 = 1$?

SIS on arbitrary graph 1

- Let $G = (V, E)$ be an undirected graph on population
- Infection can spread only through edges
- Model is slightly different from above
  - $X_i(t) =$ indicator that node-i is infected at time t (random)
  - $\{X_i(t) : i \in V, t \geq 0\}$ is a continuous time Markov chain
    — infection spread randomly with some transition rate
    — more complicated to describe than discrete time case

SIS on arbitrary graph 2

- Informal intuition (assuming discrete time)

$$\mathbb{P}\big(X_i(t+1) = 1 \big| X_i(t) = 0, X(t)\big) \propto \beta \sum_j A_{ij} X_j(t)$$

$$\mathbb{P}\big(X_i(t+1) = 0 \big| X_i(t) = 1, X(t)\big) \propto \alpha$$

Threshold for disease becoming endemic (Ganesh et al., 2005)

Let λ_1 be largest eigenvalue of the graph adjacency matrix, and $I(t) = \sum_i X_i(t)$.

$$\frac{\beta}{\alpha} < \frac{1}{\lambda_1} \qquad \Longrightarrow \qquad \mathbb{P}\big(I(t) = 0\big) \to 1 \text{ as } t \to \infty$$

Proof skipped.

Network cascades

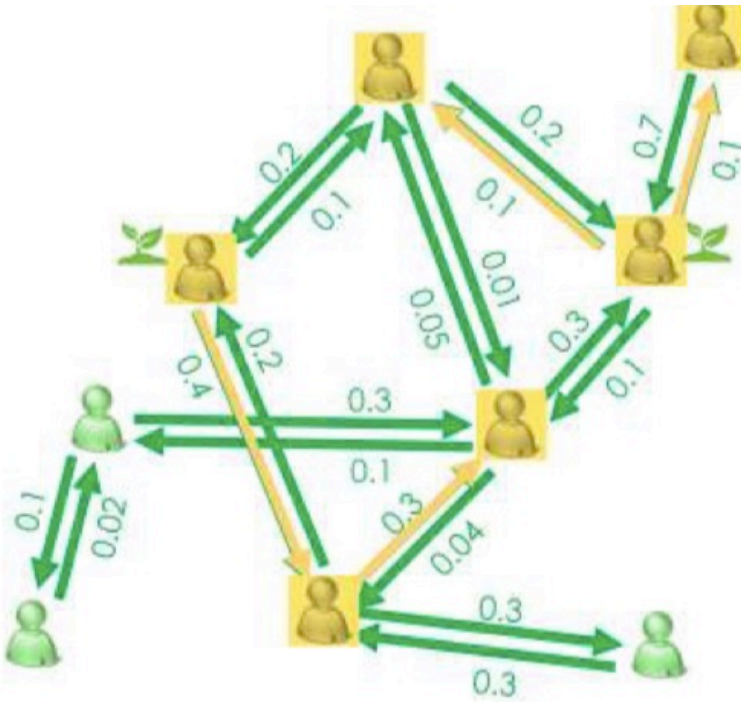
Recall: Information flow in label propagation

- Few nodes had label information $\{\pm 1\}$
- Labels shared with neighbours
- Neighbours propagate the partial label information they receive
  — A node shares his information even if it has a small value
- Does this model behaviour of:
  - forwarding tweets?
  - spread of news in media?

  — What happens in this case?

Network cascade / Information cascade

- How does information spread in internet / media?
- How does popularity (of product) spread in social network?
- How does epidemic spread?



- Information originates from few source nodes / seeds
- Seeds activate some of their neighbours (using some rule)
- In every iteration, activated nodes try to activate their neighbours

[Image: Castillo et al. 2012]

Linear threshold (LT) model

- Given weighted directed graph $G = (V, E)$
- For every node v, $\sum_u A_{uv} \leq 1$
- Every node has a threshold, $\theta_v \sim \text{Uniform}[0, 1]$
- Let $X_v(t) \in \{0, 1\}$ denote v is active at time t (discrete)
  — once activated, a node stays active
- v is seed $\implies X_v(0) = 1$
- v becomes at time t active if $\sum_u A_{uv} X_u(t-1) \geq \theta_v$
- Process stops when number of active nodes achieve steady state
  — If G is strongly connected, do all nodes get activated?

Independent cascade (IC) model

- Given unweighted directed graph $G = (V, E)$
- Each edge (u, v) has a probability p_{uv} of spreading information
- Seeds activated at time $t = 0$
- Let u is activated at time $t - 1$
- For every v such that $(u, v) \in E$

  at time t, u activates v with probability p_{uv}
- All random activations are independent
- Process stops at T if no further nodes are activated at T

LT vs. IC models

- Duration of influence
  - LT: Active nodes can always influence neighbours
  - IC: Nodes activated at $t-1$ can only influence at time t
- Source of randomness
  - LT: Every node has a personal random threshold for activation
  - IC: Activation controlled by the probabilities on edges
- Graph type
  - LT: Weighted graph
  - IC: Unweighted graph with transmission probability for each edge
    Alternative view of IC:
    — G_{live} = random directed graph with edge probability p_{uv}
    — $(u, v) \in E_{live}$ and u activated at $t-1 \implies v$ activated at t

Influence maximisation

Influence maximisation

- Basic problem of viral marketting
  - Manufacturer gives free samples to few individuals
  - Product recommendation spreads through *word of mouth*
  - Everyone who hears about it, buys the product

Who should be given free samples?

Influence maximisation

- Basic problem of viral marketting
  - Manufacturer gives free samples to few individuals
  - Product recommendation spreads through *word of mouth*
  - Everyone who hears about it, buys the product

  Who should be given free samples?

- LT and IC model how influence spreads in network

- **Influence spread** $\sigma(S)$: Starting from seed set S,

  $\sigma(S)$ = expected #active nodes when diffusion process ends

- **Problem:** For a given budget k

$$\underset{S \subset V : |S| \leq k}{\text{maximise}} \ \sigma(S) \qquad \text{(NP-hard problem)}$$

A greedy algorithm

Greedy algorithm

1. Set $\widehat{S} = \emptyset$
2. For $i = 1, \ldots, k$
   - i Let $v_i = \underset{v \in V \setminus S_{i-1}}{\arg\max} \ \sigma\left(\widehat{S} \cup \{v\}\right)$
   - ii $\widehat{S} = \widehat{S} \cup \{v_i\}$

Approximation guarantee for greedy algorithm

$$\sigma(\widehat{S}) \geq \left(1 - \frac{1}{e}\right) \max_{|S| \leq k} \sigma(S)$$

Proof: Step 1 - Show that σ is a monotone submodular function

Step 2 - Analyse greedy for maximising any monotone submodular f

Monotone submodular functions 1

Monotone function

- Let V be a set, and $f : 2^V \to \mathbb{R}$
- f is monotone if

$$S \subset T \qquad \Longrightarrow \qquad f(S) \leq f(T)$$

Submodular function

- $f : 2^V \to \mathbb{R}$ is submodular if for any $S \subset T$ and $v \in V \backslash T$

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$$

- Equivalent definition: f is submodular if for any $A, B \subset V$

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$

(Exercise: Prove equivalence)

Monotone submodular functions 2

Example:

- Assume IC model
- Influence spread $\sigma : 2^V \to \mathbb{R}$ is a monotone submodular function

Proof:

- Let G_{live} = random sample of live graph in IC model
- Let $r_{G_{live}}(S)$ = #nodes activated by S in G_{live}

  (equivalent definition of $r_{G_{live}}(S)$?)
- $\sigma(S)$ = expected #nodes activated by S

$$= \sum_{G_{live}} \mathbb{P}(G_{live}) r_{G_{live}}(S)$$

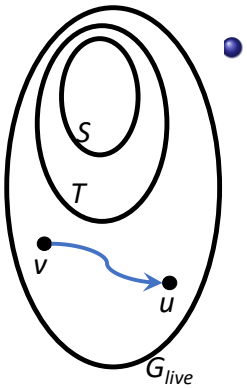
Monotone submodular functions 3

- **Exercise:**
  If $f_1, \ldots, f_m$ are monotone submodular and $a_1, \ldots, a_m \in [0, \infty)$,

$$\text{then } f = \sum_j a_j f_j \text{ is monotone submodular}$$

- We will show $r_{G_{live}}$ is monotone submodular
  $\implies \sigma$ is also monotone submodular

- $r_{G_{live}}(S) = \#\text{nodes in } G_{live} \text{ reachable from } S$
  - Obviously monotone



- $r_{G_{live}}$ is submodular:
  — Let $S \subset T \subset V$ and $v \in V \backslash T$
  — To show: $r_{G_{live}}(S \cup \{v\}) - r_{G_{live}}(S) \geq r_{G_{live}}(T \cup \{v\}) - r_{G_{live}}(T)$
  — Let u is reachable from v but not from T (u contributes to rhs)
  — Then u is also not reachable from $S \implies u$ contributes to lhs

Analysis of greedy method 1

- Let $f : 2^V \to \mathbb{R}$ be monotone, submodular and $f(\emptyset) \geq 0$
- Suppose greedy algorithm is used to maximize $\underset{|S| \leq k}{} f(S)$
- We show $f(\widehat{S}) \geq \left(1 - \frac{1}{e}\right) f(S^*)$ where $S^* = \underset{|S| \leq k}{\arg\max} f(S)$

Greedy algorithm (rephrased)

1. Set $S_0 = \emptyset$
2. For $i = 1, \ldots, k$
   - i $v_i = \underset{v \in V \setminus S_{i-1}}{\arg\max} f\left(S_{i-1} \cup \{v\}\right) = \underset{v \in V \setminus S_{i-1}}{\arg\max} \left(f(S_{i-1} \cup \{v\}) - f(S_{i-1})\right)$
   - ii $S_i = S_{i-1} \cup \{v_i\}$
3. Return $\widehat{S} = S_k$

Analysis of greedy method 2

- $S_i = \{v_1, \ldots, v_i\}$, where $v_i = \underset{v \in V \setminus S_{i-1}}{\arg\max} \big(f(S_{i-1} \cup \{v\}) - f(S_{i-1})\big)$
- Let optimal set $S^* = \{v_1^*, \ldots, v_k^*\}$
- By monotonicity, $f(S^*) \leq f(S_i \cup S^*)$

$$f(S_i \cup S^*) = f(S_i) + \sum_{j=1}^{k} \big(f(S_i \cup \{v_1^*, \ldots, v_j^*\}) - f(S_i \cup \underbrace{\{v_1^*, \ldots, v_{j-1}^*\}}_{\emptyset \text{ for } j=1})\big)$$

$$\leq f(S_i) + \sum_{j=1}^{k} \big(f(S_i \cup \{v_j^*\}) - f(S_i)\big) \qquad \text{(by submodularity)}$$

$$\leq f(S_i) + k\big(f(\underbrace{S_i \cup \{v_{i+1}\}}_{S_{i+1}}) - f(S_i)\big) \quad (v_{i+1} \text{ gives max increment})$$

Analysis of greedy method 3

- From above, we have

$$f(S_{i+1}) - f(S_i) \geq \frac{1}{k}\big(f(S^*) - f(S_i)\big)$$

- Define $\delta_i = f(S^*) - f(S_i)$

$$\implies f(S_{i+1}) - f(S_i) = \delta_i - \delta_{i+1} \geq \frac{\delta_i}{k}$$

- Observe $\delta_0 = f(S^*) - f(\emptyset) \leq f(S^*)$ and $\delta_{i+1} \leq \left(1 - \frac{1}{k}\right)\delta_i$

$$\implies \delta_k \leq \left(1 - \frac{1}{k}\right)^k \delta_0 \leq \frac{1}{e} f(S^*)$$

(use $1 - x \leq e^{-x}$)

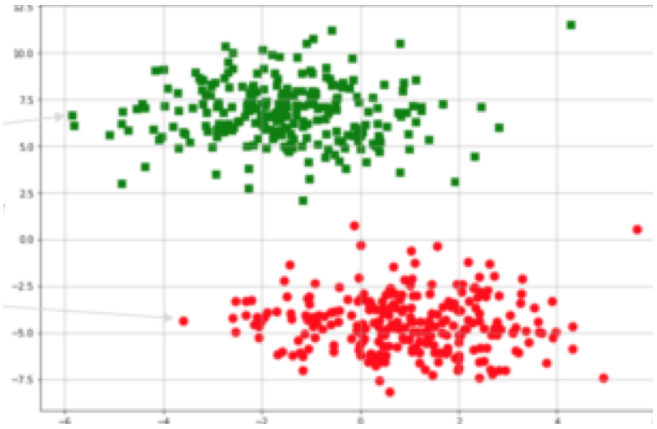
- So $f\big(\widehat{S}\big) = f(S^*) - \delta_k \geq \left(1 - \frac{1}{e}\right) f(S^*)$

Further topics (not part of exam)

Graph kernels

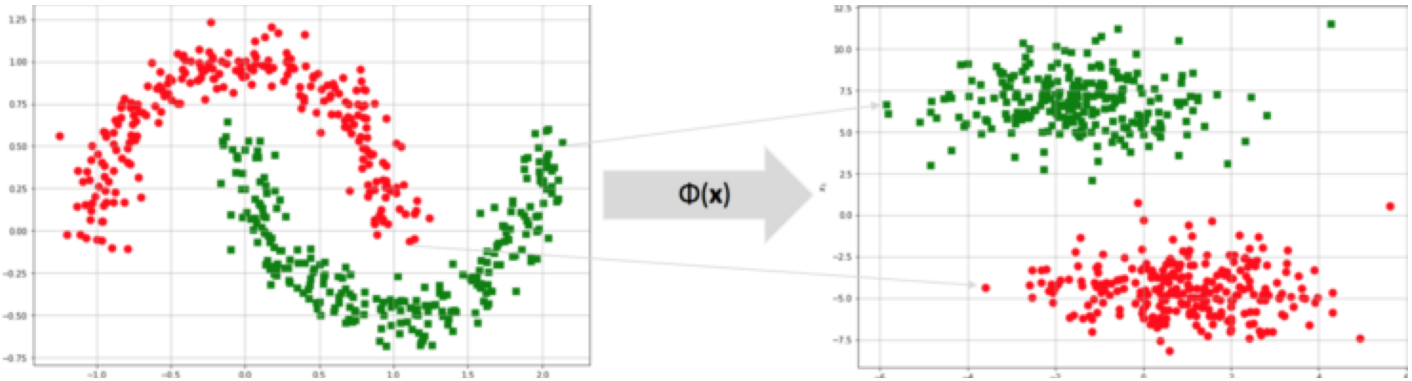
ML Recap: Kernel functions 1

Linearly separable data (easy for machine learning)



- Classification:
  — Linear hyperplane separates the two classes (SVM, LDA)
- Clustering:
  — Group into disjoint balls (k-means)

Non-linearly separable data: [Image: G. Bonaccorso]



Apply a non-linear function $\Phi(\cdot)$ to make data linearly separable

ML Recap: Kernel functions 2

- Difficult to find a suitable function Φ
- Often ML algorithms do not require $\Phi(x), \Phi(y)$

  — but only $\|\Phi(x) - \Phi(y)\|$ or $\Phi(x)^T \Phi(y)$

  — Example: kernel SVM, kernel k-means
- **Kernel function:** For an input space $\mathcal{X}$,

  $$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

  is a positive semidefinite kernel if for any $x_1, \ldots, x_n \in \mathcal{X}$,
  - $k(x_i, x_j) = k(x_j, x_i)$
  - $K \in \mathbb{R}^{n \times n}$ with $K_{ij} = k(x_i, x_j)$ is positive semidefinite
- **Result:** For every positive semidefinite kernel k, there is

  $$\Phi_k : \mathcal{X} \to \mathcal{Z} \text{ such that } k(x, y) = \Phi_k(x)^T \Phi_k(y)$$

Kernels on graphs 1

Machine learning on vertices

- Given graph $G = (V, E)$
  — Think of $V = \{v_1, \ldots, v_n\}$ as data points

- Graph embedding:
  — Conceptually similar to PCA for the vertices

- Communities / Graph partitioning:
  — Clustering of vertices

- Label propagation:
  — Predict labels of unlabeled vertices (classification)

Kernels on graphs 2

- If we have kernel function $k : V \times V \to \mathbb{R}$
  — Can do above using using kernel k-means, kernel SVM etc.

- Example:
  Diffusion kernel (Kondor & Lafferty, ICML 2002)

$$\text{Kernel matrix} \qquad K = e^{-\beta L} = \sum_{i=0}^{\infty} \frac{(-\beta)^i}{i!} L^i$$

  — L = unnormalised Laplacian, and $\beta > 0$ is a parameter

  — Connections to random walk on graph

- ML on graph without kernel
  — Is there a generic alternative to kernel based techniques?
  (distances between nodes)

Kernels and distances between graphs 1

- When can we say that graph G and G' are similar?
- Similar graph properties
  — density, degree distribution, motif counts, . . .
  — Laplacians are similar, have similar eigenvalues, . . .
- How can we quantify the similarity between graphs G and G'?
  - Graph distances (distance between two graphs)
  - Graph kernels (kernel function on space of graphs)

Kernels and distances between graphs 2

Case 1: Graphs with common vertex set

- $G = (V, E)$ and $G' = (V, E')$
  - — A_G, L_G = adjacency and Laplacian matrices of G
  - — $A_{G'}, L_{G'}$ = adjacency and Laplacian matrices of G'

- A graph distance
$$d(G, G') = \|A_G - A_{G'}\|_F$$
  - — Complicated version in (Mukherjee, Sarkar & Lin, NIPS 2017)

- Laplacian graph kernel (Kondor & Pan, NIPS 2016)
$$k(G, G') = \frac{\left|\left(\frac{1}{2}L_G + \frac{1}{2}L_{G'}\right)^{-1}\right|^{1/2}}{\left|L_G^{-1}\right|^{1/4}\left|L_{G'}^{-1}\right|^{1/4}} \qquad (|\cdot| \text{ is deteminant})$$

Kernels and distances between graphs 3

Case 2: Graphs of different sizes

- $G = (V, E)$ and $G' = (V', E')$
- Application: Compare molecular / protein structures
- Difficult to compare, and so mostly open problem
- Some kernel functions available

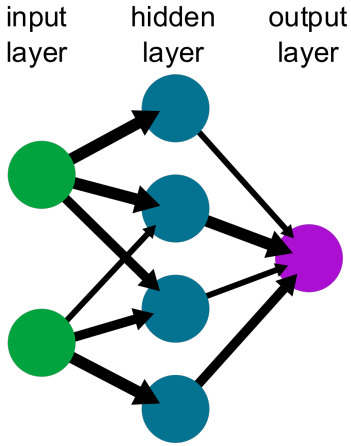
  — Random walk kernel (Vishwathan, Borgwardt, Kondor & Schraudolph, JMLR 2010)

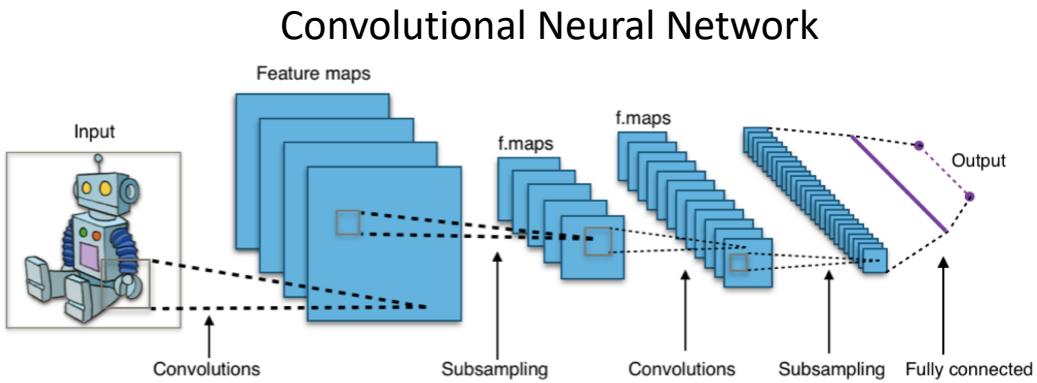
  — Weisfeller-Lehmann kernel (Shervashidze, Schweitzer, van Leeuwen, Mehlhorn & Borgwardt, JMLR 2011)

  — Multiscale Laplacian graph kernel (Kondor & Pan, NIPS 2016)

Deep learning on graphs

Convolutional neural networks 1





Neural network

- Let $h^{(t)} =$ output of t^{th} layer
- Input layer: $h^{(0)} = x$ (input vector)
- Hidden unit: $h_i^{(t+1)} = \sigma\left(\sum_j w_{ij}^{(t)} h_j^{(t)}\right)$ $\quad \sigma =$ non-linear activation
- Output of hidden layer: $h_i^{(t+1)} = \sigma\left(W^{(t)} h^{(t)}\right)$

Convolutional neural networks 2

Convolutional neural network (CNN)

- Originally used for image data
- Input layer: $h^{(0)} = x$ (input image/matrix/tensor)
  — x_i is i^{th} pixel of image
- Hidden unit: $h_i^{(t+1)} = \sigma\left(\sum_{j \in \text{Nbh}(i)} w_{ij}^{(t)} h_j^{(t)}\right)$
  — sum only over neighbourhood of i (convolution / filtering)
- Output of each conv layer: $h_i^{(t+1)} = \sigma\left(W^{(t)} h^{(t)}\right)$
  — $W^{(t)}$ has a lot of zeros
- In each stage, different activation functions are used

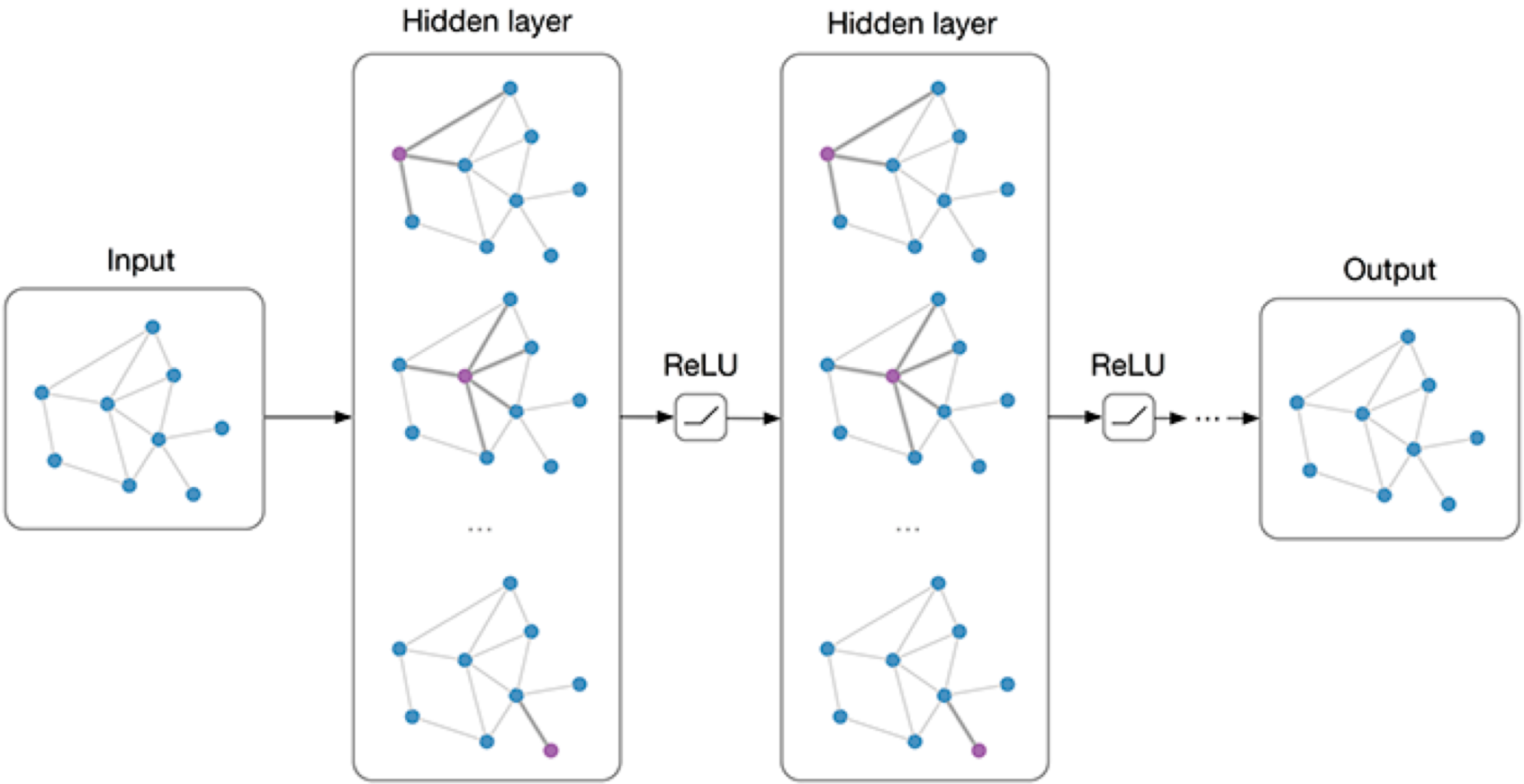
Graph convolutional network (GCN) 1

- Will discuss architechture of Kipf & Welling (ICLR 2017)
- Neighbourhood is defined by graph $G = (V, E)$
- Nodes can additional k-dim features
  — Input matrix $H^{(0} = X \in \mathbb{R}^{n \times k}$
- Can use convolution layer of the form, $H^{(t+1)} = \sigma\left(AH^{(t)}\right)$
- Typically a normalised matrix is used $M = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$
  — normalised adjacency of graph with self loops added
- Above has no parameter to tune in each layer
  — Multiply another parameter matrix,

$$H^{(t+1)} = \sigma\left(MH^{(t)}W^{(t)}\right)$$

Graph convolutional network (GCN) 2

- Output layer: Typically a matrix $Y \in \mathbb{R}^{n \times k'}$
  — k' = feature dimension for each node



Machine learning on graph data

ML with graph data

- Each data instance is a graph
  — Data $\{G_1, G_2, \ldots, G_m\}$

- Graphs on a common vertex set
  — Example: Each graph is the brain network of an individual

- Graphs on different vertices
  — Example: Each graph is a molecule / protein structure

- Learning problems:
  — Clustering, classification, hypothesis testing

- Generic methods:
  — Graph kernels / distances
  — Embedding (represent each graph as a point in Euclidean space)

Appendix

Python and NetworkX

References for Python

- You can find many tutorials / references for Python online
- If you have not used Python or Jupyter notebook before, watch this video on installing and working with Jupyter
  `https://www.youtube.com/watch?v=HW29067qVWk`
  The channel also has videos on Python for beginners
- For a crash course on Python, you can look at the tutorial by Diego Fioravanti provided during Machine Learning course
  It is in form of a Jupyter notebook (see Assignment-1)
- To find functions that you need, see documentation of important packages like `numpy` or `matplotlib`

References for NetworkX

- Python package for network analysis.
  Tutorials and list of functions in the NetworkX package can be found in their documentation.
  `https://networkx.github.io/documentation/stable/`

- For a very basic introduction to NetworkX, you can watch
  `https://www.youtube.com/watch?v=sGAT2npnNLc&t=24s`

- There are other packages like **SNAP** or **iGraph** that can be used with R, Python or C/C++.
  We will only use NetworkX for convenience.

Inequalities for sum of random variables

References

- Reviews of Linear Algebra and Probability Theory
  `http://cs229.stanford.edu/section/cs229-linalg.pdf`
  `http://cs229.stanford.edu/section/cs229-prob.pdf`
- For probability basics, Chapters 1-2 of Bruce Hajek's book
  `http://hajek.ece.illinois.edu/Papers/randomprocJuly14.pdf`
- List of some important concentration inequalities (with proofs)
  `http://www.math.ucsd.edu/~fan/wp/concen.pdf`
- Roman Vershynin's book (for concentration and also more probability)
  `https://www.math.uci.edu/~rvershyn/papers/HDP-book/HDP-book.pdf`

LLN and CLT 1

- $X_1, X_2, \ldots, X_n$ independent and identically distributed (iid) random variables
- $\mathbb{E}[X_i] = \mu$ and $\mathrm{Var}(X_i) = \mathbb{E}\left[(X_i - \mu)^2\right] = \sigma^2$
- $S_n = \sum_{i=1}^{n} X_i$

Law of Large Numbers

$$\frac{S_n}{n} \to \mu \qquad \text{as } n \to \infty$$

. . . in probability (weak LLN)

. . . almost surely (strong LLN)

Central Limit Theorem

$$\frac{S_n - n\mu}{\sqrt{n}\sigma} \to \mathcal{N}(0,1) \qquad \text{as } n \to \infty$$

. . . in distribution

LLN and CLT 2

- LLN and CLT are important, but often not enough for analysis
- Asymptotic statements
  - What happens for finite n?
  - How large is $\left|\frac{S_n}{n} - \mu\right|$ for $n = 1000$?
  - What is distribution of $\frac{S_n}{\sqrt{n}}$ for $n = 1000$?
- Independence assumption
  - What happens if the random variables are dependent?
  - What if there is a weak dependence? — Only few are dependent
- Variants of LLN and CLT that provide bounds for finite n
  - **Concentration inequalities** (deviation form of weak LLN)
  - **Berry-Esseen theorem** (deviation form of CLT)

Concentration of random variables 1

Markov's inequality

Let Y be a random variable and $h(\cdot)$ be a non-negative function.

$$\mathbb{P}\big(h(Y) \geq a\big) \leq \frac{\mathbb{E}[h(Y)]}{a} \qquad \text{for all } a > 0$$

Proof : Note that $a\mathbf{1}\big\{h(Y) \geq a\big\} \leq h(Y)$
Take expectation on both sides

- Standard Markov's inequality: If Y is non-negative r.v., then
$$P\big(Y \geq a\big) \leq \frac{\mathbb{E}[Y]}{a}$$
- Chebyshev's inequality: For any r.v. Y
$$P\big(|Y - \mathbb{E}[Y]| \geq a\big) \leq \frac{\mathrm{Var}[Y]}{a^2}$$

Concentration of random variables 2

Chernoff bound (general)

Let Y be a random variable. For any $a \in \mathbb{R}$,

$$\mathbb{P}\big(Y \geq a\big) \leq \min_{t>0} \frac{\mathbb{E}\left[e^{tY}\right]}{e^{ta}}$$

Proof : $f(x) = e^{tx}$ is a monotonic increasing function for any $t > 0$
So $\mathbb{P}\big(Y \geq a\big) = \mathbb{P}\big(e^{tY} \geq e^{ta}\big)$.
Use Markov's inequality, and note that it holds for all $t > 0$

Chernoff bound (for sum of independent r.v.)

Let $X_1, X_2, \ldots, X_n$ be independent (may not be iid). For any $a \in \mathbb{R}$,

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i \geq a\right) \leq \min_{t>0} e^{-ta} \prod_{i=1}^{n} \mathbb{E}\left[e^{tX_i}\right]$$

Concentration of random variables 3

Hoeffding's inequality (application of Chernoff's bound)

Let $X_1, \ldots, X_n$ be independent with $X_i \in [a_i, b_i]$ almost surely.

Let $S = \sum_{i=1}^{n} X_i$. For any $a > 0$,

$$\mathbb{P}\big(\,|S - \mathbb{E}[S]| \geq a\big) \leq 2\exp\left(-\frac{2a^2}{\sum_i (b_i - a_i)^2}\right)$$

Proof : See detailed proof in Wikipedia

$$\mathbb{P}\big(\,|S - \mathbb{E}[S]| \geq a\big) \leq \mathbb{P}\big(S - \mathbb{E}[S] \geq a\big) + \mathbb{P}\big(\mathbb{E}[S] - S \geq a\big)$$

. . . union bound (gives factor of 2)

Note that $S - \mathbb{E}[S] = \sum_i X_i - \mathbb{E}[X_i]$, and apply Chernoff.

Bound $\mathbb{E}\left[e^{t(X_i - \mathbb{E}[X_i])}\right]$ using Hoeffding's lemma

$$\mathbb{E}\left[e^{t(X_i - \mathbb{E}[X_i])}\right] \leq \exp\left(\tfrac{1}{8}t^2(b_i - a_i)^2\right)$$

Finally optimize over t.

Concentration of random variables 4

- **Bennett's inequality, Bernstein's inequality:** Improved bounds in terms of $\mathrm{Var}(X_i)$
- Special bounds hold when $X_i \sim \text{Bernoulli}(p_i)$
  - We can compute $\mathbb{E}[e^{tX_i}] = 1 - p_i + p_i e^t$
- What if X_i is not bounded (say Gaussian)?
  - Variants of Hoeffding or Bernstein based on sub-Gaussian norms
  - See Vershynin's book. We may not need them in this course
- **McDiarmid's inequality:** Concentration of an arbitrary function $f(X_1, \ldots, X_n)$
  - Assumption: f does not change much if only one X_i is changed
- **Azuma's inequalities:** Variants of above
  - When $X_1, \ldots, X_n$ is a martingale (particular type of dependence)
  - Useful in learning theory

Union bound and concentration

Basic union bound

Let $E_1, \ldots, E_m$ be m events

$$\mathbb{P}\left(\bigcup_{i=1}^{m} E_i\right) \leq \sum_{i=1}^{m} \mathbb{P}(E_i)$$

- Useful when we do not have independence
- Powerful in combination with Chernoff's bounds
- **Example:**
  - Let $X_1, \ldots, X_n, X_{n+1}, \ldots, X_{2n}, X_{2n+1}, \ldots X_{kn}$ be r.v.s
  - We only know $X_{(j-1)n+1}, \ldots, X_{jn}$ are independent for $j = 1, \ldots, k$
  - Decompose into k parts and use Chernoff's bound for each term

$$\mathbb{P}\left(\sum_{i=1}^{kn} X_i > a\right) \leq \mathbb{P}\left(\bigcup_{j=1}^{k} \left\{ \sum_{i=(j-1)n+1}^{jn} X_i > \frac{a}{k} \right\}\right) \leq \sum_{j=1}^{k} \mathbb{P}\left(\sum_{i=(j-1)n+1}^{jn} X_i > \frac{a}{k} \right)$$

Metric spaces, distances and norms

Distances and metrics

- V is a set of elements (finite / countably infinite / uncountable)
- **Distance:** A function that measures how far two objects are
  - No formal mathematical definition
- **Metric:** $\mathtt{d} : V \times V \to [0, \infty)$ is a metric if
  - $\mathtt{d}(u, v) = \mathtt{d}(v, u)$ for all $u, v \in V$
  - $\mathtt{d}(u, v) = 0$ if and only if $u = v$
  - $\mathtt{d}(u, v) \leq \mathtt{d}(u, w) + \mathtt{d}(v, w)$ for all $u, v, w \in V$ (triangle inequality)
- **Metric space:** $(V, \mathtt{d})$
  - Set V along with a metric $\mathtt{d}$ defined on it
  - We can define different metric spaces on the same set V

Examples of metrics

- $V = \mathbb{R}^n$ or $\{0,1\}^n$
  - Euclidean distance, $\mathtt{d}(u,v) = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}$ where $u = (u_1, \ldots, u_n)$
  - Hamming distance, $d(u,v) = \sum_{i=1}^{n} \mathbf{1}\{u_i \neq v_i\}$ where $\mathbf{1}\{\cdot\}$ is indicator
  - $\mathtt{d}(u,v) = \sum_{i=1}^{n} |u_i - v_i|$ (same as Hamming distance for $\{0,1\}^n$)
  - $\mathtt{d}(u,v) = \max_{1 \leq i \leq n} |u_i - v_i|$
- V = arbitrary set
  - Discrete metric, $\mathtt{d}(u,v) = \mathbf{1}\{u \neq v\}$
- V = set of strings
  - Edit distance, $\mathtt{d}_{edit}(u,v)$ is minimum number of substitution / insertion / deletion needed to change one string into another
    $\mathtt{d}_{edit}(\mathtt{Apple}, \mathtt{Apfel}) = 3$ ($\mathtt{Apple} \rightarrow \mathtt{Apfle} \rightarrow \mathtt{Apfl\not{e}} \rightarrow \mathtt{Apfel}$)

Graph metrics

- **Shortest path distance**

  V = vertex set of a graph

  $\mathtt{d}_{sp}(u, v)$ = length of shortest path/paths between u and v

  - Metric on an undirected connected graph
  - What happens if graph is not connected?

    — Set $\mathtt{d}_{sp}(u, v) = \infty$ if u, v are in different connected components

    — Need to change metric definition as $\mathtt{d} : V \times V \to [0, \infty]$

    (does not cause any serious problem)

- **Resistance distance**
  - Another metric for graphs (see Wikipedia)
  - Views graph as electric circuit

- **Exercise:** Why is $\mathtt{d}_{sp}$ a metric? Is it a metric for di-graphs?

Random matrices

References

- For matrix concentration, see Chapter 2 of Terence Tao's book
  `https://terrytao.files.wordpress.com/2011/02/matrix-book.pdf`
  Can be difficult without math background
- Roman Vershynin's book for concentration (scalar or matrix) and also more probability
  `https://www.math.uci.edu/~rvershyn/papers/HDP-book/HDP-book.pdf`

Norm and normed vector space

- Intuition: metric $\equiv$ distance, norm $\equiv$ length
- Let V be a vector space (over $\mathbb{R}$)
  - If $u, v \in V$, then $u + v \in V$
  - If $\lambda \in \mathbb{R}$, $v \in V$, then $\lambda v \in V$
- **Norm:** $\|\cdot\| : V \to [0, \infty)$ is a norm if
  - $\|u\| = 0$ if and only if $u = \mathbf{0}$ ($\mathbf{0} \in V$ is zero vector)
  - $\|u + v\| \leq \|u\| + \|v\|$ for all $u, v \in V$
  - $\|\lambda v\| = |\lambda| \|v\|$ for all $v \in V$, $\lambda \in \mathbb{R}$
- **Normed space:** $(V, \|\cdot\|)$
  - Vector space V along with a norm $\|\cdot\|$ defined on it
  - Note: V must be a vector space to define a norm **(Why?)**

Examples of norms: Vector norms

- $V = \mathbb{R}^n$
  - Euclidean (2-) norm, $\|v\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2}$ where $v = (v_1, \ldots, v_n)$
  - p-norm, $\|v\|_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p}$ for $1 \leq p < \infty$
  - ∞-norm, $\|v\|_\infty = \max_{1 \leq i \leq n} |v_i|$
- Every norm induces a metric
  - $\|\cdot\|$ is norm $\Rightarrow \mathtt{d}(u, v) = \|u - v\|$ is a metric
- Every metric is not generated by a norm
  - Shortest path distance on graphs (here, V is not vector space)
  - On $\mathbb{R}^n$, recall Hamming distance $\mathtt{d}(u, v) = \sum_i \mathbf{1}\{u_i - v_i \neq 0\}$
    It is induces by the zero-"norm", $\|v\|_0 = \sum_i \mathbf{1}\{v_i \neq 0\}$
    But zero-"norm" is not a norm **(Why?)**

Examples of norms: Matrix norms

- $V = \mathbb{R}^{m \times n}$
  - Frobenius norm, $\|M\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} M_{ij}^2}$
  - Nuclear norm, $\|M\|_* = \sum_{i=1}^{\min\{m,n\}} \sigma_i(M)$
    where $\sigma_1(M), \sigma_2(M), \ldots$ are singular values of M
  - Induced p-norm, $\|M\|_p = \max_{\substack{x \in \mathbb{R}^n \\ x \neq \mathbf{0}}} \frac{\|Mx\|_p}{\|x\|_p}$ for $1 \leq p \leq \infty$
- Intuition for induced p-norm
  - Think of M as a linear transformation $M : \mathbb{R}^n \to \mathbb{R}^m$
  - $\|M\|_p$ denotes the maximum rescaling of length (norm) caused by the transformation M

Examples of norms: Spectral norm

- Same as induced 2-norm / operator norm
- Assume M is symmetric $n \times n$ matrix

$$\begin{aligned} \|M\|_2 &= \max_{\substack{x \in \mathbb{R}^n \\ x \neq \mathbf{0}}} \frac{\|Mx\|_2}{\|x\|_2} \\ &= \sigma_1(M) && \sigma_1(M) \text{ is largest singular value of } M \\ &= \max_{x \in \mathbb{S}^{n-1}} \left| x^T M x \right| && \mathbb{S}^{n-1} = \{x \in \mathbb{R}^n : \|x\| = 1\} \end{aligned}$$

- **Exercise:** Show that all definitions are equivalent
- Hint: Spectral / eigenvalue decomposition of symmetric matrix

$$M = \sum_{i=1}^{n} \lambda_i x_i x_i^T \qquad (\lambda_i, x_i) \text{ is eigenvalue, eigenvector pair}$$

$\sigma_i(M) = |\lambda_i|$ and $\{x_1, \ldots, x_n\}$ are orthonormal vectors

Concentration of random matrices 1

Spectral norm of random matrix

$M \in \mathbb{R}^{n \times n}$ is a symmetric random matrix with following properties:

- $M_{ii} = 0$ for all i (zero diagonal)
- $\{M_{ij} : i < j\}$ are mutually independent
- $\mathbb{E}[M_{ij}] = 0$ and $|M_{ij}| \leq 1$ almost surely for all i, j

For any $\delta \in (0, 1)$,

$$\mathbb{P}\left(\|M\|_2 \geq C_\delta \sqrt{n}\right) \leq \delta$$

for some constant $C_\delta > 0$ that depends only on δ.

Note: If M is not random but arbitrary, then $\|M\|_2 \leq n - 1$ assuming zero diagonal and $|M_{ij}| \leq 1$ **(Why?)**

Concentration of random matrices 2

Proof:

- Recall that $\|M\|_2 = \max_{x \in \mathbb{S}^{n-1}} \left|x^T M x\right|$. Fix an $x \in \mathbb{S}^{n-1}$

$$\mathbb{P}\left(\left|x^T M x\right| \geq a\right) \;=\; \mathbb{P}\left(\left|\sum_{i<j} M_{ij} x_i x_j\right| \geq \frac{a}{2}\right) \;\leq\; 2\exp\left(-\frac{a^2}{4}\right)$$

Exercise: Prove above using Hoeffding and the fact $\|x\| = 1$

- How do we go from here to max over all $x \in \mathbb{S}^{n-1}$?
  — Union bound
  — Does not really work as $\mathbb{S}^{n-1}$ is uncountable

- ϵ-net approach: Approximate $\mathbb{S}^{n-1}$ by a finite set

Concentration of random matrices 3

ϵ-net and maximal ϵ-net

- Σ is an ϵ-net for $\mathbb{S}^{n-1}$ if
  - $\Sigma \subset \mathbb{S}^{n-1}$
  - for every $x, y \in \Sigma$, we have $\|x - y\| \geq \epsilon$
- Σ is a maximal ϵ-net if
  - we cannot add any more point to Σ and retain the property of ϵ-net
- Size of maximal ϵ-net for $\mathbb{S}^{n-1}$
  - $|\Sigma| \leq \left(1 + \frac{2}{\epsilon}\right)^n \leq \exp\left(\frac{2n}{\epsilon}\right)$ Note: $1 + x \leq e^x$
- $\|M\|_2 \leq \frac{1}{(1-2\epsilon)} \max_{x \in \Sigma} |x^T M x|$

Exercise: Let $\epsilon = \frac{1}{4}$. Use above + union bound to complete proof

Proofs: Network models

Revisiting some concentration inequalities 1

First moment method / simple Markov's inequality

X is a non-negative random variable and $t > 0$.

$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t}$$

Special cases

- If $\mathbb{E}[X] = 0$, then $\mathbb{P}(X \geq t) = 0$ for any $t > 0$

- Let $X_1, X_2, \ldots$ be a sequence of non-negative random variables.

$$\lim_{n\to\infty} \mathbb{P}(X_n \geq t) \leq \lim_{n\to\infty} \frac{\mathbb{E}[X_n]}{t} \quad \text{for any } t > 0$$

If $\mathbb{E}[X_n] \to 0$ as $n \to \infty$, then

$$\lim_{n\to\infty} \mathbb{P}(X_n \geq t) = 0, \quad \text{which we write as} \quad \mathbb{P}(X_n \geq t) = o(1)$$

Revisiting some concentration inequalities 2

Second moment method / Chebyshev's inequality

Let X be a random variable and $t > 0$.

$$\mathbb{P}\left(\left|X - \mathbb{E}[X]\right| \geq t\right) \leq \frac{\mathrm{Var}[X]}{t^2}$$

Special case

- Let $X_1, X_2, \ldots$ be a sequence of non-negative random variables. Let $\lim_{n\to\infty} \mathbb{E}[X_n] > 0$ and $E[X_n^2] \leq \big(1 + o(1)\big)\big(\mathbb{E}[X_n]\big)^2$

$$\begin{aligned}
\mathbb{P}(X_n = 0) &\leq \mathbb{P}\left(\left|X_n - \mathbb{E}[X_n]\right| \geq \mathbb{E}[X_n]\right) \\
&\leq \frac{\mathrm{Var}[X_n]}{\big(\mathbb{E}[X_n]\big)^2} \\
&= o(1) \qquad \text{(due to assumption)}
\end{aligned}$$

Isolated nodes in ER 1

Theorem: Number of isolated nodes

Let $G \sim G(n,p)$, and $X_n = \#\text{isolated nodes in } G$.

$$\mathbb{E}[X_n] = n(1-p)^{n-1}$$

$$\lim_{n\to\infty} \mathbb{E}[X_n] = 0 \quad \text{if } p > \frac{\ln n}{n}, \quad \text{and} \quad \lim_{n\to\infty} \mathbb{E}[X_n] = \infty \quad \text{if } p < \frac{\ln n}{n}.$$

Proof: $\mathbb{E}[X_n] = \sum_i \mathbb{P}(d_i = 0) = \sum_i (1-p)^{n-1}$

Let $p = \frac{c \ln n}{n}$.

$$\lim_{n\to\infty} \mathbb{E}[X_n] = \lim_{n\to\infty} n\left(1 - \frac{c\ln n}{n}\right)^n$$

$$= \lim_{n\to\infty} n e^{-c\ln n}$$

$$= \lim_{n\to\infty} n^{1-c}$$

since $\lim_{n\to\infty} e^{-a_n} = \lim_{n\to\infty} \left(1 - \frac{a_n}{n}\right)^n$

Isolated nodes in ER 2

Corollary: Presence of isolated nodes

Let $G \sim G(n, p)$

$$\mathbb{P}(G \text{ contains isolated nodes}) = \begin{cases} o(1) & \text{if } p > \dfrac{\ln n}{n} \\ 1 - o(1) & \text{if } p < \dfrac{\ln n}{n} \end{cases}$$

Here, $x = o(1)$ means $\lim_{n \to \infty} x = 0$. Equivalently,

$$\mathbb{P}(X_n \geq 1) = o(1) \quad \text{if } p > \frac{\ln n}{n}$$

$$\mathbb{P}(X_n = 0) = o(1) \quad \text{if } p < \frac{\ln n}{n}$$

Proof (first part): $\mathbb{P}(X_n \geq 1) \leq \mathbb{E}[X_n] = o(1)$ if $p > \frac{\ln n}{n}$

Isolated nodes in ER 3

Proof (second part): Use second moment method. Compute $\mathbb{E}[X_n^2]$.

$$X_n = \sum_{i=1}^{n} \mathbf{1}\{d_i = 0\}$$

$$X_n^2 = \sum_{i} \mathbf{1}\{d_i = 0\} + \sum_{i \neq j} \mathbf{1}\{d_i = 0, d_j = 0\}$$

$$\begin{aligned}\mathbb{E}[X_n^2] &= n\mathbb{P}(d_1 = 0) + n(n-1)\mathbb{P}(d_1 = 0, d_2 = 0) \\ &= n(1-p)^{n-1} + n(n-1)(1-p)^{2(n-1)-1}\end{aligned}$$

$$\frac{\mathbb{E}[X_n^2]}{\big(\mathbb{E}[X_n]\big)^2} = \frac{1}{n(1-p)^{n-1}} + \frac{1}{1-p} - \frac{1}{n(1-p)} = 1 + o(1)$$

if $p = \frac{c \ln n}{n}$ with $c < 1$. Now use second moment result.

Diameter of ER 1

Theorem: Phase transition in diameter

Let $G \sim G(n, p)$.

$$\mathbb{P}\big(\text{diameter}(G) \leq 2\big) = \begin{cases} 1 - o(1) & \text{if } p > \sqrt{\dfrac{2 \ln n}{n}} \\ o(1) & \text{if } p < \sqrt{\dfrac{2 \ln n}{n}} \end{cases}$$

Diameter of ER 2

Let X_n = number of pairs i, j such that $\mathtt{d}_{sp}(i, j) > 2$

$$\mathbb{P}\big(\text{diameter}(G) \leq 2\big) = \mathbb{P}(X_n = 0)$$

Theorem: Phase transition in diameter (restated)

Let $G \sim G(n, p)$.

$$\mathbb{P}(X_n \geq 1) = o(1) \quad \text{if } p > \sqrt{\frac{2 \ln n}{n}}$$

$$\mathbb{P}(X_n = 0) = o(1) \quad \text{if } p < \sqrt{\frac{2 \ln n}{n}}$$

Diameter of ER 3

Proof: Let us call $\{i, j\}$ bad pair

- if $\mathtt{d}_{sp}(i, j) > 2$,
- or equivalently,
  if i, j not adjacent and do not share a common neighbour,
- or equivalently,
  if $(i, j) \notin E$ and for every $v \neq i, j$, either $(i, v) \notin E$ or $(j, v) \notin E$

$$\mathbb{P}(i, j \text{ bad pair}) = (1-p)\left(1-p^2\right)^{n-2}$$

$$X_n = \sum_{i<j} \mathbf{1}\{i, j \text{ bad pair}\} = \frac{1}{2}\sum_{i \neq j} \mathbf{1}\{i, j \text{ bad pair}\}$$

$$\mathbb{E}[X_n] = \frac{n(n-1)}{2}(1-p)\left(1-p^2\right)^{n-2}$$

Put $p = c\sqrt{\frac{\ln n}{n}}$

$$\lim_{n\to\infty} \mathbb{E}[X_n] = \lim_{n\to\infty} \frac{n(n-1)}{2}\left(1 - c\sqrt{\frac{\ln n}{n}}\right)\left(1 - \frac{c^2 \ln n}{n}\right)^n$$

$$= \lim_{n\to\infty} \frac{n^2}{2}\left(1 - \frac{c^2 \ln n}{n}\right)^n$$

$$= \lim_{n\to\infty} \frac{n^2}{2} e^{-c^2 \ln n}$$

$$= \lim_{n\to\infty} \frac{n^{2-c^2}}{2}$$

For $c > \sqrt{2}$, $\lim_{n \to \infty} \mathbb{E} X_n = 0$

$c < \sqrt{2}$, $\lim_{n \to \infty} \mathbb{E} X_n = \infty$

Part 1 : $c > \sqrt{2}$ or $p > \sqrt{\frac{2 \ln n}{n}}$

$$\mathbb{P}(X_n \geq 1) \leq \mathbb{E}[X_n] = o(1)$$

Part 2: $c < \sqrt{2}$ or, $p < \sqrt{\frac{2 \ln n}{n}}$

We have $\mathbb{E}[X_n] \to \infty$. We use second moment method

$$X_n^2 = \frac{1}{4} \sum_{\substack{i \neq j \\ k \neq \ell}} \mathbb{1}\{ i,j \text{ bad pair and } k,\ell \text{ bad pair} \}$$

$$\mathbb{E}[X_n^2] = \frac{1}{4} \sum_{\substack{i \neq j \\ k \neq \ell}} \mathbb{P}(i,j \text{ bad pair and } k,\ell \text{ bad pair},$$

Case 1: $i = k, \quad j = \ell$

Total number of such cases = $n(n-1)$

$$\mathbb{P}(i,j \text{ bad and } k,\ell \text{ not bad})$$

$$= \mathbb{P}(i,j \text{ bad}) = (1-p)(1-p^2)^{n-2}$$

$$\leq (1-p^2)^{n-2}$$

Case

Case 2: Among $\{i, j, k, l\}$, exactly 3 of them are distinct

Total number of such cases = $3n(n-1)(n-2)$

Let $i = k$, $j \neq l$ (similar for others)

$$\mathbb{P}(i,j \text{ bad and } i,l \text{ bad})$$

$$= \mathbb{P}\Big((i,j) \notin E,\ (i,l) \notin E,\ \text{for every } v \notin \{i,j,l\},$$
$$\text{either } (i,v) \notin E$$
$$\text{or } (i,v) \in E \text{ but } (j,v) \notin E,\ (l,v) \notin E\Big)$$

$$= (1-p^{\bullet})^2 \Big(\underbrace{1-p}_{(i,v) \notin E} + \underbrace{p(1-p)^2}_{(i,v) \in E,\ (j,v) \notin E,\ (k,v) \notin E} \Big)^{n-3}$$

$$= (1-p^{\bullet})^2 (1 - 2p^2 + p^3)^{n-3}$$

$$\leqslant (1 - 2p^2 + p^3)^{n-3} \approx (1-2p^2)^{n-3} \text{ if } p \to 0$$

Case 3: i, j, k, ℓ are all distinct

Total number of such cases $= n(n-1)(n-2)(n-3)$

$\mathbb{P}(i,j \text{ bad and } k,\ell \text{ bad})$

$\leqslant \mathbb{P}\big((i,j) \notin E,\ (k,\ell) \notin E,\ \text{for every } v \notin \{i,j,k,\ell\},$

$(i,v) \notin E \text{ or } (j,v) \notin E,$

$\text{and } (k,v) \notin E \text{ or } (\ell,v) \notin E\big)$

because we are ignoring some edges (which ones?)

v does not lie between i,j and v does not lie between k

$\leqslant (1-p)^2 \left((1-p^2)(1-p^2)\right)^{n-4}$

both (k,v) and (ℓ,v) do not occur

$= (1-p)^2 (1-p^2)^{2(n-4)}$

$\leqslant (1-p^2)^{2n-4}$

$$\mathbb{E}[X_n^2] \leq \frac{1}{4}\left[n^2 (1-p^2)^{n-2} + 3n^3 (1-2p^2)^{n-3} + n^4 (1-p^2)^{2(n-4)} \right]$$

$$\approx \frac{1}{4}\left[n^{2-c^2} + 3n^{3-2c^2} + n^{4-2c^2} \right]$$

when $p = c\sqrt{\frac{\ln n}{n}}$

Also $\mathbb{E} X_n \approx \frac{n^{2-c^2}}{2}$

$$\therefore \frac{\mathbb{E}[X_n^2]}{(\mathbb{E} X_n)^2} \leq 1 + o(1)$$

Spectral theory (for symmetric matrices)

Eigenvalues and eigenvectors 1

- Let $M \in \mathbb{R}^{n \times n}$ be a symmetric matrix
- **Definition:** (λ, v) is eigenvalue-eigenvector pair for M

$$Mv = \lambda v \text{ and } v \neq \mathbf{0}$$

- Geometric meaning:
  - $M : \mathbb{R}^n \to \mathbb{R}^n$ is a linear function
  - Let $y = Mx$
    — In general, y may not have the same direction as x
  - Let $Mv = \lambda v$
    — v is special in the sense that M does not rotate v
    — M only rescales v by λ

Eigenvalues and eigenvectors 2

- How many eigenvectors are possible?
  - Infinite: (λ, v) is eigenpair $\Rightarrow (\lambda, cv)$ is also eigenpair
- How many different directions of eigenvectors are possible?
  - OR: How many eigenvectors v are possible such that $\|v\|_2 = 1$?
  - Can still be infinite:
    — Suppose $Mx = \lambda x$ and $My = \lambda y$ with $y \neq cx$
    — Let $z = (1-\alpha)x + \alpha y$ with $\alpha \in (0, 1)$
    $$Mz = \lambda z \text{ and } Mz' = \lambda z' \text{ for } z' = \frac{z}{\|z\|_2}$$
- How many orthonormal eigenvectors $v_1, v_2, \ldots$ are possible?
  - Orthonormal: $\|v_i\|_2 = 1$ for all i, and $v_i^T v_j = 0$ for $i \neq j$
  - Why look for orthonormal?
    — There can be at most n vectors
    — Let $Mv_1 = \lambda_1 v_1$ and $Mv_2 = \lambda_2 v_2$. If $\lambda_1 \neq \lambda_2$, then $v_1^T v_2 = 0$

Eigenvalues and eigenvectors 3

- **Fact:** M has n eigenvalues, $\lambda_1, \ldots, \lambda_n$
  - These are solutions of the equation, $\det(M - \lambda I) = 0$
    (also holds for non-symmetric matrices)
  - For M real and symmetric
    — All eigenvalues are real
    — Corresponding eigenvectors are real

- **Note:** All eigenvalues of M may not be distinct

- Suppose $\lambda_1, \ldots, \lambda_n$ are all distinct
  - Eigenvectors $v_1, \ldots, v_n$ are orthonormal
    — Let $V = [v_1, \ldots, v_n] \in \mathbb{R}^{n \times n}$, then $V^T V = V V^T = I$
  - Let $\Lambda \in \mathbb{R}^{n \times n}$ diagonal with entries $\lambda_1, \ldots, \lambda_n$

$$MV = V\Lambda \quad \text{or} \quad M = V\Lambda V^T = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

Eigenvalues and eigenvectors 4

- **Fact:** Let λ occurs k times in $\lambda_1, \ldots, \lambda_n$
  - For M symmetric,
    one can find k orthonormal vectors $v_1, \ldots, v_k$ such that $Mv_i = \lambda v_i$
  - Note: This set set of k vectors may not be unique
    — Think of eigenvectors for I

- **Spectral decomposition** (of real symmetric matrix)
  - Let $M \in \mathbb{R}^{n \times n}$ symmetric
  - There are $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$ and orthonormal $v_1, \ldots, v_n \in \mathbb{R}^n$ such that

$$M = V \Lambda V^T = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

- **Implication:** $\{v_1, \ldots, v_n\}$ is an orthonormal basis
  - Every $x \in \mathbb{R}^n$ can be written as $x = \sum_{i=1}^{n} c_i v_i$, and $Mx = \sum_{i=1}^{n} \lambda_i c_i v_i$,

Eigenvalues and eigenvectors 5

Eigenvalues govern various matrix functions **(exercise)**

- $\text{Trace}(M) = \sum_{i=1}^{n} M_{ii} = \sum_{i=1}^{n} \lambda_i$
- $\det(M) = \prod_{i=1}^{n} \lambda_i$
- Let $f(\cdot)$ be a polynomial (example: $f(M) = M^3 + 3M^2 + I$)
  $f(M) = V f(\Lambda) V^T$
- Spectral norm: $\|M\|_2 = \max_{x \neq \mathbf{0}} \frac{\|Mx\|_2}{\|x\|_2} = \max_i |\lambda_i|$
  (hint: use previous slide, and fact $\|x\|_2^2 = x^T x$)

Positive definite matrices

- Let $M \in \mathbb{R}^{n \times n}$ be symmetric
- **Definition:** M is positive semi-definite if

$$x^T M x \geq 0 \text{ for all } x \in \mathbb{R}^n$$

  M is positive definite if $x^T M x > 0$ for all $x \in \mathbb{R}^n, x \neq \mathbf{0}$
- **Results:**
  - M is positive semi-definite $\iff \lambda_i \geq 0$ for all i
  - M is positive definite $\iff \lambda_i > 0$ for all i

  (**why?** — use spectral decomposition)
- **Note:** Alternative terminology
  - M is positive definite if $x^T M x \geq 0$ for all x
  - M is strictly positive definite if $x^T M x > 0$ for all $x \neq \mathbf{0}$

Singular value decomposition 1

- What happens if $M \in \mathbb{R}^{n \times n}$ is not symmetric?
- Some of previous conclusions do not hold
  - (λ_1, x_1) and (λ_2, x_2) are eigen pairs with $\lambda_1 \neq \lambda_2$ — Cannot claim $x_1^T x_2 = 0$
  - M is positive definite may not imply $\lambda_i > 0$ for all i
- What happens if $M \in \mathbb{R}^{m \times n}$ where $m \neq n$?
- **Note:** $M^T M \in \mathbb{R}^{n \times n}$ and $M M^T \in \mathbb{R}^{m \times m}$
  - Both always symmetric and positive semi-definite **(why?)**
  - Let $M^T M = V \widetilde{\Lambda} V^T = \sum_{i=1}^{n} \widetilde{\lambda}_i v_i v_i^T$

    and $M M^T = U \widehat{\Lambda} U^T = \sum_{i=1}^{m} \widehat{\lambda}_i u_i u_i^T$

Singular value decomposition 2

- Assume $m \geq n$ (for convenience)
  - Let $\widetilde{\lambda}_1 \geq \widetilde{\lambda}_2 \geq \ldots \geq \widetilde{\lambda}_n \geq 0$ and $\widehat{\lambda}_1 \geq \widehat{\lambda}_2 \geq \ldots \geq \widehat{\lambda}_m \geq 0$
    — Can show
    $$\widetilde{\lambda}_i = \widehat{\lambda}_i \text{ for } i \leq n \quad \text{and} \quad \widehat{\lambda}_i = 0 \text{ for } n < i \leq m$$

- Let $\Sigma \in \mathbb{R}^{m \times n}$ diagonal (only principle diagonal non-zero) with entries $\sigma_i = \sqrt{\widetilde{\lambda}_i}$ for $i \leq \min\{m, n\}$

- **Singular value decomposition:** We can write $M \in \mathbb{R}^{m \times n}$ as
$$M = U \Sigma V^T = \sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^T$$
  - $\sigma_1, \sigma_2, \ldots$ — singular values
  - $u_1, u_2, \ldots$ — left singular vectors
  - $v_1, v_2, \ldots$ — right singular vectors

Consistency of spectral clustering

Spectral clustering under SBM

Consistency of spectral clustering as $n \to \infty$

Let $p, q \in (0, 1)$ be fixed scalars with $p > q$.

Let $G \sim \text{SBM}(\frac{n}{2}, 2, p, q)$ with underlying split $V = S_1 \cup S_2$.

Let unnormalised spectral clustering outputs the split $\widehat{S}_1, \widehat{S}_2$.

$$\mathbb{P}\left((\widehat{S}_1, \widehat{S}_2) \neq (S_1, S_2)\right) = 1 - o(1)$$

Proof: First part in scanned notes (next few slides).

For second part, we need few tools.

$G \sim SBM\left(\frac{n}{2}, 2, p, q\right)$.

Meaning : If $G = (V, E)$

$V = S_1 \cup S_2 \quad . \quad |S_1| = |S_2| = \frac{n}{2}$

$$\mathbb{P}\left((i,j) \in E\right) = \begin{cases} 0 & \text{if } i = j \\ p & \text{if } i,j \in S_1 \text{ or } i,j \in S_2 \\ q & \text{if } i \in S_1,\ j \in S_2 \text{ or opposite} \end{cases}$$

Let A = adjacency matrix

D = degree matrix

$L = D - A$ = unnormalised Laplacian

Let $\mathcal{A} = \mathbb{E}[A]$, $\mathcal{D} = \mathbb{E}[D]$, $\mathcal{L} = \mathbb{E}[L] = \mathcal{D} - \mathcal{A}$

$\mathcal{A}_{ij} = \mathbb{P}\left((i,j) \in E\right)$

$\mathcal{D}_{ii} = \mathbb{E}[\text{degree}(i)] = \underbrace{p\left(\frac{n}{2} - 1\right) + q\frac{n}{2}}_{= \bar{d}} \quad \Rightarrow \mathcal{D} = \bar{d}\, I$

How does A look like ?

- Let us label nodes as $S_1 = \{1, 2, \ldots, \frac{n}{2}\}$, $S_2 = \{\frac{n}{2}+1, \ldots, n\}$

$$A = \begin{pmatrix} 0 & p & \cdots & q & \cdots \\ p & \ddots & & & \\ \vdots & & 0 & & \\ & & & \ddots & p \\ q & & & p & 0 \end{pmatrix} \begin{matrix} \} \, n/2 \text{ rows} \\ \\ \\ \} \, n/2 \text{ rows} \end{matrix}$$

$$= \begin{pmatrix} p & \cdots & q \\ \vdots & \ddots & \\ q & & p \end{pmatrix} - pI$$

$$= (p-q) \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \\ 0 & & 0 \end{pmatrix} + (p-q) \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \\ 0 & & 1 \end{pmatrix}$$

$$+ q \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \\ 1 & & 1 \end{pmatrix} - pI$$

$$= (p-q) 1_{S_1} 1_{S_1}^T + (p-q) 1_{S_2} 1_{S_2}^T + q 1_n 1_n^T - pI$$

- We need to compute eigenvectors for L.

But, first let us look at eigenvectors for $\mathcal{L}$.

$$\mathcal{L} = \mathcal{D} - \mathcal{A}$$

$$= (\bar{d} + p) I - (p - q)\left(1_{S_1} 1_{S_1}^T + 1_{S_2} 1_{S_2}^T\right) - q\, 1_n 1_n^T$$

- Verify $(0, 1_n)$ is eigenpair for $\mathcal{L}$

- Next consider $f \in \mathbb{R}^n$

$$f_i = \begin{cases} +1 & \text{if } i \in S_1 \\ -1 & \text{if } i \in S_2 \end{cases}$$

Check $1_n^T f = 0$, $\left(1_{S_1} 1_{S_1}^T + 1_{S_2} 1_{S_2}^T\right) f = \frac{n}{2} f$

$$\therefore \mathcal{L} f = \left(\bar{d} + p - (p - q)\frac{n}{2}\right) f = (q n) f$$

$\Rightarrow (qn, f)$ is eigenpair for $\mathcal{L}$

- Next, let $x \in \mathbb{R}^n$ such that $x \perp 1_n$ and $x \perp f$

Verify: $x \perp 1_{S_1}$ and $x \perp 1_{S_2}$

$\Rightarrow \mathcal{L} x = (\bar{d} + P) x = \frac{n}{2}(p+q) x$

$\Rightarrow \left(\frac{n}{2}(p+q), x\right)$ is eigenpair for $\mathcal{L}$

.

Note: $\frac{n}{2}(p+q) > nq$ since $p > q$

Spectrum of $\mathcal{L}$:

$\lambda_1 = 0$, $v_1 = 1_n$

$\lambda_2 = nq$, $v_2 = f$

$\lambda_3 = \ldots = \lambda_n = \frac{n}{2}(p+q)$

If we ran unnormalised spectral clustering on $\mathcal{L}$

- computed eigenvector will be $\frac{1}{\sqrt{n}} f$ (normalised to have $\|\cdot\|_2 = 1$)
- output partition: S_1, S_2

Spectral perturbation theory 1

- Let $M \in \mathbb{R}^{n \times n}$ and $E \in \mathbb{R}^{n \times n}$ symmetric
- Let $M' = M + E$
  — Think of M' as a noisy observation of M
- How far are eigenvalues and eigenvectors of M and M'?

Weyl's inequality (simplified)

Let $\lambda_1 \leq \ldots \leq \lambda_n$ be eigenvalues of M, and
$\lambda'_1 \leq \ldots \leq \lambda'_n$ be eigenvalues of M'.

$$|\lambda_i - \lambda'_i| \leq \|E\|_2$$

Spectral perturbation theory 2

- What happens to eigenvectors?
- Difficult to answer since eigenvectors can be rotated
  - (λ, v) is eigenpair $\implies (\lambda, -v)$ is also eigenpair
  - But $\|v - (-v)\|_2$ can be very large
  - More complicated if λ has multiplicity more than 1
- Clear answer by Davis-Kahan perturbation theory
  - Complicated since it takes care of also possible rotation and rescaling of eigenvectors

Applying spectral perturbation theory

Eigenvector perturbation in our setting

Let $\widehat{f}$ be eigenvector of L corresponding to second smallest eigenvalue.

Let f be eigenvector of $\mathcal{L}$ corresponding to second smallest eigenvalue.

Let $\|\widehat{f}\|_2 = \|f\|_2 = 1$.

$$\min\left\{\|\widehat{f} - f\|_2, \|\widehat{f} + f\|_2\right\} \leq \frac{4\|L - \mathcal{L}\|_2}{\min\{\lambda_2 - \lambda_1, \lambda_3 - \lambda_2\}}$$

where $\lambda_1 \leq \ldots \leq \lambda_n$ be eigenvalues of $\mathcal{L}$

Observe: Denominator in bound is $\frac{n}{2}\min\{q, p - q\}$
(grows linearly with n)

How large is $\|L - \mathcal{L}\|_2$?

Bounding $\|L - \mathcal{L}\|_2$ 1

$$\|L - \mathcal{L}\|_2 \leq \|D - \mathcal{D}\|_2 + \|A - \mathcal{A}\|_2$$

Bounding the first term:

- $\|D - \mathcal{D}\|_2 = \max_i \left|d_i - \mathbb{E}[d_i]\right| = \max_i \left|d_i - \overline{d}\right|$
- Similar to Assignment-2, we can show for every i

$$|d_i - \overline{d}| \leq 2\sqrt{n \ln\left(\frac{1}{\delta}\right)} \text{ with probability } 1 - \delta$$

- Applying union bound, we get

$$\max_i |d_i - \overline{d}| \leq 2\sqrt{n \ln\left(\frac{n}{\delta}\right)} \text{ with probability } 1 - \delta$$

- So $\|D - \mathcal{D}\|_2 \leq C_\delta \sqrt{n \ln n}$

Bounding $\|L - \mathcal{L}\|_2$ 2

Bounding the second term:

- $A - \mathcal{A}$ is symmetric random matrix with
  - independent entries in $[-1, 1]$
  - entries have mean zero

Concentration of random matrix

$M \in \mathbb{R}^{n \times n}$ is a symmetric random matrix with following properties:

- $M_{ii} = 0$ for all i (zero diagonal)
- $\{M_{ij} : i < j\}$ are mutually independent
- $\mathbb{E}[M_{ij}] = 0$ and $|M_{ij}| \leq 1$ almost surely for all i, j

For any $\delta \in (0, 1)$,

$$\mathbb{P}\left(\|M\|_2 \geq C'_\delta \sqrt{n}\right) \leq \delta$$

for some constant $C'_\delta > 0$ that depends only on δ.

Finishing the proof

$$\min\left\{\|\widehat{f}-f\|_2, \|\widehat{f}+f\|_2\right\} \leq \frac{4\|L-\mathcal{L}\|_2}{\frac{n}{2}\min\{q, p-q\}}$$
$$\leq C''_\delta \sqrt{\frac{\ln n}{n}}$$

- Bound holds with probability $1-2\delta$
- If we let $\delta \to 0$ as $n \to \infty$,

  C''_δ grows slowly
- So as $n \to \infty$, $\widehat{f} \approx \pm f$ with probability $1-o(1)$
- Hence, partitioning is also correct with probability $1-o(1)$