Algorithmic Techniques for Modeling and Mining Large Graphs (AMAzING)

Alan Frieze [1] Aristides Gionis [2]
Charalampos E. Tsourakakis [2]

[1]Carnegie Mellon University

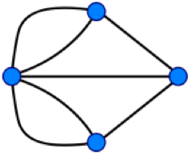
[2]Aalto University

Tutorial website:
http://www.math.cmu.edu/~ctsourak/kdd13.html

Introduction to graphs and networks

Graphs: a simple model

- entities – set of vertices
- pairwise relations among vertices
  – set of edges
- can add directions, weights,...
- graphs can be used to model many real datasets
  - people who are friends
  - computers that are interconnected
  - web pages that point to each other
  - proteins that interact

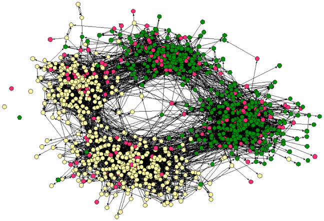


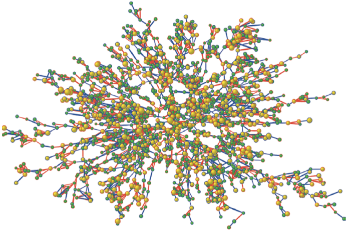
Graph theory

- graph theory started in the 18th century, with Leonhard Euler
  - the problem of Königsberg bridges
  - since then, graphs have been studied extensively

Analysis of graph datasets in the past

- graphs datasets have been studied in the past
  e.g., networks of highways, social networks
  - usually these datasets were small
  - visual inspection can reveal a lot of information

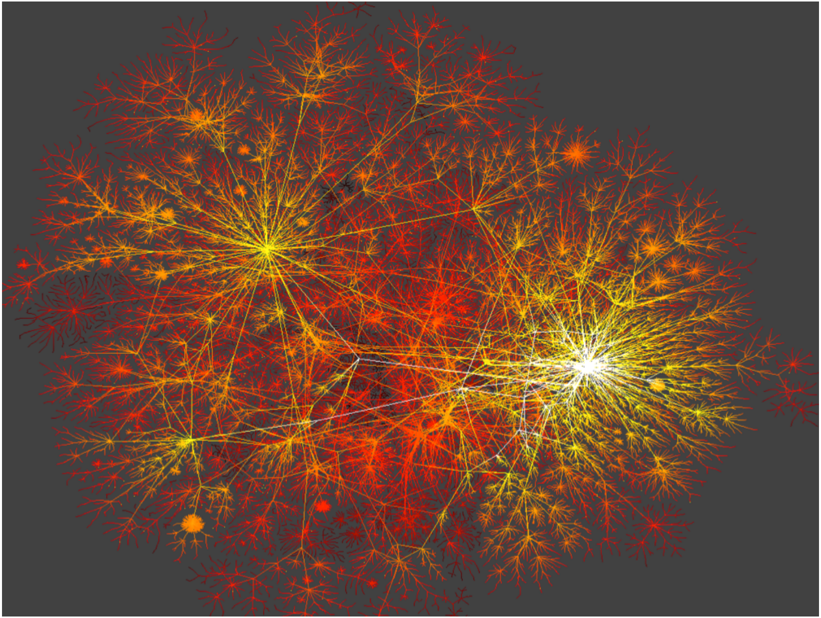




Analysis of graph datasets now

- more and larger networks appear
  - products of technological advancement
    - e.g., internet, web
  - result of our ability to collect more, better-quality, and more complex data
    - e.g., gene regulatory networks
- networks of thousands, millions, or billions of nodes
  - impossible to visualize

The internet map

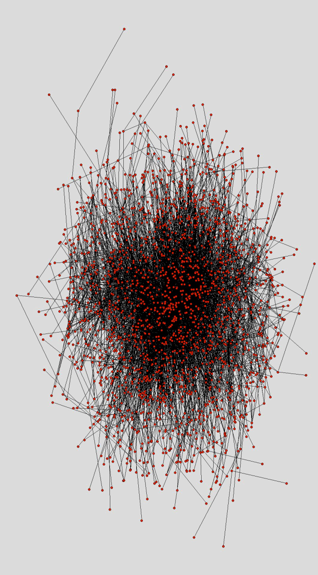


Types of networks

- social networks
- knowledge and information networks
- technology networks
- biological networks

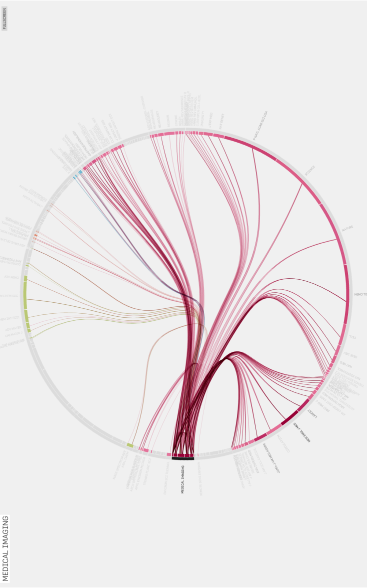
Social networks

- links denote a social interaction
    - networks of acquaintances
    - collaboration networks
        - actor networks
        - co-authorship networks
        - director networks
    - phone-call networks
    - e-mail networks
    - IM networks
    - sexual networks



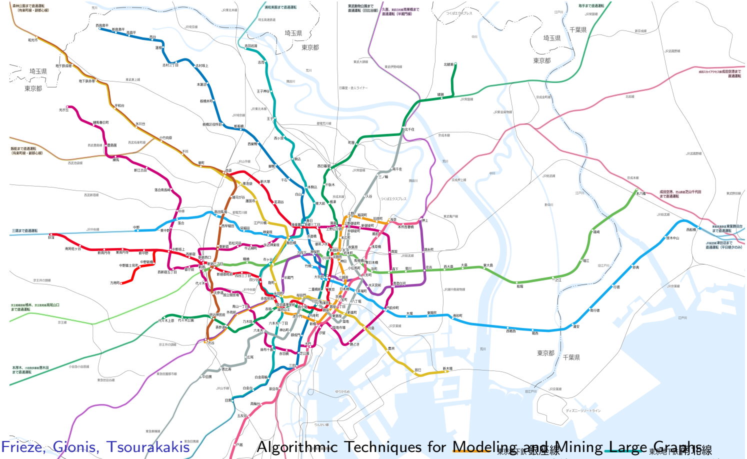
Knowledge and information networks

- nodes store information, links associate information
  - citation network (directed acyclic)
  - the web (directed)
  - peer-to-peer networks
  - word networks
  - networks of trust
  - software graphs
  - bluetooth networks
  - home page/blog networks

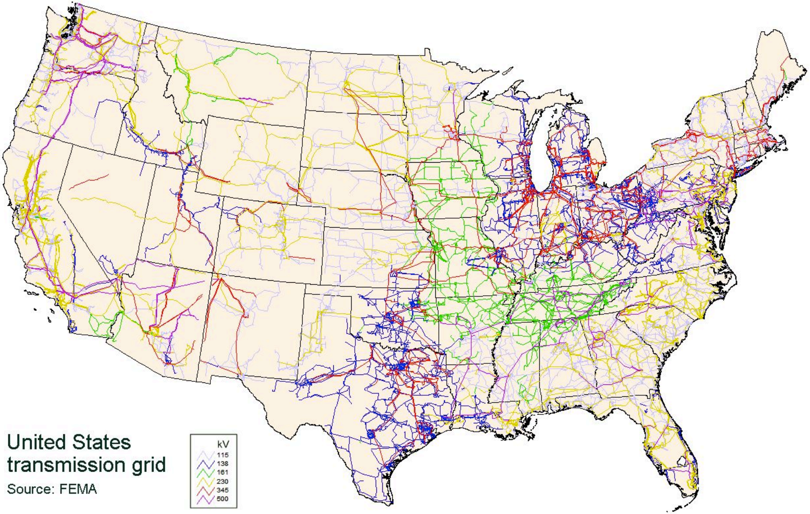


Technological networks

- networks built for distribution of a commodity
  - the internet, power grids, telephone networks
  - airline networks, transportation networks

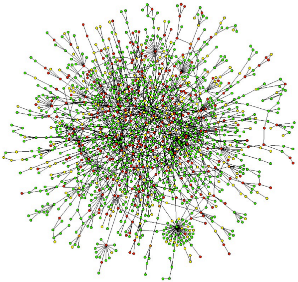


US power grid



Biological networks

- biological systems represented as networks
  - protein-protein interaction networks
  - gene regulation networks
  - gene co-expression networks
  - metabolic pathways
  - the food web
  - neural networks



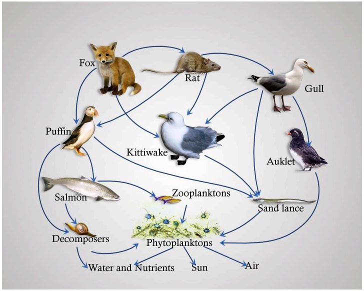
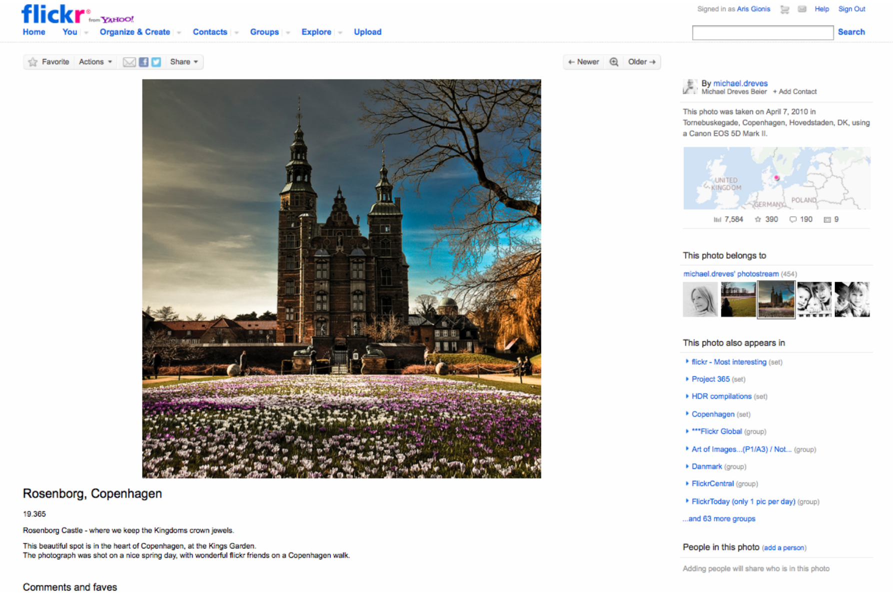


Photo-sharing site



What is the underlying graph?

- nodes: photos, tags, users, groups, albums, sets, collections, geo, query, ...
- edges: upload, belong, tag, create, join, contact, friend, family, comment, fave, search, click, ...
- also many interesting induced graphs
  - tag graph: based on photos
  - tag graph: based on users
  - user graph: based on favorites
  - user graph: based on groups

- which graph to pick — not an easy choice

Recurring theme

- social media, user-generated content
- user interaction is composed by many atomic actions
  - post, comment, like, mark, join, comment, fave, thumps-up, ...
  - generates all kind of interesting graphs to mine

Network science

- the world is full with networks
- what do we do with them?
  - understand their topology and measure their properties
  - study their evolution and dynamics
  - create realistic models
  - create algorithms that make use of the network structure

Outline

- introduction and graphs and networks
- random graphs as models of real-world networks
  - properties of real-world networks
  - Erdős-Rényi graphs
  - models of real-world networks
  - applications of random graphs
- algorithm design for large-scale networks
  - graph partitioning and community detection
  - dense subgraphs

Properties of real-world networks

Properties of real-world networks

diverse collections of graphs arising from different phenomena

are there typical patterns?

- static networks
  1. heavy tails
  2. clustering coefficients
  3. communities
  4. small diameters
- time-evolving networks
  1. densification
  2. shrinking diameters
- web graph
  1. bow-tie structure
  2. bipartite cliques

Heavy tails

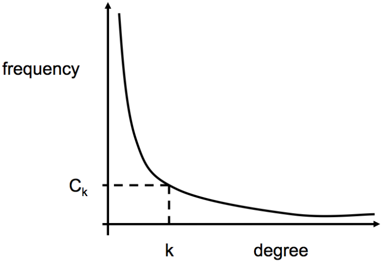
What do the proteins in our bodies, the Internet, a cool collection of atoms and sexual networks have in common? One man thinks he has the answer and it is going to transform the way we view the world.

Scientist 2002

Albert-László Barabási

Degree distribution

- C_k = number of vertices with degree k



- problem : find the probability distribution that fits best the observed data

Power-law degree distribution

- C_k = number of vertices with degree k, then

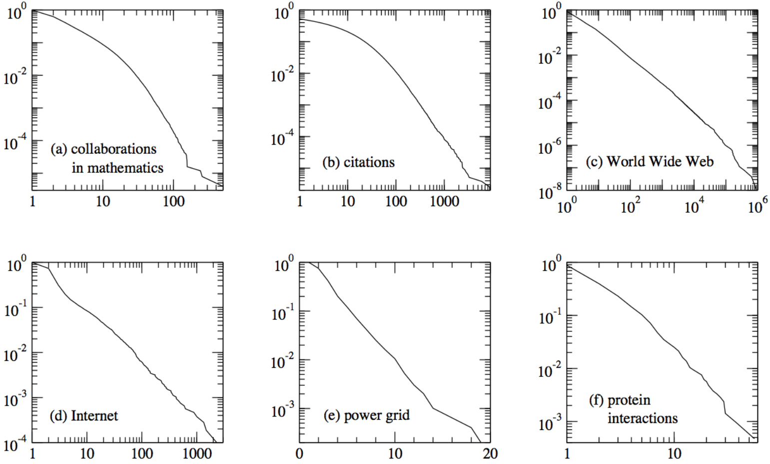
$$C_k = ck^{-\gamma}$$

  with $\gamma > 1$, or

$$\ln C_k = \ln c - \gamma \ln k$$

- plotting $\ln C_k$ versus $\ln k$ gives a straight line with slope $-\gamma$
- heavy-tail distribution : there is a non-negligible fraction of nodes that has very high degree (hubs)
- scale free : average is not informative

Power-law degree distribution



power-laws in a wide variety of networks ([Newman, 2003])
sheer contrast with Erdős-Rényi random graphs

Power-law degree distribution

do the degrees follow a power-law distribution?

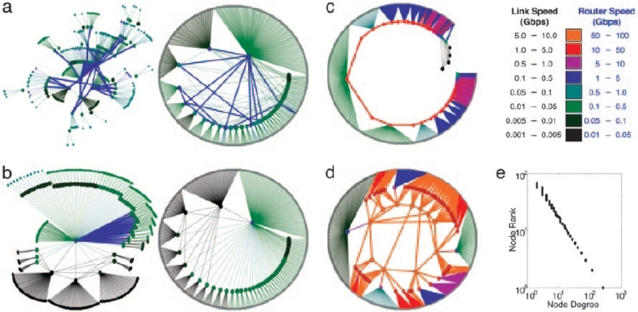
three problems with the initial studies

- graphs generated with traceroute sampling, which produces power-law distributions, even for regular graphs [Lakhina et al., 2003].
- methodological flaws in determining the exponent see [Clauset et al., 2009] for a proper methodology
- other distributions could potentially fit the data better but were not considered, e.g., lognormal.

disclaimer: we will be referring to these distributions as heavy-tailed, avoiding a specific characterization

Power-law degree distribution

- frequently, we hear about "scale-free networks" correct term is networks with scale-free degree distribution



all networks above have the same degree sequence but structurally are very different (source [Li et al., 2005])

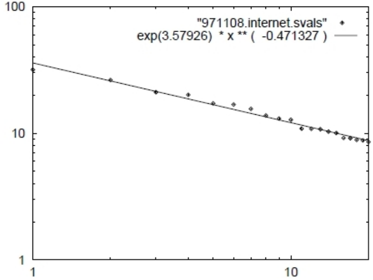
Maximum degree

- for random graphs, the maximum degree is highly concentrated around the average degree z
- for power-law graphs

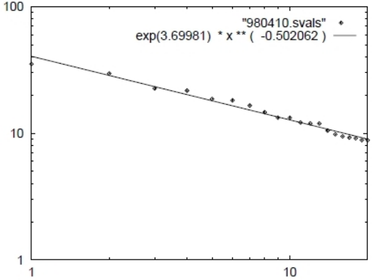
$$d_{\max} \approx n^{1/(\alpha-1)}$$

- hand-waving argument: solve $\quad n \Pr[X \geq d] = \Theta(1)$

Heavy tails, eigenvalues



(a) Int-11-97

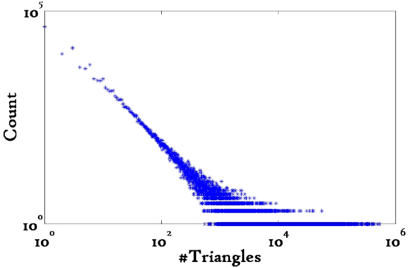


(b) Int-04-98

log-log plot of eigenvalues of the Internet graph in decreasing order

again a power law emerges [Faloutsos et al., 1999]

Heavy tails, triangles



- triangle distribution in `flickr`
- figure shows the count of nodes with k triangles vs. k in log-log scale
- again, heavy tails emerge [Tsourakakis, 2008]

Clustering coefficients

- a proposed measure to capture local clustering is the graph transitivity

$$T(G) = \frac{3 \times \text{number of triangles in the network}}{\text{number of connected triples of vertices}}$$

- captures "transitivity of clustering"
- if u is connected to v and
  v is connected to w, it is also likely that
  u is connected to w

Clustering coefficients

- alternative definition
- local clustering coefficient

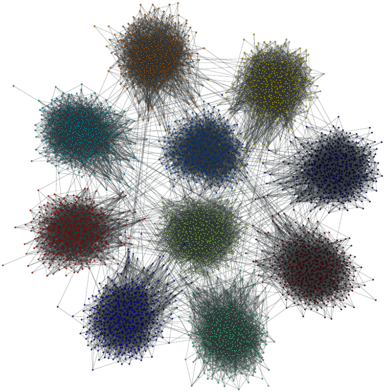
$$C_i = \frac{\text{Number of triangles connected to vertex } i}{\text{Number of triples centered at vertex } i}$$

- global clustering coefficient

$$C(G) = \frac{1}{n} \sum_i C_i$$

Community structure

loose definition of community: a set of vertices densely connected to each other and sparsely connected to the rest of the graph



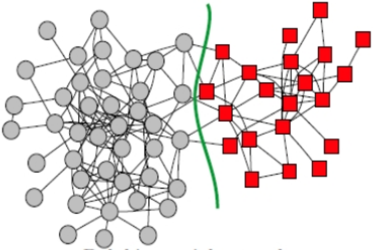
artificial communities:
http://projects.skewed.de/graph-tool/

Community structure

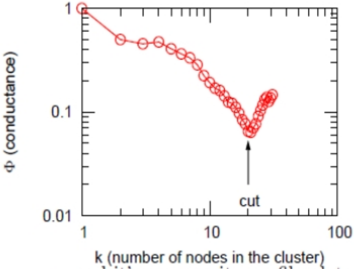
[Leskovec et al., 2009]

- study community structure in an extensive collection of real-world networks
- authors introduce the network community profile plot
- it characterizes the best possible community over a range of scales

Community structure



Dolphins social network ...

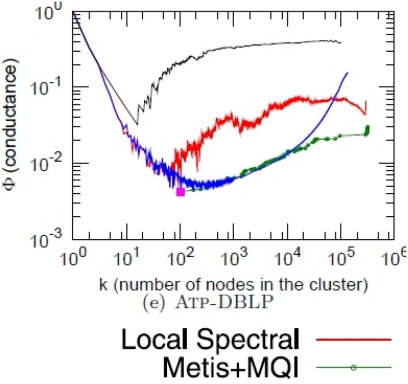


...and it's community profile plot

`dolphins` network and its NCP
(source [Leskovec et al., 2009])

Community structure

- do large-scale real-world networks have this nice artifical structure? NO!



NCP of a DBLP graph (source [Leskovec et al., 2009])

Community structure

important findings of [Leskovec et al., 2009]

1. up to a certain size k ($k \sim 100$ vertices) there exist good cuts
   - as the size increases so does the quality of the community
2. at the size k we observe the best possible community
   - such communities are typically connected to the remainder with a single edge
3. above the size k the community quality decreases
   - this is because they blend in and gradually disappear

Small-world phenomena

small worlds : graphs with short paths

- Stanley Milgram (1933-1984)
  "The man who shocked the world"
- obedience to authority (1963)
- small-World experiment (1967)

- we live in a small-world
- for criticism on the small-world experiment, see *"Could It Be a Big World After All? What the Milgram Papers in the Yale Archives Reveal About the Original Small World Study"* by Judith Kleinfeld

Small-world experiments

- letters were handed out to people in Nebraska to be sent to a target in Boston
- people were instructed to pass on the letters to someone they knew on first-name basis
- the letters that reached the destination (64 / 296) followed paths of length around 6
- *Six degrees of separation* : (play of John Guare)
- also:
  - the Kevin Bacon game
  - the Erdős number
- small-World project: http://smallworld.columbia.edu/index.html

Small diameter

proposed measures

- diameter : largest shortest-path over all pairs.
- effective diameter : upper bound of the shortest path of 90% of the pairs of vertices.
- average shortest path : average of the shortest paths over all pairs of vertices.
- characteristic path length : median of the shortest paths over all pairs of vertices.
- hop-plots : plot of $|N_h(u)|$, the number of neighbors of u at distance at most h, as a function of h [Faloutsos et al., 1999].

Other properties

- assortativity
- distribution of size of connected components
- distribution of motifs
- ...

Time-evolving networks

J. Leskovec

J. Kleinberg

C. Faloutsos

[Leskovec et al., 2005b]

- densification power law:

$$|E_t| \propto |V_t|^\alpha \qquad 1 \leq \alpha \leq 2$$

- shrinking diameters: diameter is shrinking over time.

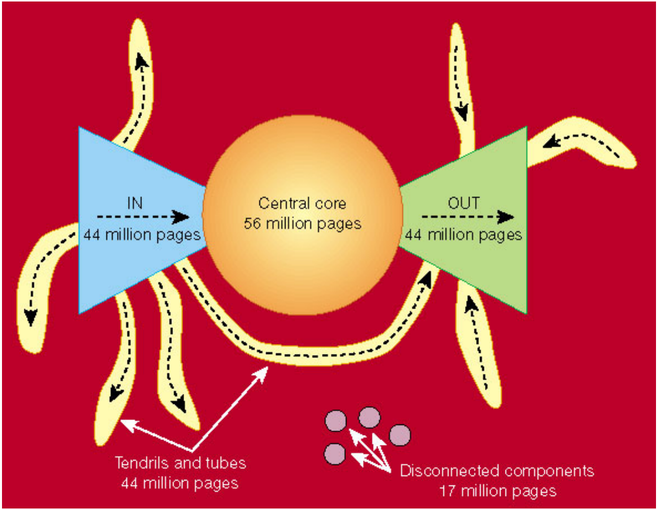
Web graph

- the Web graph is a particularly important real-world network

> *Few events in the history of computing have wrought as profound an influence on society as the advent and growth of the World Wide Web*
>
> [Kleinberg et al., 1999a]

- vertices correspond to static web pages
- directed edge (i, j) models a link from page i to page j
- will discuss two structural properties of the web graph:
  1. the bow-tie structure [Broder et al., 2000]
  2. abundance of bipartite cliques [Kleinberg et al., 1999a, Kumar et al., 2000]

Web is a bow-tie

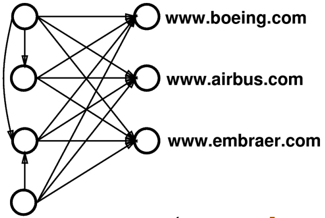


(source [Broder et al., 2000])

Bipartite subgraphs

- websites that are part of the same community frequently do not reference one another

  (competitive reasons, disagreements, ignorance) [Kumar et al., 1999].
- similar websites are *co-cited*
- therefore, web communities are characterized by dense directed bipartite subgraphs



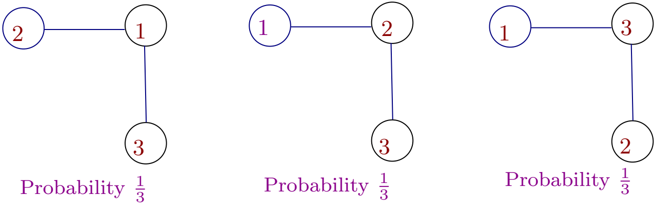
(source [Kleinberg et al., 1999a])

Erdős-Rényi graphs

Random graphs

- a random graph is a set of graphs together with a probability distribution on that set

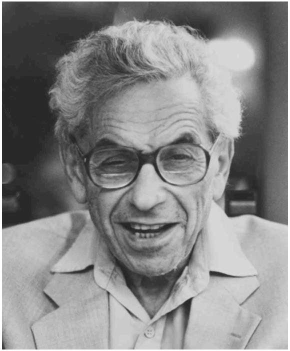
- example



a random graph on $\{1, 2, 3\}$ with 2 edges with the uniform distribution

Random graphs

- Erdős-Rényi (or Gilbert-Erdős-Rényi) random graph model



Paul Erdős
1913 – 1996

Alfréd Rényi
1921 – 1970

Random graphs

- the $G(n, p)$ model:
- n : the number of vertices
- $0 \leq p \leq 1$: probability
- for each pair (u, v), independently generate the edge (u, v) with probability p
- $G(n, p)$ a family of graphs, in which a graph with m edges appears with probability $p^m(1-p)^{\binom{n}{2}-m}$

- the $G(n, m)$ model: related, but not identical

Properties of random graphs

- a property P holds almost surely/with high probability ($\mathbf{whp} \to 1 - o(1)$) if

$$\lim_{n\to\infty} \Pr[G \text{ has } P] = 1$$

- which properties hold as p increases?
- threshold phenomena : many properties appear suddenly
- there exist a probability p_c such that

  for $p < p_c$ the property does not hold a.s.

  for $p > p_c$ the property holds a.s.

The giant component

- let $z = np$ be the average degree
- if $z < 1$ the largest component has size $O(\log n)$ a.s.
- if $z > 1$ the largest component has size $\Theta(n)$ a.s.;
  the second largest component has size $O(\log n)$ a.s.
- if $z = \omega(\log n)$ the graph is connected a.s.

Phase transition

- if $z = 1$ there is a phase transition
  - the largest component has size $O(n^{2/3})$
  - the sizes of the components follow a power-law

Phase transition — proof sketch

Michael Krivelevich

Benny Sudakov

the phase transition in random graphs — a simple proof

The Erdős-Rényi paper, which launched the modern theory of random graphs, has had enormous influence on the development of the field and is generally considered to be a single most important paper in Probabilistic Combinatorics, if not in all of Combinatorics

Phase transition — proof sketch

[Krivelevich and Sudakov, 2013] give a simple proof for the transition based on running depth first search (DFS) on G

- S : vertices whose exploration is complete
- T : unvisited vertices
- $U = V - (S \cup T)$: vertices in stack

observation:

- the set U always spans a path
- when a vertex u is added in U, it happens because u is a neighbor of the last vertex v in U; thus, u augments the path spanned by U, of which v is the last vertex
- epoch is the period of time between two consecutive emptyings of U
- each epoch corresponds to a connected component

Phase transition — proof sketch

Lemma

Let $\epsilon > 0$ *be a small enough constant and let* $N = \binom{n}{2}$
Consider the sequence $\bar{X} = (X_i)_{i=1}^{N}$ *of i.i.d. Bernoulli random variables with parameter* p

1. *let* $p = \frac{1-\epsilon}{n}$ *and* $k = \frac{7}{\epsilon^2} \ln n$
   then **whp** *there is no interval of length* kn *in* $[N]$*, in which at least* k *of the random variables* X_i *take value 1*
2. *let* $p = \frac{1+\epsilon}{n}$ *and* $N_0 = \frac{\epsilon n^2}{2}$
   then **whp** $\left|\sum_{i=1}^{N_0} X_i - \frac{\epsilon(1+\epsilon)n}{2}\right| \leq n^{2/3}$

Phase transition — useful tools

Lemma (Union bound)

For any events $A_1, \ldots, A_n$, $\mathbf{Pr}\left[A_1 \cup \ldots A_n\right] \leq \sum_{i=1}^{n} \mathbf{Pr}\left[A_i\right]$

Lemma (Chebyshev's inequality)

Let X *be a random variable with finite expectation* $\mathbb{E}\left[X\right]$ *and finite non-zero variance* $\mathbb{V}ar\left[X\right]$*. Then for any* $t > 0$,

$$\mathbf{Pr}\left[|X - \mathbb{E}\left[X\right]| \geq t\right] \leq \frac{\mathbb{V}ar\left[X\right]}{t^2}$$

Lemma (Chernoff bound, upper tail)

Let $0 \leq \epsilon \leq 1$*. Then,*

$$\mathbf{Pr}\left[Bin(n, p) \geq (1+\epsilon)np\right] \leq e^{-\frac{\epsilon^2}{3}np}$$

Phase transition — proof sketch

Proof.

- fix interval I of length kn in $[N]$, $N = \binom{n}{2}$
  then $\sum_{i \in I} X_i \sim Bin(kn, p)$
  1. apply Chernoff bound to the upper tail of $B(kn, p)$.
  2. apply union bound on all $(N - k + 1)$ possible intervals of length kn
  - upper bound the probability of the existence of a violating interval

$$(N - k + 1) Pr[B(kn, p) \geq k] < n^2 \cdot e^{-\frac{\epsilon^2}{3}(1-\epsilon)k} = o(1)$$

- sum $\sum_{i=1}^{N_0} X_i$ distributed binomially (params N_0 and p)
- expectation: $N_0 p = \frac{\epsilon n^2 p}{2} = \frac{\epsilon(1+\epsilon)n}{2}$
- standard deviation of order n
- applying Chebyshev's inequality gives the estimate

□

Phase transition — proof sketch

Proof.

CASE I: $p = \frac{1-\epsilon}{n}$

- assume to the contrary that G contains a connected component C with more than $k = \frac{7}{\epsilon^2} \ln n$ vertices
- consider the moment inside this epoch when the algorithm has found the $(k+1)$-st vertex of C and is about to move it to U
- denote $\Delta S = S \cap C$ at that moment then $|\Delta S \cup U| = k$, and thus the algorithm got exactly k positive answers to its queries to random variables X_i during the epoch, with each positive answer being responsible for revealing a new vertex of C, after the first vertex of C was put into U in the beginning of the epoch.

$\square$

Phase transition — proof sketch

Proof.

- at that moment during the epoch only pairs of edges touching $\Delta S \cup U$ have been queried, and the number of such pairs is therefore at most $\binom{k}{2} + k(n-k) < kn$
- it thus follows that the sequence $\bar{X}$ contains an interval of length at most kn with at least k 1's inside — a contradiction to Property 1 of Lemma 1

CASE II: $p = \frac{1+\epsilon}{n}$

- same type of argument:

  assume the result does not hold and reach a contradiction by examining carefully the number of queries

$\square$

Degree distribution

- degree distribution : binomial

$$C_k = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

- the limit distribution of the normalized binomial distribution $Bin(n, p)$ is the normal distribution provided that $np(1-p) \to +\infty$ as $n \to +\infty$.
- if $p = \frac{\lambda}{n}$ the limit distribution of $Bin(n, p)$ is the Poisson distribution.

Degree distribution

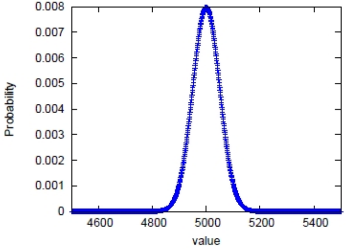


FIGURE 1. The Binomial distribution $B(10000, 0.5)$.

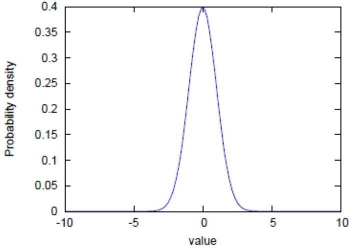


FIGURE 2. The Standard normal distribution $N(0, 1)$.

Bin$(10000, 0.5)$ and Gaussian(0,1)

Degree distribution

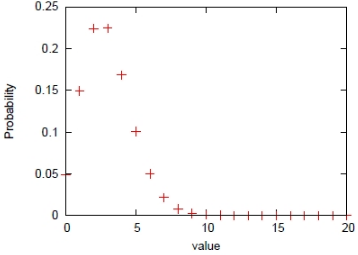


FIGURE 3. The Binomial distribution $B(1000, 0.003)$.

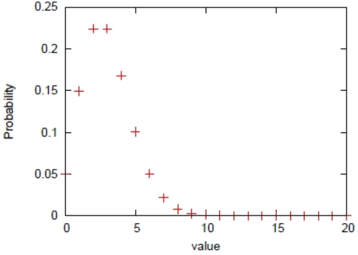


FIGURE 4. The Poisson distribution $P(3)$.

$Bin(1000, 0.003)$ and Poisson(3)

Degree distribution

Theorem

Let $p = \frac{\log n}{n} \cdot \omega(n)$ with $\omega(n) \to +\infty$ arbitrarily slowly. Fix $x \in G$ and $\epsilon > 0$. Then in $G(n,p)$ **whp** *for all vertices x*

$$deg(x) \sim (n-1)p$$

Theorem ([McKay and Wormald, 1997])

Let X_k be the number of vertices of degree k in $G(n,p)$ when $p = \frac{c}{n}$, with $c > 0$ constant. Then **whp** *for $k = 0, 1, \ldots$*

$$\frac{c^k e^{-c}}{k!} \leq \frac{X_k}{n} \leq (1+\epsilon)\frac{c^k e^{-c}}{k!}, \text{ as } n \to +\infty$$

Random graphs and real datasets

- a beautiful and elegant theory studied exhaustively
- have been used as idealized generative models
- unfortunately, they don't always capture reality...

Models of real-world networks

Models

1. classic
   - grown versus static random graphs (CHKNS)
   - growth with preferential attachment
   - structure + randomness $\rightarrow$ small-world networks
2. more models
   - Copying model
   - Cooper-Frieze model
   - Kronecker graphs
   - Chung-Lu model
   - Forest-fire model

CHKNS model

Callaway, Hopcroft, Kleinberg, Newman and Strogatz
[Callaway et al., 2001]

- simple growth model for a random graph without preferential attachment
- main thesis: grown graphs, however randomly they are constructed, are fundamentally different from their static random-graph counterparts

CHKNS model

- start with 0 vertices at time 0.
- at time t, a new vertex is created
- with probability δ add a random edge by choosing two existing vertices uniformly at random

CHKNS model

let $d_k(t)$ be the number of vertices of degree k at time t

then

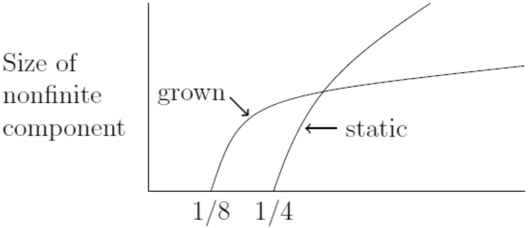
$$\mathbb{E}\left[d_0(t+1)\right] = \mathbb{E}\left[d_0(t)\right] + 1 - \delta \frac{2\mathbb{E}\left[d_0(t)\right]}{t}$$

$$\mathbb{E}\left[d_k(t+1)\right] = \mathbb{E}\left[d_k(t)\right] + \delta \Big(\frac{2\mathbb{E}\left[d_{k-1}(t)\right]}{t} - \frac{2\mathbb{E}\left[d_k(t)\right]}{t} \Big)$$

it turns out that

$$\frac{\mathbb{E}\left[d_k(t)\right]}{t} = \frac{1}{2\delta+1} \Big(\frac{2\delta}{2\delta+1} \Big)^k$$

CHKNS model



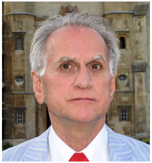
size of giant component for a CHKNS random graph and a static random graph with the same degree distribution

- why are grown and static random graphs so different?
- intuition:
  - positive correlation between the degrees of connected vertices in the grown graph
  - older vertices tend to have higher degree, and to link with other high degree vertices, merely by virtue of their age

Preferential attachment

R. Albert

L. Barabási



B. Bollobás

O. Riordan

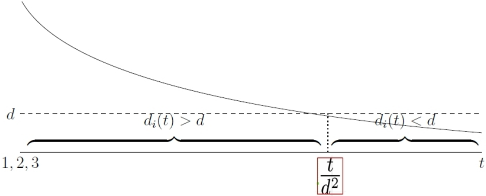
growth model:

- at time n, vertex n is added to the graph
- one edge is attached to the new vertex
- the other vertex is selected at random with probability proportional to its degree
- obtain a sequence of graphs $\{G_1^{(n)}\}$.

Preferential attachment — generalization

The case of $G_m^{(n)}$ where instead of a single edge we add m edges reduces to $G_1^{(n)}$ by creating a $G_1^{(nm)}$ and then collapsing vertices $km, km-1, \ldots, (k-1)m+1$ to create vertex k.

Preferential attachment



at time t, vertices 1 to $\frac{1}{d^2}$ have degrees greater than d (Source [Hopcroft and Kannan, 2012])

heuristic analysis

- $\deg_i(t)$ the *expected* degree of the i-th vertex at time t
- the probability an edge is connected to i is $\frac{\deg_i(t)}{2t}$
- therefore

$$\frac{\partial \deg_i(t)}{\partial t} = \frac{\deg_i(t)}{2t}$$

- the solution is $\deg_i(t) = \sqrt{\frac{t}{i}}$

Preferential attachment

$$\int_0^d \mathbf{Pr}\left[\text{degree} = d\right] \partial d = \mathbf{Pr}\left[\text{degree} \leq d\right] = 1 - \frac{1}{d^2}$$

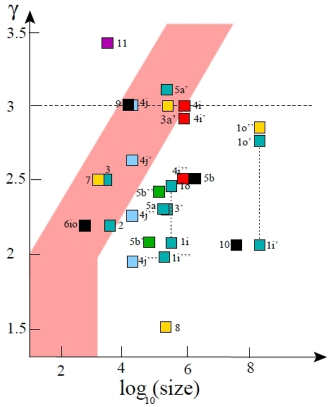
by using the fact that $d_i(t) < d$ if $i > \frac{t}{d^2}$ and by taking the derivative

$$\mathbf{Pr}\left[\text{degree} = d\right] = \frac{\partial}{\partial d}\left(1 - \frac{1}{d^2}\right) = \frac{2}{d^3}$$

power law distribution!

these results can be proved rigorously using the linearized chord diagrams (LCD) model and also prove strong concentration around the expectation using martingales

Generalized preferential attachment



log-linear plot of the exponents of all the networks reported as having power-law (source [Dorogovtsev and Mendes, 2002])

many real-world networks have a power-law slope $2 < \alpha < 3$

Generalized preferential attachment

how can we tune the power-law slope?

- [Buckley and Osthus, 2004] analyze a modified preferential attachment process where $\alpha > 0$ is a *fitness* parameter
- when t vertex comes in, it chooses i according to

$$\mathbf{Pr}\left[t \text{ chooses } i\right] = \begin{cases} \frac{\deg_{t-1}(i)+\alpha-1}{(\alpha+1)t-1}, & \text{if } 1 \le i \le t-1 \\ \frac{\alpha}{(\alpha+1)t-1}, & \text{if } i = t \end{cases}.$$

- $\alpha = 1$ gives the Barabási-Albert/Bollobás-Riordan $G_1^{(n)}$ model
- the power-law slope is $2 + \alpha$.

Generalized preferential attachment

- clustering coefficient of $G_m^{(n)}$ is $\frac{(m-1)\log^2 n}{8n}$ in expectation
- therefore tends to 0 [Bollobás and Riordan, 2003].
- can also be fixed by generalizing the model [Holme and Kim, 2002, Ostroumova et al., 2012].
- triangle formation: if an edge between v and u was added in the previous preferential attachment step, then add one more edge from v to a randomly chosen neighbor of u.

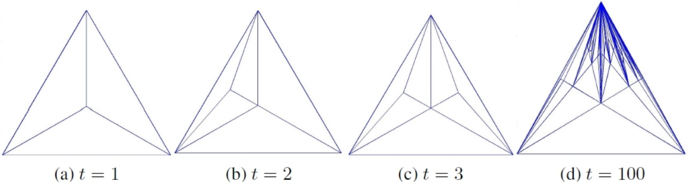
Holme-Kim Model

- perform a preferential attachment step
- the perform with probability β_t another preferential attachment step or a triangle formation step with probability $1 - \beta_t$

diameter for PA and GPA is $\frac{\log n}{\log \log n}$ and $\log n$ respectively

Random Apollonian networks

are there power-law planar graphs?

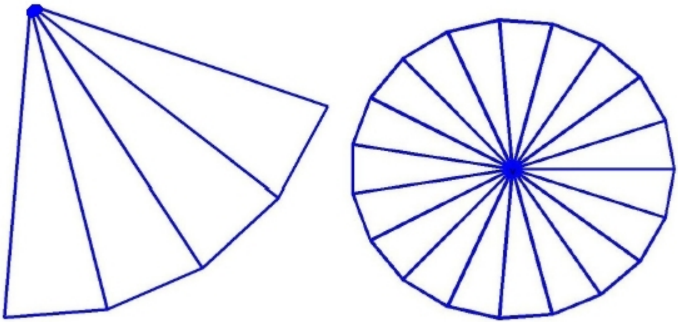


snapshots of a random Apollonian network (RAN) at:
(a) $t = 1$ (b) $t = 2$ (c) $t = 3$ (d) $t = 100$

- at time $t + 1$ we choose a face F uniformly at random among the faces of G_t
- let (i, j, k) be the vertices of F
- we add a new vertex inside F and we connect it to i, j, k

Random Apollonian networks

Preferential attachment mechanism



what each vertex “sees” (boundary and the rest respectively)

Random Apollonian networks

Theorem ([Frieze and Tsourakakis, 2013])

Let $Z_k(t)$ *denote the number of vertices of degree* k *at time* t, $k \geq 3$. *For any* $t \geq 1$ *and any* $k \geq 3$ *there exists a constant* b_k *depending on* k *such that*

$$|\mathbb{E}\left[Z_k(t)\right] - b_k t| \leq K, \quad \textit{where} \quad K = 3.6.$$

Furthermore, for t *sufficiently large and any* $\lambda > 0$

$$\mathbf{Pr}\left[|Z_k(t) - \mathbb{E}\left[Z_k(t)\right]| \geq \lambda\right] \leq e^{-\frac{\lambda^2}{72t}}$$

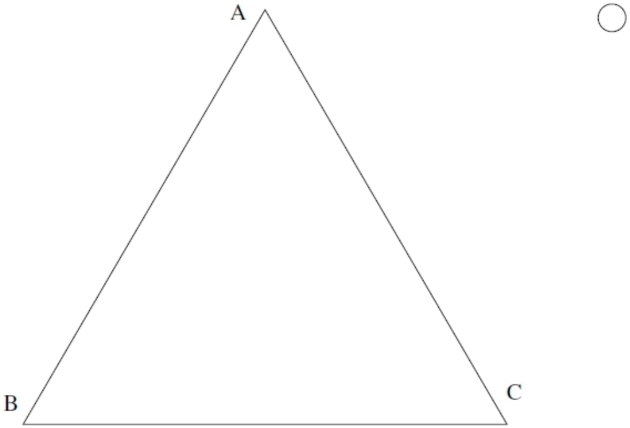
Corollary

The diameter $d(G_t)$ *of* G_t *satisfies asymptotically* **whp**

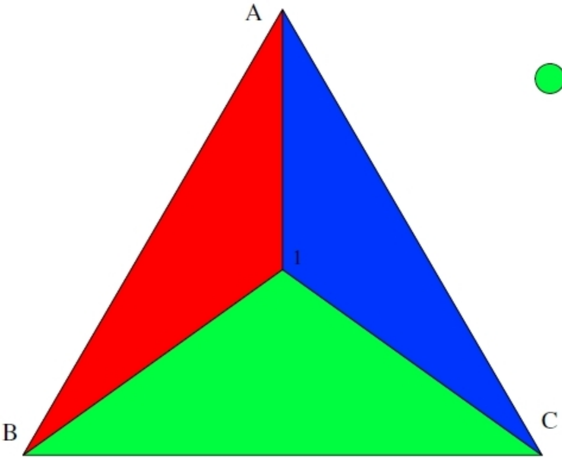
$$\mathbf{Pr}\left[d(G_t) > 7.1 \log t\right] \to 0$$

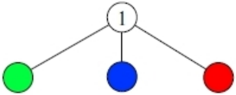
Random Apollonian networks

key idea: establish a bijection with random ternary trees

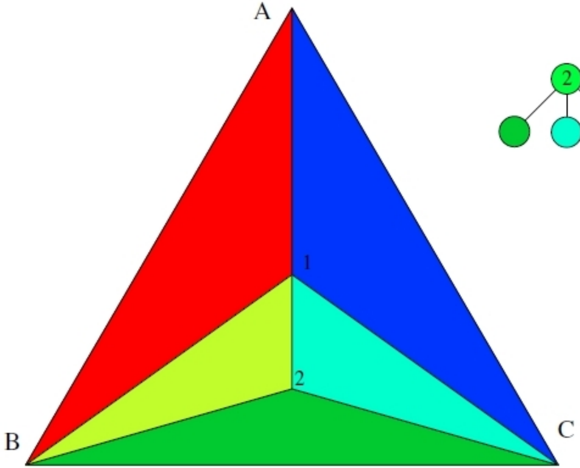


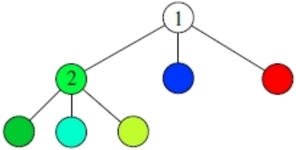
Random Apollonian networks



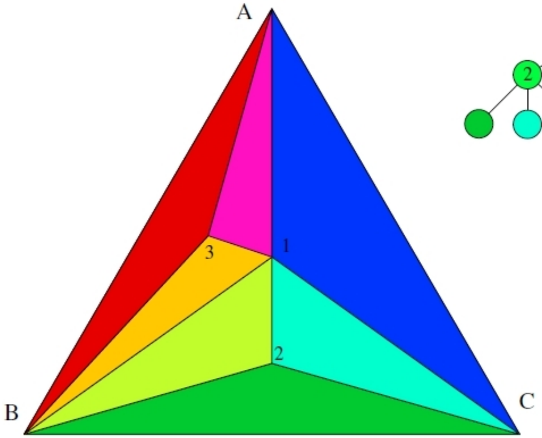


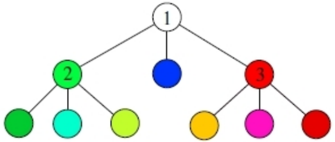
Random Apollonian networks





Random Apollonian networks





Small-world models

Duncan Watts

Steven Strogatz

construct a network with

- small diameter
- positive density of triangles

Small-world models

why should we want to construct a network with

- small diameter,
- positive density of triangles?

$$L(G) = \sum_{\text{pairs} u,v} \frac{d(u,v)}{\binom{n}{2}}, C(G) = \frac{1}{n}\sum_i C_i.$$

Graph	$\sim |V|$	$2|E|/|V|$	L_{actual}	L_{random}	C_{actual}	C_{random}
Film actors	225K	61	3.65	2.99	0.79	0.00027
Power grid	5K	2.67	18.7	12.4	0.08	0.005
C. elegans	0.3K	14	2.65	2.25	0.28	0.05

Small-world models

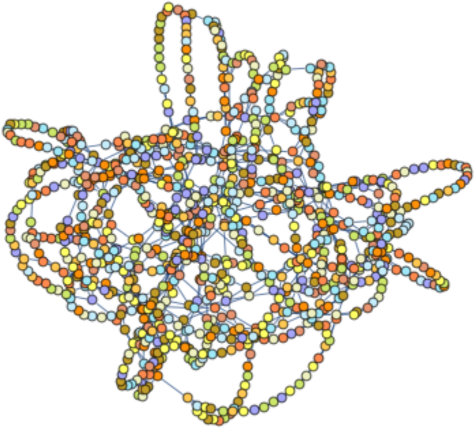
model

- let G be the r-th power of the cycle on n vertices

- notice that $\mathrm{diam}(G) = \frac{n}{2r}$ and $C(G) = \frac{3(r-1)}{2(2r-1)}$

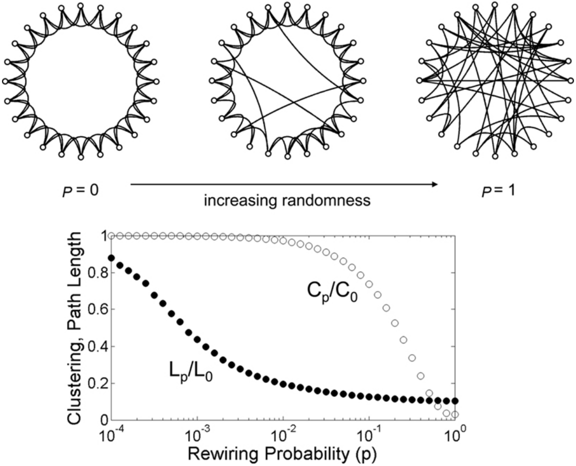
- let $G(p)$ be the graph obtained from G by deleting independently each edge with probability and then adding the same number of edges back at random

Small-world models



Watts-Strogatz on 1 000 vertices with rewiring probability $p = 0.05$

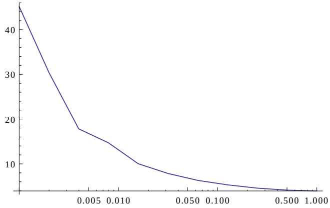
Small-world models



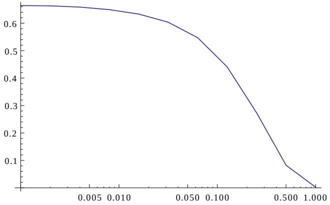
rewiring probability, p

even for a small value of p, $L(G(p))$ drops to $O(\log n)$, which $C(G(p)) \approx \frac{3}{4}$

Small-world models



average distance



clustering coefficient

Watts-Strogatz graph on 4 000 vertices, starting from a 10-regular graph

- intuition: if you add a little bit of randomness to a structured graph, you get the small world effect
- related work: see [Bollobás and Chung, 1988]

Navigation in a small world

Jon Kleinberg

how to find short paths using only local information?

- we will use a simple directed model [Kleinberg, 2000].
- a local algorithm
  - can remember the source, the destination and its current location
  - can query the graph to find the long-distance edge at the current location.

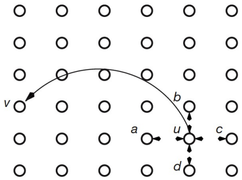
Navigation in a small world

$d(u, v)$: shortest path distance using only original grid edges

directed graph model, parameter r :

- each vertex is connected to its four adjacent vertices
- for each vertex v we add an extra link (v, u) where u is chosen with probability proportional to $d(v, u)^{-r}$

notice: compared to the Watts-Strogatz model the long range edges are added in a biased way



(source [Kleinberg, 2000])

Navigation in a small world

- $r = 0$: random edges, independent of distance
- as r increases the length of the long distance edges decreases in expectation

results

1. $r < 2$: the end points of the long distance edges tend to be uniformly distributed over the vertices of the grid
   - is unlikely on a short path to encounter a long distance edge whose end point is close to the destination
   - no local algorithm can find them
2. $r = 2$: there are short paths
   - a short path can be found be the simple algorithm that always selects the edge that takes closest to the destination
2. $r > 2$: there are no short paths, with high probability

Copying model

[Kumar et al., 2000] analyze the copying model of [Kleinberg et al., 1999b].

- $\alpha \in (0,1)$: copy factor
- d constant out degree.

evolving copying model, time $t+1$

- create a new vertex $t+1$
- choose a prototype vertex $u \in V_t$ uniformly at random
- the i-th out-link of $t+1$ is chosen as follows:

with probability α we select $x \in V_{t-1}$ uniformly at random, and with the remaining probability it copies the i-th out-lin of u

Copying model

in-degrees follow power-law distribution [Kumar et al., 2000]

Theorem

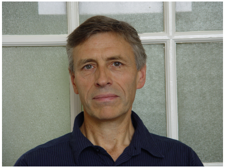
for $r > 0$ the limit $P_r = \lim_{t\to+\infty} \frac{N_t(r)}{t}$ exists and satisfies

$$P_r = \Theta(r^{-\frac{2-\alpha}{1-\alpha}}).$$

explains the large number of bipartite cliques in the web graph

static models with power-law degree distributions do not account for this phenomenon!

Cooper-Frieze model



Colin Cooper

Alan Frieze

Cooper and Frieze [Cooper and Frieze, 2003] introduce a general model

1. many parameters
2. generalizes preferential attachment, generalized preferential attachment and copying models
3. whose attachment rule is a mixture of preferential and uniform

Cooper-Frieze model

findings

1. we can obtain densification and shrinking diameters
 - add edges among existing vertices
2. power law in expectation and strong concentration under mild assumptions.
3. novel techniques for concentration

   martingales + Laplace

Kronecker graphs

reminder: Kronecker product

$A = [a_{ij}]$ an $m \times n$ matrix

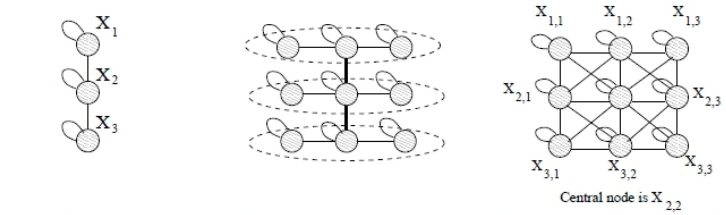
$B = [b_{ij}]$ a $p \times q$ matrix

then, $A \otimes B$ is the $mp \times nq$ matrix

$$\begin{pmatrix} a_{11}B & .. & a_{1n}B \\ .. & .. & .. \\ a_{m1}B & .. & a_{mn}B \end{pmatrix}$$

[Leskovec et al., 2010] propose a model based on the Kronecker product, generalizing RMAT [Chakrabarti et al., 2004].

Kronecker graphs



(a) Graph K_1 (b) Intermediate stage (c) Graph $K_2 = K_1 \otimes K_1$

1	1	0
1	1	1
0	1	1

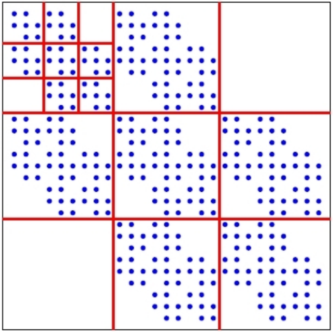
(d) Adjacency matrix of K_1

K_1	K_1	0
K_1	K_1	K_1
0	K_1	K_1

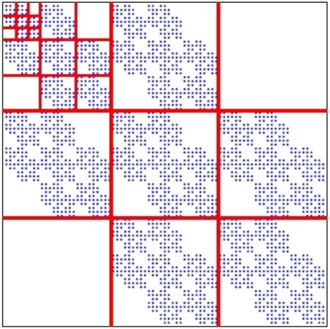
(e) Adjacency matrix of $K_2 = K_1 \otimes K_1$

source [Leskovec et al., 2010]

Kronecker graphs



(a) K_3 adjacency matrix (27×27)



(b) K_4 adjacency matrix (81×81)

source [Leskovec et al., 2010]

Kronecker graphs

a stochastic Kronecker graph is defined by two parameters

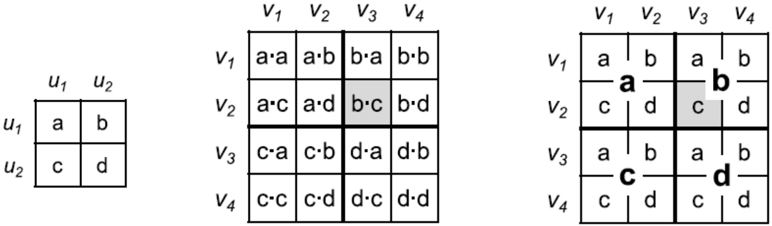
- an integer k
- the seed/initiator matrix θ

$$\begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

- we obtain a graph with $n = 2^k$ vertices by taking repeatedly Kronecker products
- let $A_{k,\theta} = \underbrace{\theta \otimes \ldots \otimes \theta}_{l \text{ times}}$ be the resulting matrix
- adjacency matrix $\bar{A}_{k,\theta}$ obtained by a randomized rounding
- typically 2×2 seed matrices are used;

  however, one can use other seed matrices

Kronecker graphs



in practice we never need to compute A, but we can actually do a sampling based on the hierarchical properties of Kronecker products.

Kronecker graphs

consider $G(V, E)$ such that $|V| = n = 2^k$.

- Erdős-Rényi

$$\begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

- hierarchical community structure

$$\begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}$$

- More known structures obtained by other seed matrices.

Kronecker graphs

- power-law degree distributions [Leskovec et al., 2010]
- power-law eigenvalue distribution [Leskovec et al., 2010]
- small diameter [Leskovec et al., 2010]
- densification power law [Leskovec et al., 2010]
- shrinking diameter [Leskovec et al., 2010]
- triangles [Tsourakakis, 2008]
- connectivity [Mahdian and Xu, 2007]
- giant components [Mahdian and Xu, 2007]
- diameter [Mahdian and Xu, 2007]
- searchability [Mahdian and Xu, 2007]

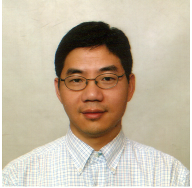
Kronecker graphs

how do we find a seed matrix θ such that $A_G \approx \underbrace{\theta \otimes \ldots \otimes \theta}_{k \text{ times}}$?

- maximum-likelihood estimation: $argmax_\theta \mathbf{Pr}[G|\theta]$
  - hard since exact computation requires $O(n!n^2)$ time, but
  - Metropolis sampling and approximations allow $O(m)$ time good approximations [Leskovec and Faloutsos, 2007]
- moment based estimation: express the expected number of certain subgraphs (e.g., edges, triangles, triples) as a function of a, b, c and solve a system of equations [Gleich and Owen, 2012]

Chung-Lu model

Fan Chung Graham



Linyuan Lu

- model is specified by $w = (w_1, \ldots, w_n)$ representing expected degree sequence
- certices i, j are connected with probability

$$p_{ij} = \frac{w_i w_j}{\sum_{k=1}^{n} w_k} = \rho w_i w_j.$$

- to have a proper probability distribution $w_{\max}^2 \leq \rho$
- can obtain an Erdős-Rényi random graph by setting

$$w = (pn, \ldots, pn)$$

Chung-Lu model

how to set the weights to get power law exponent β?

- the probability of having degree k in power law

$$\mathbf{Pr}\left[\deg(v) = k\right] = \frac{k^{-\beta}}{\zeta(\beta)}$$

- hence, for $\beta > 1$

$$\mathbf{Pr}\left[\deg(v) \geq k\right] = \sum_{l \geq k}^{+\infty} \frac{k^{-\beta}}{\zeta(\beta)} = \frac{1}{\zeta(\beta)(\beta - 1)k^{\beta - 1}}$$

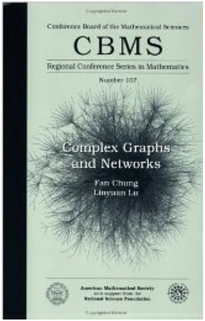
- assuming weights are decreasing and setting
  $w_i = k,\ i/n = \mathbf{Pr}\left[\deg(v) \geq k\right]$

$$w_i = \left(\frac{i}{\zeta(\beta)(\beta - 1)i}\right)^{-\frac{1}{\beta - 1}}$$

Chung-Lu model

rigorous results on:

- degree sequence
- giant component
- average distance and the diameter
- eigenvalues of the adjacency and the Laplacian matrix
- ...

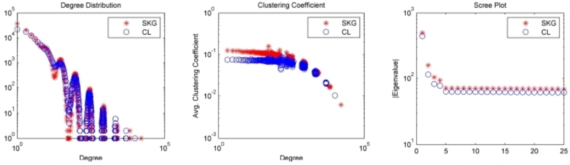


Complex graphs and networks, AMS

Kronecker vs. Chung-Lu

"the SKG model is close enough to its associated CL model that most users of SKG could just as well use the CL model for generating graphs."

[Pinar et al., 2011]



(a) Degree distribution (b) Clustering coefficients (c) Eigenvalues of adjacency matrices

Comparison of the graph properties of SKG and an equivalent CL.

Forest-fire model

J. Leskovec

J. Kleinberg

C. Faloutsos

[Leskovec et al., 2007] propose the forest fire model that is able to re-produce at a qualitative scale most of the established properties of real-world networks

Forest-fire model

basic version of the model

1. p : forward burning probability
2. r : backward burning ratio

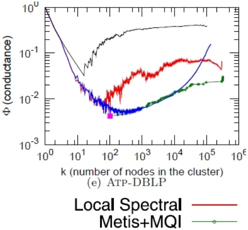
- initially, we have a single vertex
- at time t a new vertex v arrives to G_t
  - node v picks an *ambassador/seed* node u uniformly at random link to u
  - two numbers x, y are sampled from two geometric distributions with parameters $\frac{p}{1-p}$ and $\frac{rp}{1-rp}$ respectively
  - then, v chooses x out-links and y in-links of u which are incident to unvisited vertices
  - let $u_1, \ldots, u_{x+y}$ be these chosen endpoints
  - mark $u_1, \ldots, u_{x+y}$ as visited and apply the previous step recursively to each of them

Forest-fire model

the forest-fire model is able to explain

- heavy tailed in-degrees and out-degrees
- densification power law
- shrinking diameter
- ...
- deep cuts at small size scales and the absence of deep cuts at large size scales

reminder



NCP of a DBLP graph (source [Leskovec et al., 2009]).

Applications of random graphs

Influence of search engines on preferential attachment

Junghoo Cho

search-engine bias project

- in early days, search engines merely observed and exploited the web graph for ranking
- nowadays, they are unquestionably influencing the evolution of the web graph
- how?

Influence of search engines on preferential attachment

- "virtuous circle of limelight"

  a search engine ranks a page highly

  $\rightarrow$ web page owners find this page more often and link to it

  $\rightarrow$ raises its popularity

  and so on...

- main finding

  [Cho and Roy, 2004] estimate that the time taken for a page to reach prominence can be delayed by a factor of over 60 if a search engine diverts clicks to popular pages

- random graphs used to obtain insights into this phenomenon [Chakrabarti et al., 2005]

Influence of search engines on preferential attachment

Chakrabarti, Frieze and Vera [Chakrabarti et al., 2005] introduce a model with three parameters:

- p: a probability
- N: maximum number of celebrity nodes listed by the search engine
- m: edge parameter

notation:

- sequence of graphs $\{G_t\}_{t=1}^{+\infty}$. G_t will have t vertices and mt edges.
- $D_t(U) = \sum_{x \in U} deg_t(x)$
- S_t the set of at most N vertices with largest degrees in G_t.
- $d_k(t)$ denotes the number of vertices of degree k at time t. in the set $V_t - S_t$.

Influence of search engines on preferential attachment

- time step 1: the process is initialized with graph G_1 which consists of an isolated vertex x_1 and m loops
- time step $t > 1$: we add a vertex x_t to G_{t-1}

- we then add m random edges (x_t, y_i), $i = 1, \ldots, m$ incident with x_t, where y_i are nodes in G_{t-1}
- for each i:
  - with probability p we choose $y_i \in S_{t-1}$
  - with probability $1 - p$ we choose $y_i \in V_{t-1}$

  in both case y_i is selected by preferential attachment, i.e.,

$$\mathbf{Pr}\left[y_i = x\right] = \frac{deg_{t-1}(x)}{\sum_{u \in U} deg_{t-1}(u)}$$

where $U = S_{t-1}$ or $U = V_{t-1}$

Influence of search engines on preferential attachment

Theorem

Let $m \geq \max\{15, \frac{2}{1-p}\}$ *and* $0 < p < 1$

- *Let* $S_t = \{s_1, \ldots s_N\}$ *in decreasing order of degree. Then* $\mathbb{E}[deg_t(s_i)] \sim \alpha_i t$ *for every* $i \leq N$ *for some constant* $\alpha_i > 0$
- *There is an absolute constant* A_1 *such that for every* $k \geq m$

$$\mathbb{E}[d_k(t)] = \frac{A_1 n}{k^{1+\frac{2}{1-p}}} + \textit{second order terms}$$

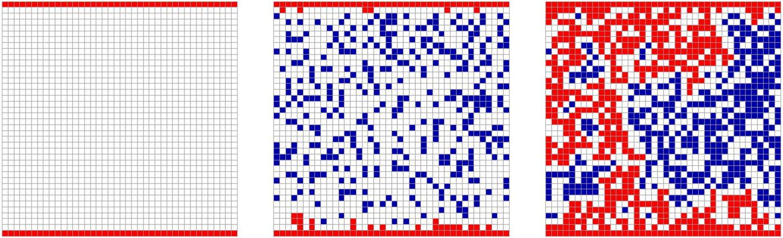
Influence of search engines on preferential attachment

the theorem and its proof verify our intuition

- the celebrity lit gets fixed quickly
- each celebrity page captures a constant fraction of all edges ever generated in the graph
- the non-celebrity vertices obey a power law which is steeper

Robustness and vulnerability

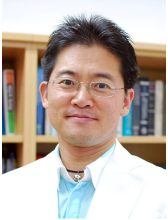
- intuitively, a complex network is *robust* if it keeps its basic functionality under the failure of some of its components.
- distinguish between random failure and intentional attacks
- related to percolation



percolation

Robustness and vulnerability

R. Albert



H. Jeong

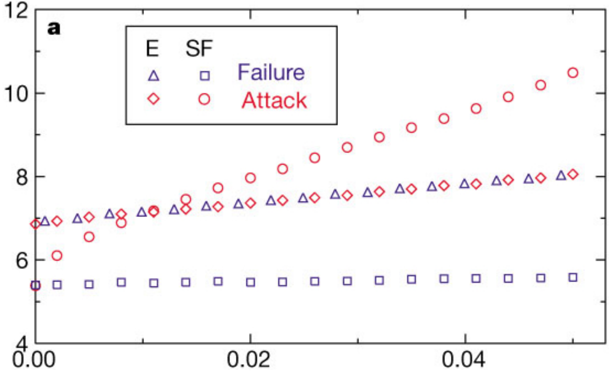
L. Barabási

[Albert et al., 2000] provide simulations indicating that scale free networks are robust to random failures

10 second sound bite science

> *The Internet is robust yet fragile. 95% of the links can be removed and the graph will stay connected. However, targeted removal of 2.3% of the hubs would disconnect the Internet.*

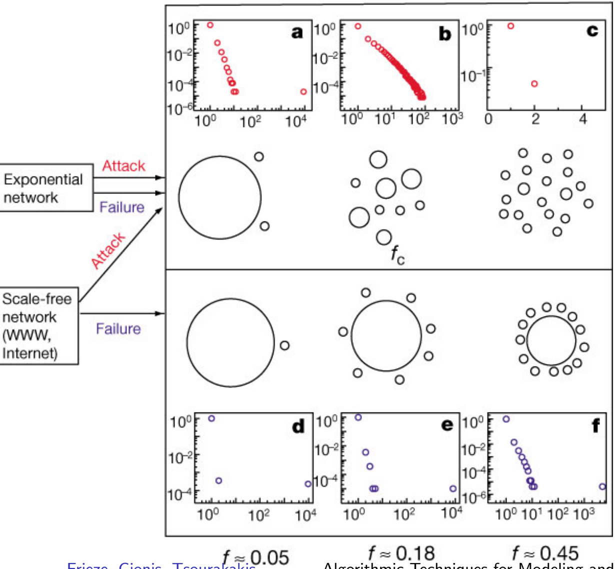
Robustness and vulnerability



$n = 10\,000$ vertices $m = 20\,000$ links

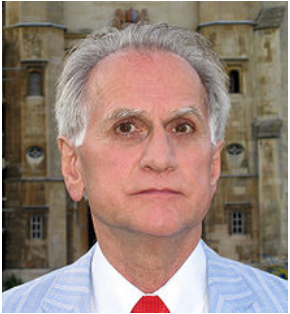
- diameter of an Erdős-Rényi and a scale-free network as a function of the fraction f of vertices deleted
- the power-law distribution implies that under random sampling, vertices with small degree are selected selected with much higher probability

Robustness and vulnerability



- the cluster size distribution for various values of f for an Erdős-Rényi graph and a scale-free network under random and malicious failures (source [Albert et al., 2000])
- Intuition: scale-free graphs are inhomogeneous which implies both better performance under random failures and reduced attack survivability

Robustness and vulnerability



Bélla Bollobás

Oliver Riordan

[Bollobás and Riordan, 2004] studied the robustness and vulnerability of a scale-free graph, using specifically the Barabási-Albert model

Robustness and vulnerability

- when vertices of $G_m^{(n)}$ are deleted independently with probability $1-p$, there is always a giant component!
- no critical p
- however the size of the giant component depends on p

Theorem

Let $m \geq 2$, $0 < p < 1$ be fixed and let G_p be obtained from $G_m^{(n)}$ by deleting vertices independently with probability $1-p$
Then as $n \to +\infty$ **whp** *the largest component of G_p has order $((c(p,m)+o(1))n$*
Furthermore, as $p \to 0$ with m fixed, $c(p,m) = \exp\left(\frac{1}{O(p)}\right)$

Robustness and vulnerability

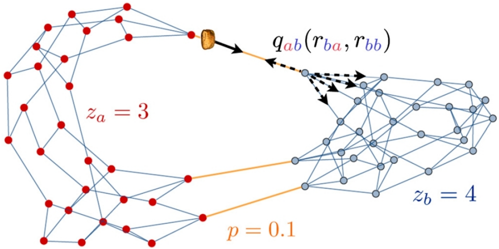
- when $G_m^{(n)}$ is deliberately attacked, finding the "best" attack is hard
- Bollobás and Riordan consider the natural attack of deleting the earliest vertices up to some cutoff cn

Theorem

Let G_c be obtained by $G_m^{(n)}$ by deleting all vertices with index less than cn, where $0 < c < 1$ is a constant.
Let $c_m = \frac{m-1}{m+1}$.
If $c < c_m$ then **whp** *G_c has a component with $\Theta(n)$ vertices.*
If $c > c_m$ then **whp** *G_c has no such component.*

More applications

- study of interdependent networks [Brummitt et al., 2012]



a random three- and four-regular graph connected by Bernoulli distributed coupling with interconnectivity parameter $p = 0.1$

More applications

Itai Ashlagi

Alvin Roth

compatibility graph : each vertex is a donor-patient pair and each edge between two vertices denotes compatibility for kidney exchange.

- model kidney exchange with many patient-donor pairs as a random compatibility graph

More applications

Motivation

- we wish to messages in a cellular network G, between any two vertices in a pipeline
- we require that each link on the route between the vertices (namely, each edge on the path) is assigned a distinct channel (e.g., a distinct frequency)

an edge colored graph G is rainbow edge connected if any two vertices are connected by a path whose edges have distinct colors

goal: Find the minimum number of colors needed to rainbow color the edges of G

[Frieze and Tsourakakis, 2012] study rainbow connectivity in sparse random graphs

More applications

- [Cooper and Frieze, 2004] studied the performance of crawlers in random evolving scale-free graphs
- [Valiant, 2005] uses random graphs to model memorization and association functionalities of the brain
- simulations (epidemics, performance of algorithms etc.)
- graph anonymization [Leskovec et al., 2005a]
- allow to argue about the structure of real-world networks for instance, given a random graph with a fixed degree distribution, what do we expect for the spectrum, subgraphs etc?
- give rise to objectives by using them as null models (modularity)
- and many more ..

Conclusions (random graphs)

- just scratched the tip of the iceberg
  - random geometric graphs [Penrose, 2003]
    - hyperbolic geometry [Gugelmann et al., 2012]
    - line of sight networks [Frieze et al., 2009]
  - protean graphs [Łuczak and Prałat, 2006]
  - geometric preferential attachment [Flaxman et al., 2006]
  - affiliation networks [Lattanzi and Sivakumar, 2009]
  - many other interesting stochastic models ..
  - optimization based models for topology
    - Doyle et al. [Doyle and Carlson, 2000, Li et al., 2005]
    - heuristically optimized trade-offs [Fabrikant et al., 2002]
  - a different line of research, networks as biproduct of strategy selection [Dutta and Jackson, 2003], [Fabrikant et al., 2003], [Borgs et al., 2011]

Conclusions (random graphs)

- there is no single model that matches all established existing properties
  - the forest-fire model appears to match most, but we do not understand well this model
- many types of networks (social networks, information networks, technological networks), develop specialized models

Outline

- introduction and graphs and networks
- random graphs as models of real-world networks
  - properties of real-world networks
  - Erdős-Rényi graphs
  - models of real-world networks
  - applications of random graphs
- algorithm design for large-scale networks
  - graph partitioning and community detection
  - dense subgraphs

Graph partitioning and community detection

Graph partitioning and community detection

- knowledge discovery
  - partition the web into sets of related pages (web graph)
  - find groups of scientists who collaborate with each other (co-authorship graph)
  - find groups of related queries submitted in a search engine (query graph)

- performance
  - partition the nodes of a large social network into different machines so that, to a large extent, friends are in the same machine (social networks)

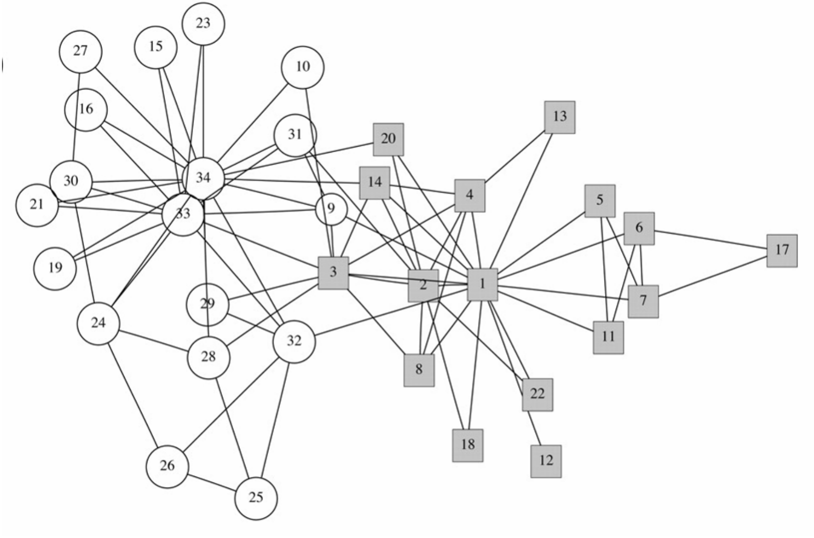
Graph partitioning — high-level problem definition

- graph $G = (V, E, w)$
- edge (u, v) denotes affinity between u and v
- weight of edge $w(u, v)$ can be used to quantify the degree of affinity

- we want to partition the vertices in clusters so that:
  - vertices within clusters are well connected, and
  - vertices across clusters are sparsely connected

- typical graph-partitioning problems are **NP**-hard

Graph partitioning



(Zachary's karate-club network, figure from [Newman and Girvan, 2004])

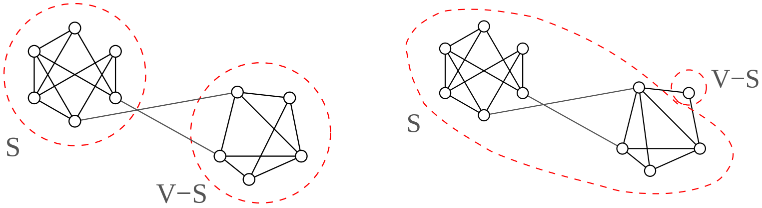
Objective functions: (1) min cut

- the minimum number of edges cut by a two-component partitioning
- cut:

$$E(S, T) = \{(u, v) \in E \mid u \in S \text{ and } v \in T\}$$

- min cut:

$$c(G) = \min_{S \subseteq V} |E(S, V \setminus S)|$$



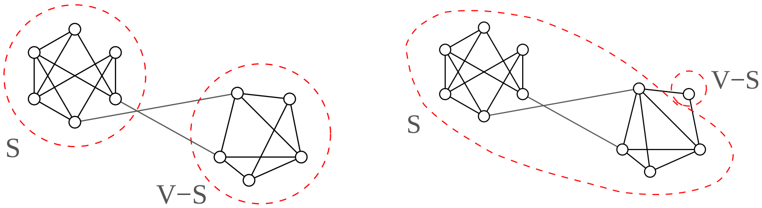
Objective functions: (2) graph expansion

- normalize the cut by the size of the smallest component
- ratio cut:

$$\alpha(G, S) = \frac{|E(S, V \setminus S)|}{\min\{|S|, |V \setminus S|\}}$$

- graph expansion:

$$\alpha(G) = \min_{S} \frac{|E(S, V \setminus S)|}{\min\{|S|, |V \setminus S|\}}$$



Objective functions: (3) conductance

- normalize by volume

$$\mathrm{vol}(S) = \sum_{i \in S} d_i, \text{ for } S \subseteq V \quad (\text{so, } \mathrm{vol}(V) = 2m)$$

- set conductance:

$$\phi(G, S) = \frac{|E(S, V \setminus S)|}{\min\{vol(S), vol(V \setminus S)\}}$$

- graph conductance:

$$\phi(G) = \min_{S \subseteq V} \frac{|E(S, V \setminus S)|}{\min\{vol(S), vol(V \setminus S)\}}$$

Background: linear algebra and eigenvalues

- consider a real $n \times n$ matrix A
- $(\lambda, \mathbf{u})$ an eigenvalue–eigenvector pair if $A\,\mathbf{u} = \lambda\,\mathbf{u}$
- a symmetric real matrix has real eigenvalues
- the set of eigenvalues of a matrix is called the spectrum of the matrix

$$\sigma(A) = \{\lambda_1, \ldots, \lambda_n\}$$

  index them so that $\lambda_1 \leq \ldots \leq \lambda_n$
- A is positive semi-definite if $\mathbf{x}^T A\,\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$
- a symmetric positive semi-definite real matrix has non negative eigenvalues

Background: linear algebra and eigenvalues

- for a symmetric matrix, the eigenvectors that corespond to different eigenvalues are orthogonal ($\lambda_i \neq \lambda_j$ implies $\mathbf{u}_i^T \mathbf{u}_j = 0$)
- the range of A is the linear space spanned by the columns of A

$$\mathrm{range(A)} = \{\mathbf{x} \in \mathbb{R}^n \mid A\,\mathbf{y} = \mathbf{x},\ \text{for some } \mathbf{y} \in \mathbb{R}^n\}$$

- for a real and symmetric matrix A, the range of A is spanned by the eigenvectors with non-zero eigenvalues
- for a real and symmetric matrix A, with eigenvalues $\lambda_1 \leq \ldots \leq \lambda_n$ and corresponding eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$

$$A = \sum_{i=1}^{n} \lambda_i\, \mathbf{u}_i\, \mathbf{u}_i^T$$

Background: min–max characterization of eigenvalues

- for a real and symmetric matrix A with eigenvalues $\lambda_1 \leq \ldots \leq \lambda_n$

$$\lambda_n = \max_{\mathbf{v}^T\mathbf{v}=1} \mathbf{v}^T A \mathbf{v}$$

$$\lambda_1 = \min_{\mathbf{v}^T\mathbf{v}=1} \mathbf{v}^T A \mathbf{v}$$

$$\lambda_2 = \min_{\substack{\mathbf{v}^T\mathbf{v}=1 \\ \mathbf{v}^T\mathbf{u}_1=0}} \mathbf{v}^T A \mathbf{v}$$

and in general

$$\lambda_k = \min_{\substack{\mathbf{v}^T\mathbf{v}=1 \\ \mathbf{v}^T\mathbf{u}_i=0, i=1\ldots k-1}} \mathbf{v}^T A \mathbf{v}$$

Spectral analysis of graphs

- $G = (V, E)$ an undirected graph
- A the adjacency matrix of G:
- define the *Laplacian matrix* of A as

$$L = D - A \qquad \text{or} \qquad L_{ij} = \begin{cases} d_i & \text{if } i = j \\ -1 & \text{if } (i,j) \in E, i \neq j \\ 0 & \text{if } (i,j) \notin E, i \neq j \end{cases}$$

- where $D = \mathrm{diag}(d_1, \ldots, d_n)$, a *diagonal* matrix
- L is symmetric positive semi-definite
- The smallest eigenvalue of L is $\lambda_1 = 0$, with eigenvector $\mathbf{u}_1 = (1, 1, \ldots, 1)^T$

Spectral analysis of graphs

- consider the second smallest eigenvector λ_2 of L

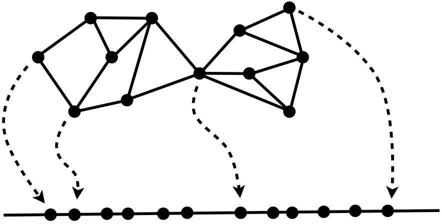
$$\lambda_2 = \min_{\substack{||\mathbf{x}||=1 \\ \mathbf{x}^T \mathbf{u}_1 = 0}} \mathbf{x}^T L \, \mathbf{x} = \min_{\sum x_i = 0} \frac{\sum_{(i,j)\in E} (x_i - x_j)^2}{\sum_i x_i^2}$$

- the corresponding eigenvector $\mathbf{u}_2$ is called Fielder vector
- ordering according to the values of $\mathbf{u}_2$ will group similar (connected) vertices together
- one-dimensional embedding that preserves the graph structure
- physical interpretation: minimize elastic potential energy if graph is materialized with springs at its edges

Spectral analysis of graphs

$$\lambda_2 = \min_{\substack{||\mathbf{x}||=1 \\ \mathbf{x}^T \mathbf{u}_1 = 0}} \mathbf{x}^T L \, \mathbf{x} = \min_{\sum x_i = 0} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_i x_i^2}$$

- ordering according to the values of $\mathbf{u}_2$ will group similar (connected) vertices together
- one-dimensional embedding that preserves the graph structure



Random walks

- consider random walk on the graph G by following edges
- from vertex i move to vertex j with prob. $1/d_i$ if $(i,j) \in E$
- $\mathbf{p}_i^{(t)}$ probability of being at vertex i at time t
- process is described by equation $\mathbf{p}^{(t+1)} = \mathbf{p}^{(t)} P$, where $P = D^{-1} A$ is row-stochastic
- process converges to stationary distribution $\pi = \pi P$ (under certain irreducibility conditions)
- for undirected and connected graphs

$$\pi_i = \frac{d_i}{2m} \quad \text{(stationary distribution} \sim \text{degree)}$$

Random walks — useful concepts

- hitting time $H(i,j)$: expected number of steps before visiting vertex j, starting from i
- commute time $\kappa(i,j)$: expected number of steps before visiting j and i again, starting at i

$$\kappa(i,j) = H(i,j) + H(j,i)$$

- cover time R: expected number of steps to reach every node
- mixing time $\tau(\epsilon)$: a measure of how fast the random walk approaches its stationary distribution

$$\tau(\epsilon) = \min\{t \mid d(t) \leq \epsilon\}$$

where

$$d(t) = \max_i ||\mathbf{p}^t(i,\cdot) - \pi|| = \max_i \left\{ \sum_j |\mathbf{p}^t(i,j) - \pi_j| \right\}$$

Random walks — spectral analysis

- instead of $L = D - A$ consider normalized Laplacian $L' = I - D^{-1/2} A\, D^{-1/2}$

$$
\begin{aligned}
L' \mathbf{u} &= \lambda \mathbf{u} \\
(I - D^{-1/2} A\, D^{-1/2}) \mathbf{u} &= \lambda \mathbf{u} \\
(D - A) \mathbf{u} &= \lambda D \mathbf{u} \\
D \mathbf{u} &= A \mathbf{u} + \lambda D \mathbf{u} \\
(1 - \lambda) \mathbf{u} &= D^{-1} A \mathbf{u} \\
\mu \mathbf{u} &= P \mathbf{u}
\end{aligned}
$$

- $(\lambda, \mathbf{u})$ is an eigenvalue–eigenvector pair for L' if and only if $(1 - \lambda, \mathbf{u})$ is an eigenvalue–eigenvector pair for P
- the eigenvector with smallest eigenvalue for L' is the eigenvector with largest eigenvalue for P

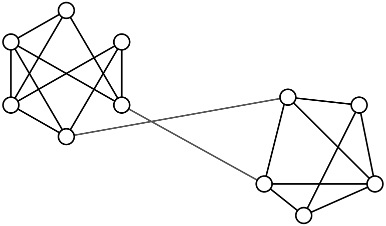
Random walks — spectral analysis

- stochastic matrix P, describing the random walk
- eigenvalues: $-1 < \mu_n \leq \ldots \leq \mu_2 < \mu_1 = 1$
- spectral gap: $\gamma_* = 1 - \mu_2$
- relaxation time: $\tau_* = \frac{1}{\gamma_*}$
- theorem: for an aperiodic, irreducible, and reversible random walk, and any ϵ

$$(\tau_* - 1)\log\left(\frac{1}{2\epsilon}\right) \leq \tau(\epsilon) \leq \tau_* \log\left(\frac{1}{2\epsilon\sqrt{\pi_{\min}}}\right)$$

Random walks — spectral analysis

- intuition: fast mixing related to graph being an expander
- large mixing time $\Rightarrow$ bottlenecks $\Rightarrow$ clusters



- large spectral gap $\Rightarrow$ no clusters

Spectral analysis and clustering measures

- clustered structure of G captured by
  min cut $c(G)$
  expansion $\alpha(G)$
  conductance $\phi(G)$
- no surprise those clustering measures are related to spectral gap

Cheeger inequality

- eigenvalues of the stochastic matrix P, describing the random walk: $-1 < \mu_n \leq \ldots \leq \mu_2 < \mu_1 = 1$
- eigenvalues of normalized Laplacian: $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$
- spectral gap: $\gamma_* = 1 - \mu_2 = \lambda_2$
- Cheeger inequality:

$$\frac{\phi(G)^2}{2} \leq \gamma_* = \lambda_2 \leq 2\,\phi(G)$$

- [reminder] graph conductance:

$$\phi(G) = \min_{S \subseteq V} \frac{|E(S, V \setminus S)|}{\min\{vol(S), vol(V \setminus S)\}}$$

Spectral analysis of graphs

- consider the second smallest eigenvector λ_2 of L

$$\lambda_2 = \min_{\substack{||\mathbf{x}||=1 \\ \mathbf{x}^T \mathbf{u}_1 = 0}} \mathbf{x}^T L \, \mathbf{x} = \min_{\sum x_i = 0} \frac{\sum_{(i,j)\in E} (x_i - x_j)^2}{\sum_i x_i^2}$$

- ordering according to the values of $\mathbf{u}_2$ will group similar (connected) vertices together
- one-dimensional embedding that preserves the graph structure
- λ_2 corresponds to spectral gap
- the smaller λ_2 the better the clusters

Interesting special case

- the smaller λ_2 the better the clusters
- theorem: let L be the Laplacian of a graph $G = (V, E)$.
  $\lambda_2 > 0$ if and only if G is connected

  proof: if G disconnected then

  $$L = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix}$$

  consider also

  $$\lambda_2 = \min_{\substack{||\mathbf{x}||=1 \\ \mathbf{x}^T \mathbf{u}_1 = 0}} \mathbf{x}^T L \mathbf{x} = \min_{\sum x_i = 0} \frac{\sum_{(i,j)\in E} (x_i - x_j)^2}{\sum_i x_i^2}$$

  $\square$

Inside the proof of Cheeger's inequality

- $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$ (normalized Laplacian)
- Cheeger inequality

$$\frac{\phi(G)^2}{2} \leq \gamma_* = \lambda_2 \leq 2\,\phi(G)$$

[$\lambda_2 \leq 2\,\phi(G)$]

- $2\phi(G)$ can be written as an expression over $x_i \in \{0, 1\}$ indicating whether $i \in S$
- λ_2 can be written as the fractional relaxation of the previous expression

Inside the proof of Cheeger's inequality

- $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$ (normalized Laplacian)
- Cheeger inequality

$$\frac{\phi(G)^2}{2} \leq \gamma_* = \lambda_2 \leq 2\,\phi(G)$$

[$\phi(G) \leq \sqrt{2\,\lambda_2}$]

- constructive
- order graph vertices according to the eigenvector of λ_2
- form S by spliting vertices around their median
- show for that partitioning $\phi(S) \leq \sqrt{2\,\lambda_2}$

Basic spectral-partition algorithm

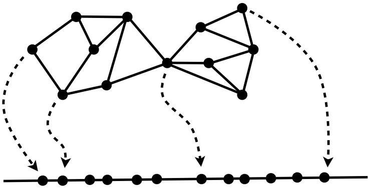
1. form normalized Laplacian $L' = I - D^{-1/2} A\, D^{-1/2}$
2. compute the eigenvector $\mathbf{u}_2$ that corresponds to λ_2
3. order vertices according their coefficient value on $\mathbf{u}_2$
4. consider only sweeping cuts: splits that respect the order
5. take the sweeping cut S that minimizes $\phi(S)$

theorem the basic spectral-partition algorithm finds a cut S such that $\phi(S) \leq 2\sqrt{\phi(G)}$

proof by Cheeger inequality $\quad \phi(S) \leq \sqrt{2 \cdot \lambda_2} \leq \sqrt{2 \cdot 2 \cdot \phi(G)}$

Spectral partitioning rules

1. conductance: find the partition that minimizes $\phi(G)$
2. bisection: split in two equal parts
3. sign: separate positive and negative values
4. gap: separate according to the largest gap



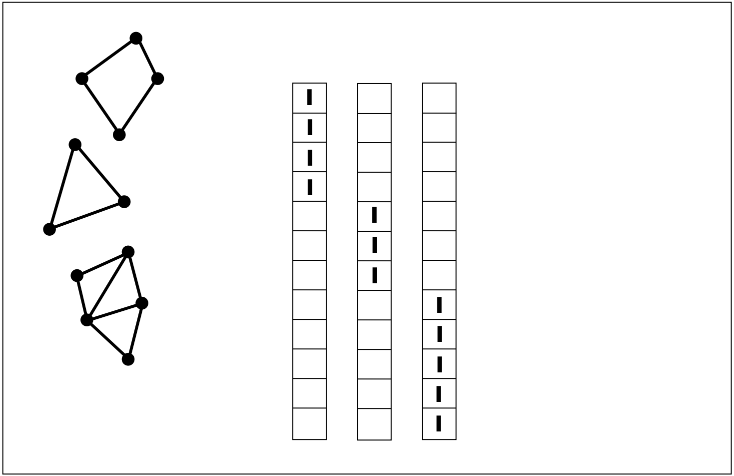
Other common spectral-partitioning algorithms

1. utilize more eigenvectors than just the Fielder vector
   use k eigenvectors
2. different versions of the Laplacian matrix

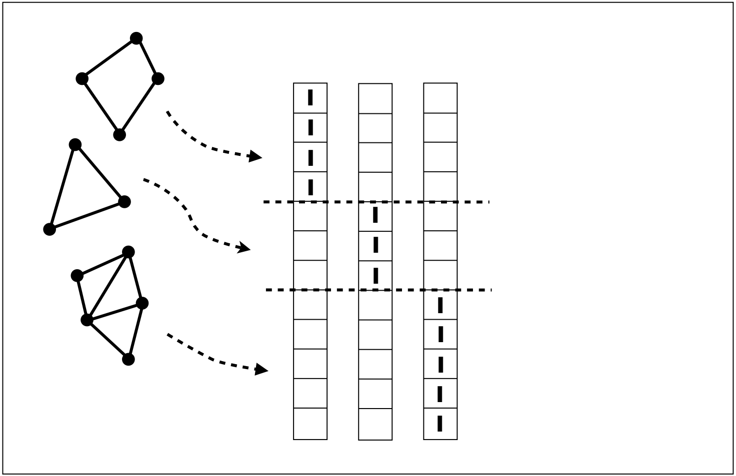
Using k eigenvectors

- ideal scenario: the graph consists of k disconnected components (perfect clusters)
- then: eigenvalue 0 of the Laplacian has multplicity k the eigenspace of eigenvalue 0 is spanned by indicator vectors of the graph components

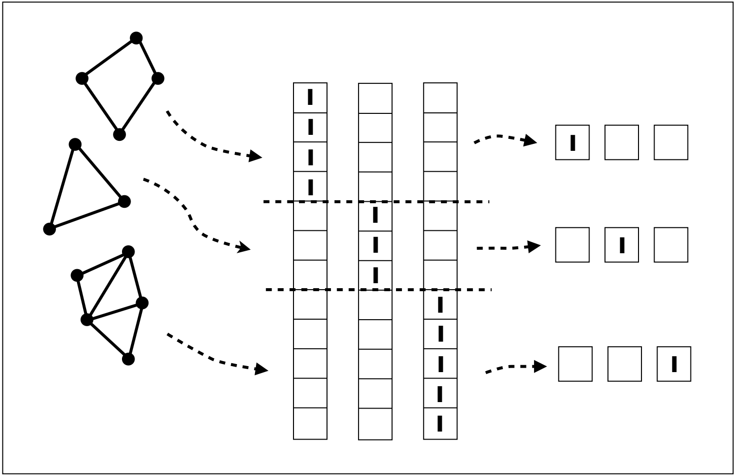
Using k eigenvectors



Using k eigenvectors



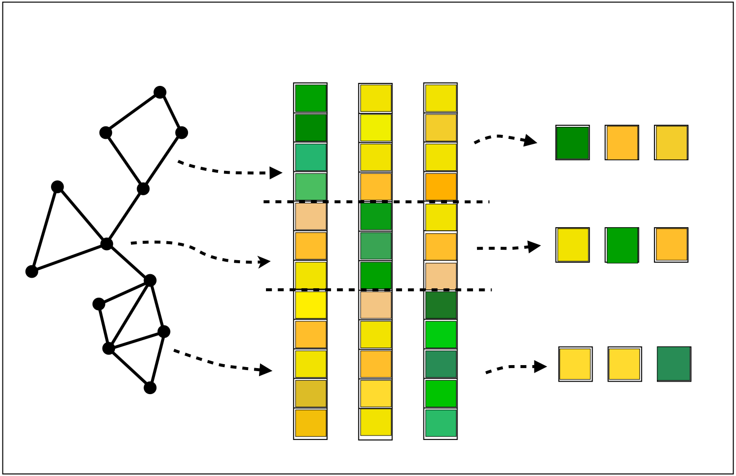
Using k eigenvectors



Using k eigenvectors

- robustness under perturbations: if the graph has less well-separated components the previous structure holds approximately
- clustering of Euclidean points can be used to separate the components

Using k eigenvectors



Laplacian matrices

- unormalized Laplacian: $L = D - A$

$$L_{ij} = \begin{cases} d_i & \text{if } i = j \\ -1 & \text{if } (i,j) \in E, i \neq j \\ 0 & \text{if } (i,j) \notin E, i \neq j \end{cases}$$

- normalized symmetric Laplacian: $L' = I - D^{-1/2} A\, D^{-1/2}$
- normalized "random-walk" Laplacian: $L_{\mathrm{rw}} = I - D^{-1} A$

All Laplacian matrices are related

- unormalized Laplacian: $\lambda_2 = \min_{\substack{||\mathbf{x}||=1 \\ \mathbf{x}^T\mathbf{u}_1=0}} \sum_{(i,j)\in E}(x_i - x_j)^2$
- normalized Laplacian:

$$\lambda_2 = \min_{\substack{||\mathbf{x}||=1 \\ \mathbf{x}^T\mathbf{u}_1=0}} \sum_{(i,j)\in E} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}}\right)^2$$

- $(\lambda, \mathbf{u})$ is an eigenvalue/vector of L_{rw} if and only if $(\lambda, D^{1/2}\,\mathbf{u})$ is an eigenvalue/vector of L'
- $(\lambda, \mathbf{u})$ is an eigenvalue/vector of L_{rw} if and only if $(\lambda, \mathbf{u})$ solve the generalized eigen-problem $L\,\mathbf{u} = \lambda\, D\,\mathbf{u}$

Algorithm 1: unormalized spectral clustering

input graph adjacency matrix A, number k

1. form diagonal matrix D
2. form unormalized Laplacian $L = D - A$
3. compute the first k eigenvectors $u_1, \ldots, u_k$ of L
4. form matrix $U \in \mathbb{R}^{n \times k}$ with columns $u_1, \ldots, u_k$
5. consider the i-th row of U as point $y_i \in \mathbb{R}^k, i = 1, \ldots, n$,
6. cluster the points $\{y_i\}_{i=1,\ldots,n}$ into clusters $C_1, \ldots, C_k$

   e.g., with k-means clustering

output clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$

Algorithm 2: normalized spectral clustering

[Shi and Malik, 2000]

input graph adjacency matrix A, number k

1. form diagonal matrix D
2. form unormalized Laplacian $L = D - A$
3. compute the first k eigenvectors $u_1, \ldots, u_k$ of the generalized eigen-problem $L\,\mathbf{u} = \lambda\,D\,\mathbf{u}$ (eigvctrs of L_{rw})
4. form matrix $U \in \mathbb{R}^{n \times k}$ with columns $u_1, \ldots, u_k$
5. consider the i-th row of U as point $y_i \in \mathbb{R}^k, i = 1, \ldots, n$,
6. cluster the points $\{y_i\}_{i=1,\ldots,n}$ into clusters $C_1, \ldots, C_k$

   e.g., with k-means clustering

output clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$

Algorithm 3: normalized spectral clustering

[Ng et al., 2001]

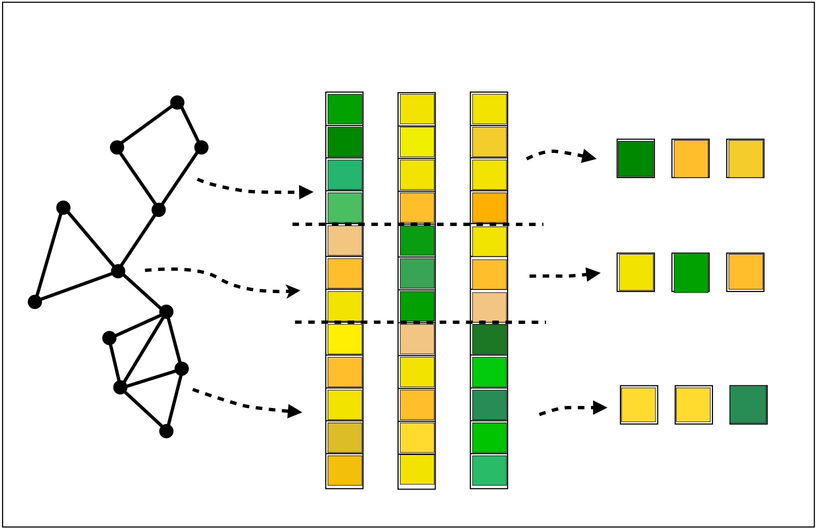
input graph adjacency matrix A, number k

1. form diagonal matrix D
2. form normalized Laplacian $L' = I - D^{-1/2} A \, D^{-1/2}$
3. compute the first k eigenvectors $u_1, \ldots, u_k$ of L'
4. form matrix $U \in \mathbb{R}^{n \times k}$ with columns $u_1, \ldots, u_k$
5. normalize U so that rows have norm 1
6. consider the i-th row of U as point $y_i \in \mathbb{R}^k, i = 1, \ldots, n$,
7. cluster the points $\{y_i\}_{i=1,\ldots,n}$ into clusters $C_1, \ldots, C_k$

   e.g., with k-means clustering

output clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$

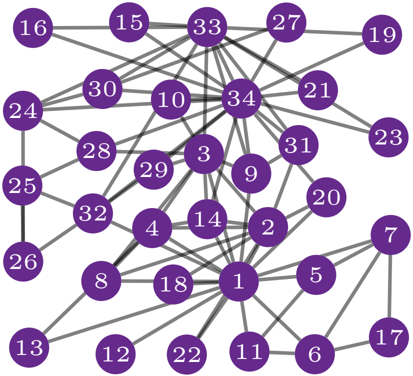
intuition of the spectral algorithms



Notes on the three spectral algorithms

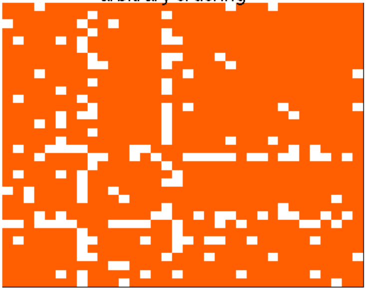
- quite similar except for using the three different Laplacians
- can be used to cluster any type of data, not just graphs form all-pairs similarity matrix and use as adjacency matrix
- computation of the first eigenvectors of sparse matrices can be done efficiently using the Lanczos method

Zachary's karate-club network

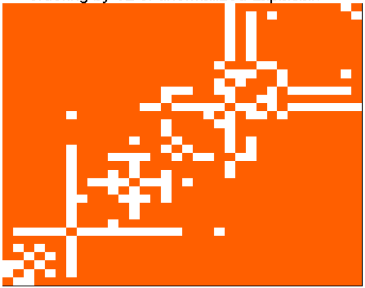


Zachary's karate-club network

arbitrary ordering

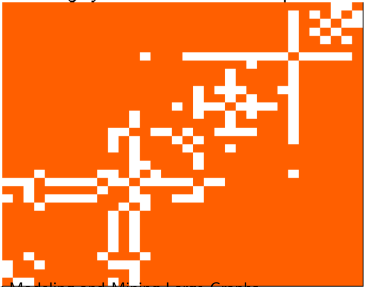


ordering by v2 of unormalized Laplacian

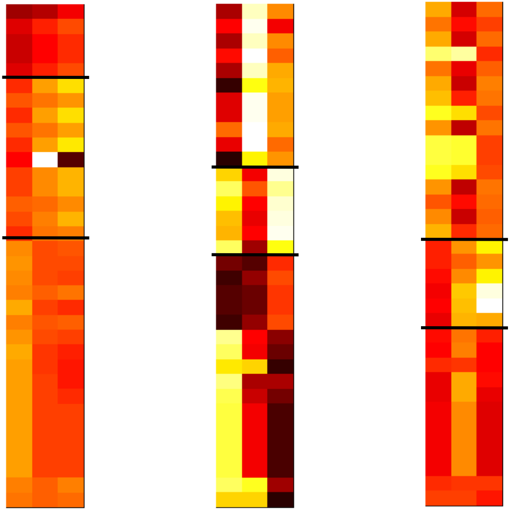


ordering by v2 of normalized sym. Laplacian

ordering by v2 of normalized RW Laplaciar



Zachary's karate-club network

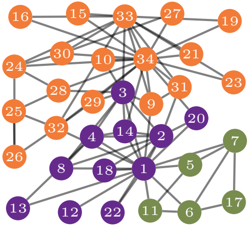


unormalized Laplacian

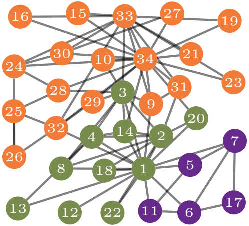
normalized symmetric Laplacian

normalized random walk Laplacian

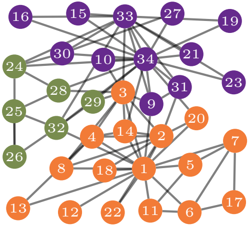
Zachary's karate-club network



unormalized
Laplacian



normalized
symmetric
Laplacian



normalized
random walk
Laplacian

Which Laplacian to use?

[von Luxburg, 2007]

- when graph vertices have about the same degree all Laplacians are about the same
- for skewed degree distributions normalized Laplacians tend to perform better, and L_{rw} is preferable
- normalized Laplacians are associated with conductance, which is preferable than ratio cut

  (conductance involves $\mathrm{vol}(S)$ rather than $|S|$ and captures better community structure)

Modularity

- cut ratio, graph expension, conductance useful to extract one component
- not clear how to extend to measure quality of graph partitions
- related question: what is the optimal number of partitions?
- *modularity measure* has been used to answer those questions
- [Newman and Girvan, 2004]
- originally developed to find the optimal number of partitions in hierarchical graph partitioning

Modularity

- intuition: compare actual subgraph density with expected subgraph density, if vertices were attached regardless of community structure

$$
\begin{aligned}
Q &= \frac{1}{2m}\sum_{ij}(A_{ij} - P_{ij})\delta(C_i, C_j) \\
&= \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{d_i d_j}{2m})\delta(C_i, C_j) \\
&= \sum_{c}\left[\frac{m_c}{2m} - \left(\frac{d_c}{2m}\right)^2\right]
\end{aligned}
$$

$P_{ij} = 2mp_ip_j = 2m(d_i/2m)(d_j/2m) = (d_id_j/2m)$
m_c: edges within cluster c
d_c: total degree of cluster c

Values of modularity

- 0 random structure; 1 strong community structure; [0.3..0.7]; typical good structure; can be negative, too
- Q measure is not monotone with k

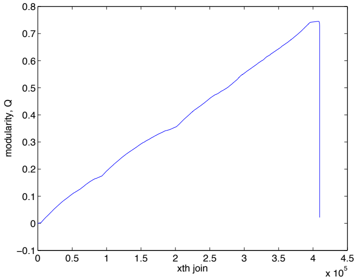


FIG. 1: The modularity Q over the course of the algorithm (the x axis shows the number of joins). Its maximum value is $Q = 0.745$, where the partition consists of 1684 communities.

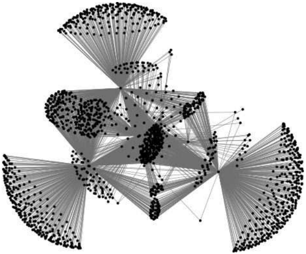


FIG. 2: A visualization of the community structure at maximum modularity. Note that the some major communities have a large number of "satellite" communities connected only to them (top, lower left, lower right). Also, some pairs of major communities have sets of smaller communities that act as "bridges" between them (e.g., between the lower left and lower right, near the center).

(figures from [Clauset et al., 2004])

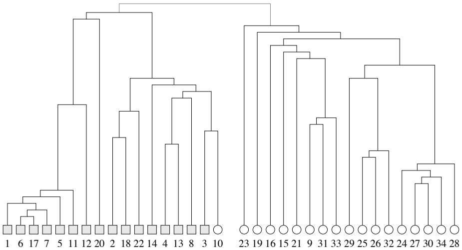
Optimizing modularity

- problem: find the partitioning that optimizes modularity
- **NP**-hard problem [Brandes et al., 2006]
- top-down approaches [Newman and Girvan, 2004]
- spectral approaches [Smyth and White, 2005]
- mathematical-programming [Agarwal and Kempe, 2008]

Top-down algorithms for optimizing modularity

[Newman and Girvan, 2004]

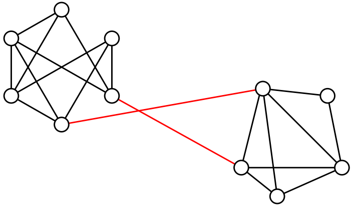
- a set of algorithms based on removing edges from the graph, one at a time
- the graph gets progressively disconnected, creating a hierarchy of communities



(figure from [Newman, 2004])

Top-down algorithms

- select edge to remove based on *"betweenness"*



three definitions

- shortest-path betweenness: number of shortest paths that the edge belongs to
- random-walk betweenness: expected number of paths for a random walk from u to v
- current-flow betweenness: resistance derived from considering the graph as an electric circuit

Top-down algorithms

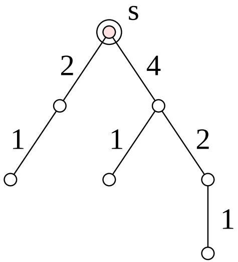
general scheme

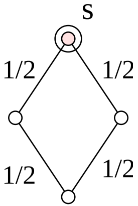
TopDown

1. Compute betweenness value of all edges
2. Remove the edge with the highest betweenness
3. Recompute betweenness value of all remaining edges
4. Repeat until no edges left

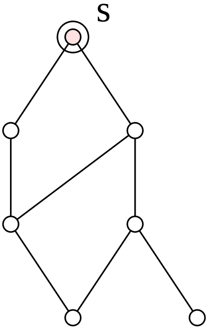
Shortest-path betweenness

- how to compute shortest-path betweenness?
- BFS from each vertex
- leads to $O(mn)$ for all edge betweenness
- OK if there are single paths to all vertices

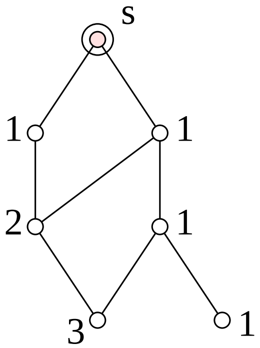




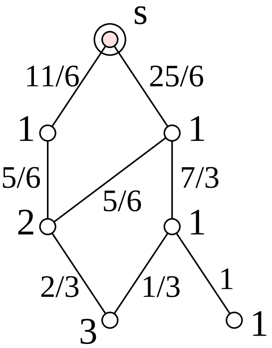
Shortest-path betweenness



Shortest-path betweenness



Shortest-path betweenness



overall time of TopDown is $O(m^2 n)$

Random-walk betweenness

- stochastic matrix of random walk is $P = D^{-1} A$
- $\mathbf{s}$ is the vector with 1 at position s and 0 elsewhere
- probability distribution over vertices at time n is $\mathbf{s}\, P^n$
- expected number of visits at each vertex given by

$$\sum_n \mathbf{s}\, P^n = \mathbf{s}\,(1 - P)^{-1}$$

$$c_u = \mathrm{E}[\#\text{ times passing from } u \text{ to } v] = \left[\mathbf{s}\,(1 - P)^{-1}\right]_u \frac{1}{d_u}$$

$$\mathbf{c} = \mathbf{s}\,(1 - P)^{-1}\, D^{-1} = \mathbf{s}\,(D - A)^{-1}$$

- define *random-walk betweenness* at (u, v) as $|c_u - c_v|$

Random-walk betweenness

- *random-walk betweenness* at (u, v) is $|c_u - c_v|$ with $\mathbf{c} = \mathbf{s}\,(D - A)^{-1}$
- one matrix inversion $O(n^3)$
- in total $O(n^3 m)$ time with recalculation
- not scalable

- *current-flow betweenness* is equivalent!

  [Newman and Girvan, 2004] recommend shortest-path betweenness

Other modularity-based algorithms

spectral approach [Smyth and White, 2005]

$$
\begin{aligned}
Q &= \sum_{c=1}^{k}\left[\frac{m_c}{2m}-\left(\frac{d_c}{2m}\right)^2\right] \propto \sum_{c=1}^{k}\left[(2m)\, m_c - d_c^2\right] \\
&= \sum_{c=1}^{k}\left[(2m)\sum_{i,j=1}^{n} w_{ij}x_{ic}x_{jc} - \left(\sum_{i=1}^{n} d_i x_{ic}\right)^2\right] \\
&= \sum_{c=1}^{k}\left[(2m)\,\mathbf{x}_c^T W\,\mathbf{x}_c - \mathbf{x}_c^T D\,\mathbf{x}_c\right] \\
&= \mathrm{tr}(X^T(W'-D)\,X)
\end{aligned}
$$

where $X = [\mathbf{x}_1 \ldots \mathbf{x}_k] = [x_{ic}]$ point-cluster assignment matrix

Spectral-based modularity optimization

$$\begin{array}{ll} \text{maximize} & \mathrm{tr}(X^T (W' - D)\, X) \\ \text{such that} & X \text{ is an assignment matrix} \end{array}$$

solution:

$$L_Q\, X = X\Lambda$$

where $L_Q = W' - D$, Q-Laplacian

- standard eigenvalue problem
- but solution is fractional, we want integral
- treat rows of X as vectors and cluster graph vertices using k-means
- [Smyth and White, 2005] propose two algorithms, based on this idea

Spectral-based modularity optimization

spectral algorithms perform almost as good as the agglomerative, but they are more efficient

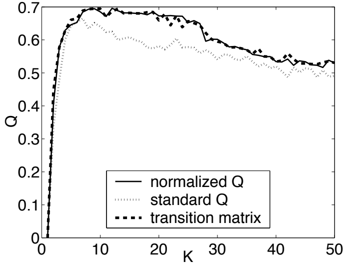


Figure 3: Q versus k for the WordNet data.

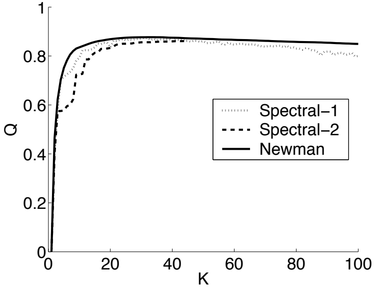


Figure 7: Q versus k for NIPS coauthorship data.

[Smyth and White, 2005]

Other modularity-based algorithms

mathematical programming [Agarwal and Kempe, 2008]

$$Q \;\propto\; \sum_{i,j=1}^{n} B_{ij}(1 - x_{ij})$$

where

$$x_{ij} = \begin{cases} 0 & \text{if } i \text{ and } j \text{ get assigned to the same cluster} \\ 1 & \text{otherwise} \end{cases}$$

it should be

$$x_{ik} \leq x_{ij} + x_{jk} \quad \text{for all vertices } i, j, k$$

solve the integer program with triangle inequality constraints

Mathematical-programming approach for modularity optimization

[Agarwal and Kempe, 2008]

- integer program is **NP**-hard
- relax integrality constraints
  replace $x_{ij} \in \{0, 1\}$ with $0 \leq x_{ij} \leq 1$
- corresponding linear program can be solved in polynomial time
- solve linear program and round the fractional solution
- place in the same cluster vertices i and j if x_{ij} is small
  (pivot algorithm [Ailon et al., 2008])

Results

Network	size n	GN	DA	EIG	VP	LP	UB
KARATE	34	0.401	0.419	0.419	0.420	0.420	0.420
DOLPH	62	0.520	-	-	0.526	0.529	0.531
MIS	76	0.540	-	-	0.560	0.560	0.561
BOOKS	105	-	-	0.526	0.527	0.527	0.528
BALL	115	0.601	-	-	0.605	0.605	0.606
JAZZ	198	0.405	0.445	0.442	0.445	0.445	0.446
COLL	235	0.720	-	-	0.803	0.803	0.805
META	453	0.403	0.434	0.435	0.450	-	-
EMAIL	1133	0.532	0.574	0.572	0.579	-	-

Table 2. The modularity obtained by many of the previously published methods and by the methods introduced in this paper, along with the upper bound.

(table from [Agarwal and Kempe, 2008])

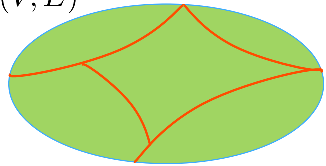
Need for scalable algorithms

- spectral, agglomerative, LP-based algorithms
- not scalable to very large graphs
- handle datasets with billions of vertices and edges
  - `facebook`: $\sim$ 1 billion users with avg degree 130
  - `twitter`: $\geq$ 1.5 billion social relations
  - `google`: web graph more than a trillion edges (2011)
- design algorithms for streaming scenarios
  - real-time story identification using `twitter` posts
  - election trends, `twitter` as election barometer

Graph partitioning

- graph partitioning is a way to split the graph vertices in multiple machines
- graph partitioning objectives guarantee low communication overhead among different machines
- additionally balanced partitioning is desirable

$G = (V, E)$



- each partition contains $\approx n/k$ vertices

Off-line k-way graph partitioning

METIS algorithm [Karypis and Kumar, 1998]

- popular family of algorithms and software
- multilevel algorithm
- coarsening phase in which the size of the graph is successively decreased
- followed by bisection (based on spectral or KL method)
- followed by uncoarsening phase in which the bisection is successively refined and projected to larger graphs

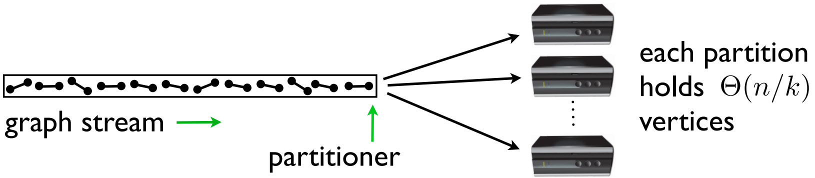
Off-line k-way graph partitioning

Krauthgamer, Naor and Schwartz [Krauthgamer et al., 2009]

- problem: minimize number of edges cut, subject to cluster sizes $\Theta(n/k)$
- approximation guarantee: $O(\sqrt{\log k \log n})$
- based on the work of Arora-Rao-Vazirani for the sparsest-cut problem ($k = 2$) [Arora et al., 2009]

streaming k-way graph partitioning

- input is a data stream
- graph is ordered
  - arbitrarily
  - breadth-first search
  - depth-first search
- generate an approximately balanced graph partitioning



Graph representations

- adjacency stream
  - at time t, a vertex arrives with its neighbors
- edge stream
  - at time t, an edge arrives

Partitioning strategies

- hashing: place a new vertex to a cluster/machine chosen uniformly at random
- neighbors heuristic: place a new vertex to the cluster/machine with the maximum number of neighbors
- non-neighbors heuristic: place a new vertex to the cluster/machine with the minimum number of non-neighbors

Partitioning strategies

[Stanton and Kliot, 2012]

- $d_c(v)$: neighbors of v in cluster c
- $t_c(v)$: number of triangles that v participates in cluster c

- balanced: vertex v goes to cluster with least number of vertices
- hashing: random assignment
- weighted degree: v goes to cluster c that maximizes $d_c(v) \cdot w(c)$
- weighted triangles: v goes to cluster j that maximizes $t_c(v)/\binom{d_c(v)}{2} \cdot w(c)$

Weight functions

- s_c: number of vertices in cluster c
- unweighted: $w(c) = 1$
- linearly weighted: $w(c) = 1 - s_c(k/n)$
- exponentially weighted: $w(c) = 1 - e^{(s_c - n/k)}$

FENNEL algorithm

[Tsourakakis et al., 2012]

$$\begin{aligned} \text{minimize}_{\mathcal{P}=(S_1,\dots,S_k)} \quad & |\partial\, e(\mathcal{P})| \\ \text{subject to} \quad & |S_i| \le \nu \frac{n}{k}, \text{ for all } 1 \le i \le k \end{aligned}$$

- hits the ARV barrier

$$\text{minimize}_{\mathcal{P}=(S_1,\dots,S_k)} \quad |\partial\, E(\mathcal{P})| + c_{\mathrm{IN}}(\mathcal{P})$$

where $c_{\mathrm{IN}}(\mathcal{P}) = \sum_i s(|S_i|)$, so that objective self-balances

- relax hard cardinality constraints

FENNEL algorithm

[Tsourakakis et al., 2012]

- for $S \subseteq V$, $f(S) = e[S] - \alpha|S|^\gamma$, with $\gamma \geq 1$
- given partition $\mathcal{P} = (S_1, \ldots, S_k)$ of V in k parts define

$$g(\mathcal{P}) = f(S_1) + \ldots + f(S_k)$$

- the goal: maximize $g(\mathcal{P})$ over all possible k-partitions
- notice:

$$g(\mathcal{P}) = \underbrace{\sum_i e[S_1]}_{\text{number of edges cut}} - \underbrace{\alpha \sum_i |S_i|^\gamma}_{\text{minimized for balanced partition!}}$$

Connection

notice

$$f(S) = e[S] - \alpha \binom{|S|}{2}$$

- related to modularity
- related to quasicliques (see next)

FENNEL algorithm

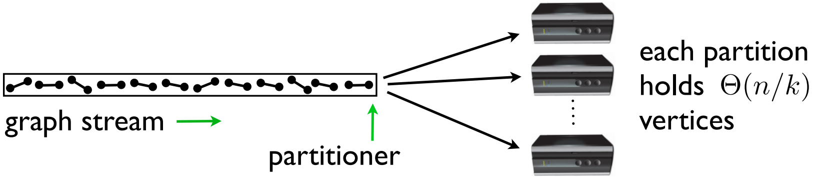
theorem [Tsourakakis et al., 2012]

- $\gamma = 2$ gives approximation factor $\log(k)/k$
  where k is the number of clusters
- random partitioning gives approximation factor $1/k$
- no dependence on n
  mainly because relaxing the hard cardinality constraints

FENNEL algorithm — greedy scheme

- $\gamma = 2$ gives non-neighbors heuristic
- $\gamma = 1$ gives neighbors heuristic
- interpolate between the two heuristics, e.g., $\gamma = 1.5$

FENNEL algorithm — greedy scheme



- send v to the partition / machine that maximizes

$$
\begin{aligned}
& f(S_i \cup \{v\}) - f(S_i) \\
& \qquad = e[S_i \cup \{v\}] - \alpha(|S_i| + 1)^\gamma - (e[S_i] - \alpha|S_i|^\gamma) \\
& \qquad = d_{S_i}(v) - \alpha\mathcal{O}(|S_i|^{\gamma-1})
\end{aligned}
$$

- fast, amenable to streaming and distributed setting

FENNEL algorithm — results

$$\lambda = \frac{\#\{\text{edges cut}\}}{m} \qquad \rho = \max_{1 \le i \le k} \frac{|S_i|}{n/k}$$

		Fennel		METIS	
m	k	λ	ρ	λ	ρ
7 185 314	4	62.5 %	1.04	65.2%	1.02
6 714 510	8	82.2 %	1.04	81.5%	1.02
6 483 201	16	92.9 %	1.01	92.2%	1.02
6 364 819	32	96.3%	1.00	96.2%	1.02
6 308 013	64	98.2%	1.01	97.9%	1.02
6 279 566	128	98.4 %	1.02	98.8%	1.02

- $\gamma = 1.5$
- comparable results in quality, but FENNEL is lightway, fast, and streamable

Conclusions (graph partitioning)

summary

- spectral techniques, modularity-based methods, graph partitioning
- well-studied and mature area

future directions

- develop alternative notions for communities, e.g., accounting for graph labels, constraints, etc.
- further improve efficiency of methods
- overlapping communities

Dense subgraphs

What is a dense subgraph?

- a set of vertices with abundance of edges
- a highly connected subgraph
- key primitive for detecting communities
- related problem to community detection and graph partitioning, but not identical
    - not constrainted for a disjoint partition of all vertices

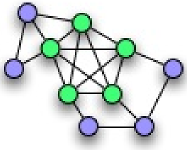
Applications of finding dense subgraphs

- thematic communities and spam link farms [Kumar et al., 1999]
- graph visualization [Alvarez-Hamelin et al., 2005]
- real-time story identification [Angel et al., 2012]
- motif detection [Fratkin et al., 2006]
- epilepsy prediction [Iasemidis et al., 2003]
- finding correlated genes [Zhang and Horvath, 2005]
- many more ...

Density measures

- consider subgraph induced by $S \subseteq V$ of $G = (V, E)$

- clique: each vertex in S is connected to every other vertex in S



- α-quasiclique: the set S has at least $\alpha|S|(|S|-1)/2$ edges
- k-core: every vertex in S is connected to at least k other vertices in S

Density measures

- consider subgraph induced by $S \subseteq V$ of $G = (V, E)$
- density:
$$\delta(S) = \frac{e[S]}{\binom{|S|}{2}} = \frac{2e[S]}{|S|(|S|-1)}$$
- average degree:
$$d(S) = \frac{2e[S]}{|S|}$$
- *k*-densest subgraph:
$$\delta(S) = \frac{2e[S]}{|S|}, \quad \text{such that } |S| = k$$

Density measures

compare with measures we saw previously....

graph expansion:

$$\alpha(G) = \min_{S} \frac{e[S, V \setminus S]}{\min\{|S|, |V \setminus S|\}}$$

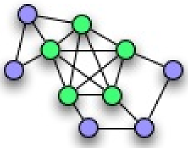
graph conductance:

$$\phi(G) = \min_{S \subseteq V} \frac{e[S, V \setminus S]}{\min\{vol(S), vol(V \setminus S)\}}$$

edges within ($e[S]$) instead of edges accross ($e[S, V \setminus S]$)

Complexity of density problems — clique

- find the max-size clique in a graph:
  NP-hard problem



- strong innaproximability result:

  for any $\epsilon > 0$, there cannot be a polynomial-time algorithm that approximates the maximum clique problem within a factor better than $\mathcal{O}(n^{1-\epsilon})$, unless $\mathbf{P} = \mathbf{NP}$

  [Håstad, 1997]

Complexity of other density problems

density	$\delta(S) = \frac{e[S]}{\binom{\|S\|}{2}}$	pick a single edge
average degree	$d(S) = \frac{2e[S]}{\|S\|}$	in **P**
k-densest subgraph	$\delta(S) = \frac{2e[S]}{\|S\|}, \|S\| = k$	**NP**-hard
DalkS	$\delta(S) = \frac{2e[S]}{\|S\|}, \|S\| \geq k$	**NP**-hard
DamkS	$\delta(S) = \frac{2e[S]}{\|S\|}, \|S\| \leq k$	*L*-reduction to DkS

Densest subgraph problem

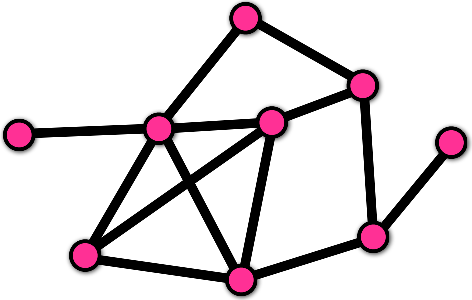
- find set of vertices $S \subseteq V$ with maximum average degree $d(S) = 2e[S]/|S|$
- solvable in polynomial time
  - max-flow [Goldberg, 1984]
  - LP relaxation [Charikar, 2000]
- simple linear-time greedy algorithm gives factor-2 approximation [Charikar, 2000]

Greedy algorithm for densest subgraph

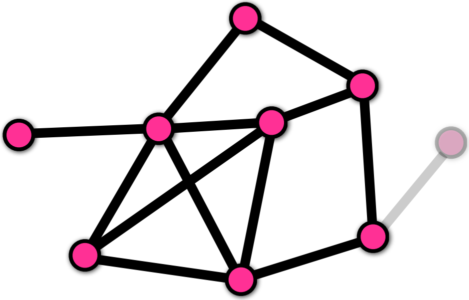
[Charikar, 2000]

input: undirected graph $G = (V, E)$
output: S, a dense sungraph of G
1 set $G_n \leftarrow G$
2 for $k \leftarrow n$ downto 1
2.1 let v be the smallest degree vertex in G_k
2.2 $G_{k-1} \leftarrow G_k \setminus \{v\}$
3 output the densest subgraph among $G_n, G_{n-1}, \ldots, G_1$

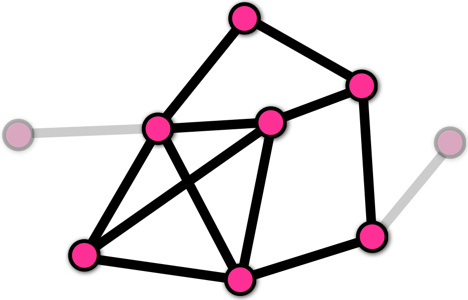
Greedy algorithm for densest subgraph — example



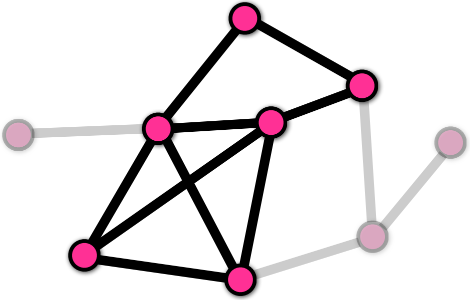
Greedy algorithm for densest subgraph — example



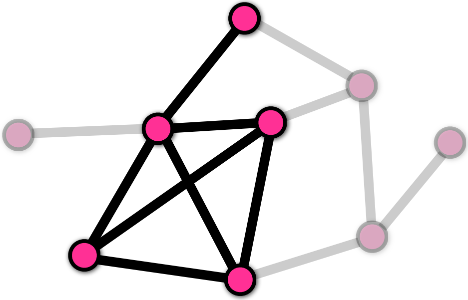
Greedy algorithm for densest subgraph — example



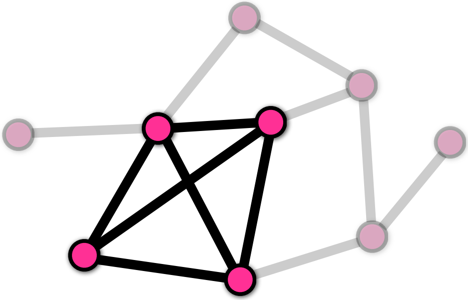
Greedy algorithm for densest subgraph — example



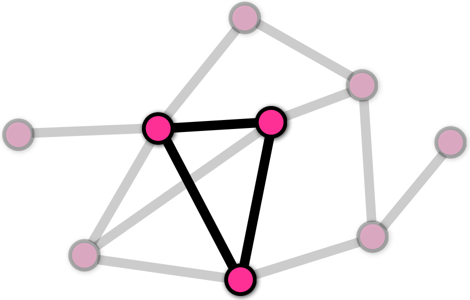
Greedy algorithm for densest subgraph — example



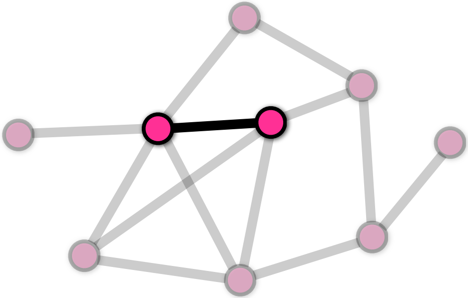
Greedy algorithm for densest subgraph — example



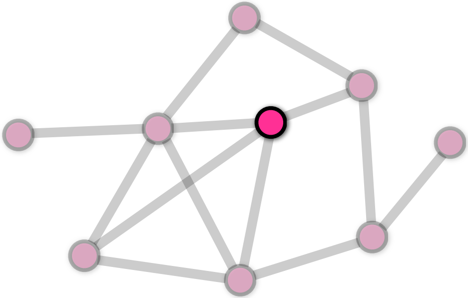
Greedy algorithm for densest subgraph — example



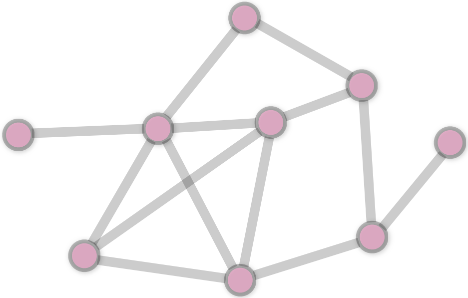
Greedy algorithm for densest subgraph — example



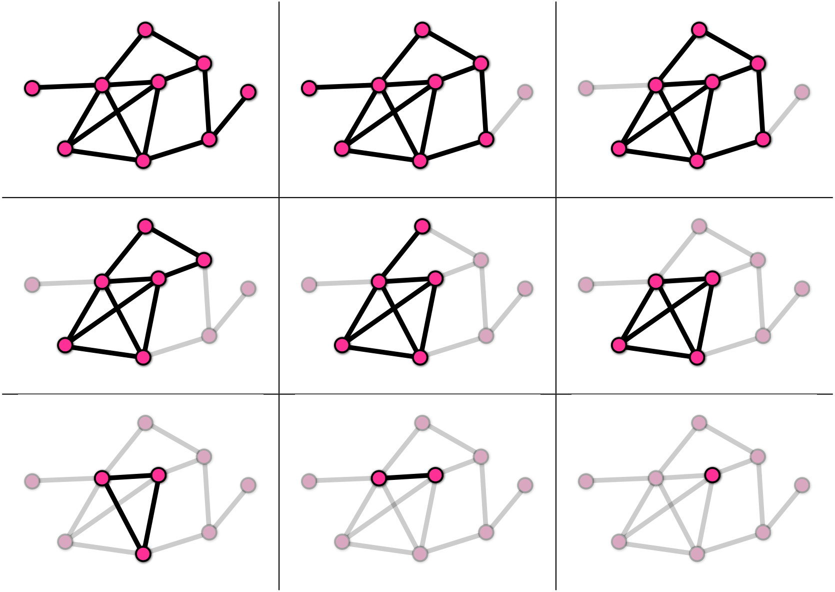
Greedy algorithm for densest subgraph — example



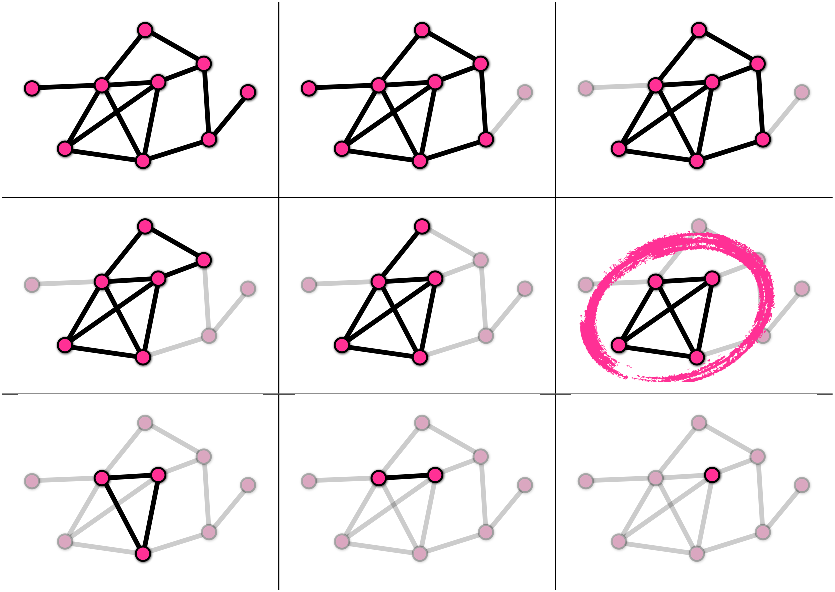
Greedy algorithm for densest subgraph — example



Greedy algorithm for densest subgraph — example



Greedy algorithm for densest subgraph — example



Other notions and generalizations

- *k*-core: every vertex in S is connected to at least k other vertices in S
- α-quasiclique: the set S has at least $\alpha|S|(|S|-1)/2$ edges
- enumerate all α-quasicliques [Uno, 2010]
- dense subgraphs of directed graphs: find sets $S, T \subseteq V$ to maximize

$$d(S,T) = \frac{e[S,T]}{\sqrt{|S|\,|T|}}$$

[Charikar, 2000, Khuller and Saha, 2009]

Edge-surplus framework

- for a set of vertices S define edge surplus

$$f(S) = g(e[S]) - h(|S|)$$

where g and h are both strictly increasing

- optimal (g, h)-edge-surplus problem:

find S^* such that

$$f(S^*) \geq f(S), \quad \text{for all sets } S \subseteq S^*$$

Edge-surplus framework

- edge surplus $f(S) = g(e[S]) - h(|S|)$
- example 1

$$g(x) = h(x) = \log x$$

find S that maximizes $\log \frac{e[S]}{|S|}$

densest-subgraph problem

- example 2

$$g(x) = x, \quad h(x) = \begin{cases} 0 & \text{if } x = k \\ +\infty & \text{otherwise} \end{cases}$$

k-densest-subgraph problem

The optimal quasiclique problem

- edge surplus $f(S) = g(e[S]) - h(|S|)$

- consider
$$g(x) = x, \quad h(x) = \alpha \, \frac{x(x-1)}{2}$$
find S that maximizes $e[S] - \alpha \binom{|S|}{2}$
optimal quasiclique problem [Tsourakakis et al., 2013]

- theorem: let $g(x) = x$ and $h(x)$ concave
then the optimal (g, h)-edge-surplus problem is polynomially-time solvable

The optimal quasiclique problem

theorem: let $g(x) = x$ and $h(x)$ concave

then the optimal (g, h)-edge-surplus problem is polynomially-time solvable

proof

$g(x) = x$ is supermodular

if $h(x)$ concave $h(x)$ is submodular

$-h(x)$ is supermodular

$g(x) - h(x)$ is supermodular

maximizing supermodular functions is solvable in polynomial time

Optimal quasicliques in practice

densest subgraph vs. optimal quasiclique

	densest subgraph				optimal quasi-clique			
	$\frac{\|S\|}{\|V\|}$	δ	D	τ	$\frac{\|S\|}{\|V\|}$	δ	D	τ
Dolphins	0.32	0.33	3	0.04	0.12	0.68	2	0.32
Football	1	0.09	4	0.03	0.10	0.73	2	0.34
Jazz	0.50	0.34	3	0.08	0.15	1	1	1
Celeg. N.	0.46	0.13	3	0.05	0.07	0.61	2	0.26

[Tsourakakis et al., 2013]

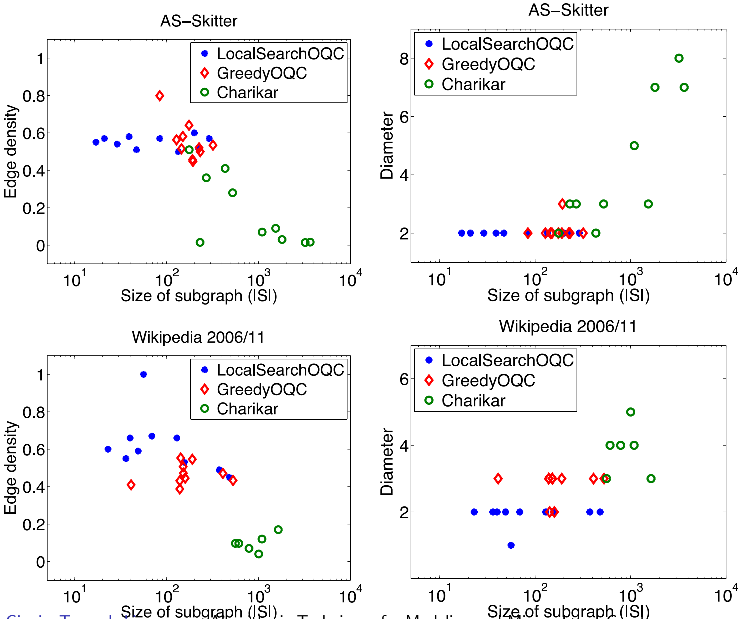
Finding and optimal quasiclique

adaptation of the greedy algorithm of [Charikar, 2000]

input: undirected graph $G = (V, E)$
output: a quasiclique S
1 set $G_n \leftarrow G$
2 for $k \leftarrow n$ downto 1
2.1 let v be the smallest degree vertex in G_k
2.2 $G_{k-1} \leftarrow G_k \setminus \{v\}$
3 output the subgraph in $G_n, \ldots, G_1$ that maximizes $f(S)$

additive approximation guarantee [Tsourakakis et al., 2013]

top-k densest subgraphs and quasicliques



The community-search problem

- a dense subgraph that contains a given subset of vertices $Q \subseteq V$ (the query vertices)

- the center-piece subgraph problem
- the team formation problem
- the cocktail party problem

applications

- find the community of a given set of users
  - a meaningful way to address the issue of overlapping communities
- find a set of proteins related to a given set
- form a team to solve a problem

Center-piece subgraph

[Tong and Faloutsos, 2006]

- given: graph $G = (V, E)$ and set of query vertices $Q \subseteq V$
- find: a connected subgraph H that
  - (a) contains Q
  - (b) optimizes a goodness function $g(H)$

- main concepts:
- `k_softAND`: a node in H should be well connected to at least k vertices of Q
- $r(i, j)$ goodness score of j wrt $q_i \in Q$
- $r(Q, j)$ goodness score of j wrt Q
- $g(H)$ goodness score of a candidate subgraph H
- $H^* = \arg\max_H g(H)$

Center-piece subgraph

[Tong and Faloutsos, 2006]

- $r(i, j)$ goodness score of j wrt $q_i \in Q$

  probability to meet j in a random walk with restart to q_i

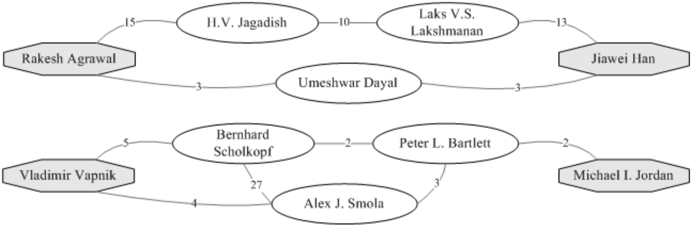
- $r(Q, j)$ goodness score of j wrt Q

  probability to meet j in a random walk with restart to k vertices of Q

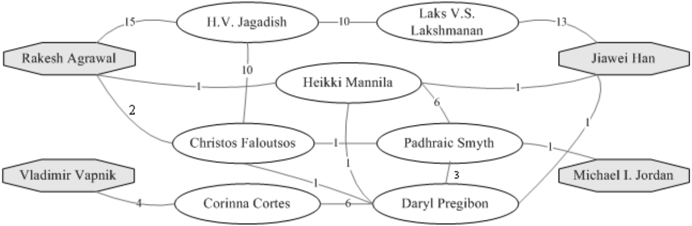
- proposed algorithm:

1. greedy: find a good destination vertex j ito add in H
2. add a path from each of top-k vertices of Q path to j
3. stop when H becomes large enough

Center-piece subgraph — example results



(a) "K_softANDquery": $k = 2$



(b) "AND query"

[Tong and Faloutsos, 2006]

The community-search problem

[Sozio and Gionis, 2010]

- given: graph $G = (V, E)$ and set of query vertices $Q \subseteq V$
- find: a connected subgraph H that
  (a) contains Q
  (b) vertices of H are close to Q
  (c) optimizes a density function $d(H)$

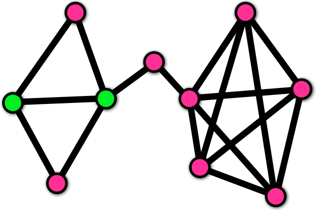
- distance constraint (b):

$$d(Q, j) = \sum_{q \in Q} d^2(q_i, j) \leq B$$

- density function (c):

  average degree, minimum degree, quasiclique, measured on the induced subgraph H

The community-search problem



both the distance constraint and the minimum-degree density help addressing the problem of free riders

The community-search problem

algorithm proposed by [Sozio and Gionis, 2010]

adaptation of the greedy algorithm of [Charikar, 2000]

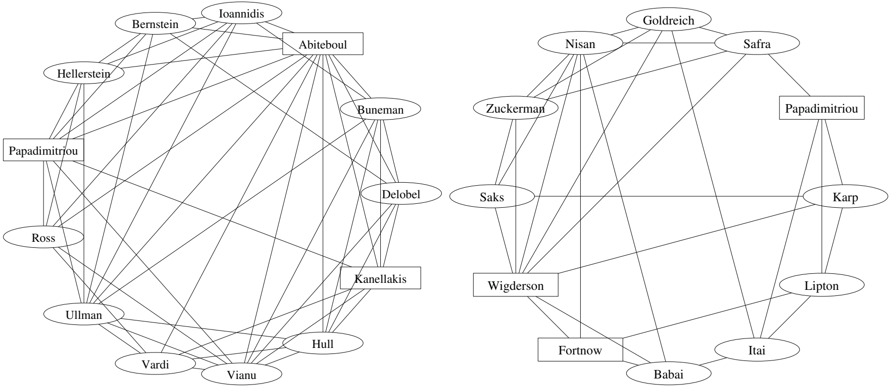
input: undirected graph $G = (V, E)$, query vertices $Q \subseteq V$
output: connected, dense subgraph H

1 set $G_n \leftarrow G$
2 for $k \leftarrow n$ downto 1
2.1 remove all vertices violating distance constraints
2.2 let v be the smallest degree vertex in G_k among all vertices not in Q
2.3 $G_{k-1} \leftarrow G_k \setminus \{v\}$
2.4 if left only with vertices in Q or disconnected graph, stop
3 output the subgraph in $G_n, \ldots, G_1$ that maximizes $f(H)$

Properties of the greedy algorithm

- returns optimal solution if no size constraints or lower-bound constraints
- heuristic variants proposed when upper-bound constraints
- generalized for monotone constraints and monotone objective functions

The community-search problem — example results



(a) Database theory

(b) Complexity theory

(from [Sozio and Gionis, 2010])

Conclusions (dense subgraphs)

summary

- discussed a number of different density measures
- discussed a number of diiferent problem formulations
- polynomial-time solvable or **NP**-hard problems
- global dense subgraphs or relative to query vertices

promising future directions

- explore further the concept of α-quasiclique
- better algorithms for upper-bound constraints
- top-k versions of dense subgraphs
- adapt concepts for labeled graphs
- local algorithms

thank you!

references I

Agarwal, G. and Kempe, D. (2008).
Modularity-maximizing graph communities via mathematical programming.
The European Physical Journal B, 66(3).

Ailon, N., Charikar, M., and Newman, A. (2008).
Aggregating inconsistent information: ranking and clustering.
Journal of the ACM (JACM), 55(5).

Albert, R., Jeong, H., and Barabási, A.-L. (2000).
Error and attack tolerance of complex networks.
Nature, 406(6794):378–382.

references II

Alvarez-Hamelin, J. I., Dall'Asta, L., Barrat, A., and Vespignani, A. (2005).

Large scale networks fingerprinting and visualization using the *k*-core decomposition.

In *NIPS*.

Angel, A., Koudas, N., Sarkas, N., and Srivastava, D. (2012).

Dense Subgraph Maintenance under Streaming Edge Weight Updates for Real-time Story Identification.

arXiv.org.

Arora, S., Rao, S., and Vazirani, U. (2009).

Expander flows, geometric embeddings and graph partitioning.

Journal of the ACM (JACM), 56(2).

references III

Bollobás, B. and Chung, F. R. K. (1988).
The diameter of a cycle plus a random matching.
SIAM Journal on discrete mathematics, 1(3):328–333.

Bollobás, B. and Riordan, O. (2003).
Mathematical results on scale-free random graphs.
Handbook of graphs and networks, 1:34.

Bollobás, B. and Riordan, O. (2004).
Robustness and vulnerability of scale-free random graphs.
Internet Mathematics, 1(1):1–35.

Borgs, C., Chayes, J. T., Ding, J., and Lucier, B. (2011).
The hitchhiker's guide to affiliation networks: A game-theoretic approach.
In *ICS*.

references IV

Brandes, U., Delling, D., Gaertler, M., Görke, R., Höfer, M., Nikoloski, Z., and Wagner, D. (2006).

Maximizing modularity is hard.

Technical report, DELIS – Dynamically Evolving, Large-Scale Information Systems.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000).

Graph structure in the web: Experiments and models.

In *Proceedings of the Ninth Conference on World Wide Web*, pages 309–320, Amsterdam, Netherlands. ACM Press.

Brummitt, C. D., DSouza, R. M., and Leicht, E. (2012).

Suppressing cascades of load in interdependent networks.

Proceedings of the National Academy of Sciences, 109(12):E680–E689.

references V

Buckley, P. G. and Osthus, D. (2004).
Popularity based random graph models leading to a scale-free degree sequence.
Discrete Mathematics, 282(1):53–68.

Callaway, D. S., Hopcroft, J. E., Kleinberg, J. M., Newman, M. E., and Strogatz, S. H. (2001).
Are randomly grown graphs really random?
Physical Review E, 64(4):041902.

Chakrabarti, D., Zhan, Y., and Faloutsos, C. (2004).
R-mat: A recursive model for graph mining.
Computer Science Department, page 541.

references VI

Chakrabarti, S., Frieze, A., and Vera, J. (2005).

The influence of search engines on preferential attachment.

In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 293–300. Society for Industrial and Applied Mathematics.

Charikar, M. (2000).

Greedy approximation algorithms for finding dense components in a graph.

In *APPROX*.

Cho, J. and Roy, S. (2004).

Impact of search engines on page popularity.

In *Proceedings of the 13th international conference on World Wide Web*, pages 20–29. ACM.

references VII

Clauset, A., Newman, M., and Moore, C. (2004).
Finding community structure in very large networks.
arXiv.org.

Clauset, A., Shalizi, C. R., and Newman, M. E. (2009).
Power-law distributions in empirical data.
SIAM review, 51(4):661–703.

Cooper, C. and Frieze, A. (2003).
A general model of web graphs.
Random Structures & Algorithms, 22(3):311–335.

Cooper, C. and Frieze, A. (2004).
Crawling on simple models of web graphs.
Internet Mathematics, 1(1):57–90.

references VIII

Dorogovtsev, S. N. and Mendes, J. F. (2002).
Evolution of networks.
Advances in physics, 51(4):1079–1187.

Doyle, J. and Carlson, J. M. (2000).
Power laws, highly optimized tolerance, and generalized source coding.
Physical Review Letters, 84(24):5656.

Dutta, B. and Jackson, M. O. (2003).
Networks and Groups: Models of Strategic Formation.
Springer Heidelberg.

Fabrikant, A., Koutsoupias, E., and Papadimitriou, C. H. (2002).
Heuristically optimized trade-offs: A new paradigm for power laws in the internet.
In *Automata, languages and programming*, pages 110–122. Springer.

references IX

Fabrikant, A., Luthra, A., Maneva, E., Papadimitriou, C. H., and Shenker, S. (2003).

On a network creation game.

In *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, PODC '03, pages 347–351, New York, NY, USA. ACM.

Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999).

On power-law relationships of the internet topology.

In *SIGCOMM*.

Flaxman, A. D., Frieze, A. M., and Vera, J. (2006).

A geometric preferential attachment model of networks.

Internet Mathematics, 3(2):187–205.

references X

Fratkin, E., Naughton, B. T., Brutlag, D. L., and Batzoglou, S. (2006).

MotifCut: regulatory motifs finding with maximum density subgraphs.

Bioinformatics, 22(14).

Frieze, A., Kleinberg, J., Ravi, R., and Debany, W. (2009).

Line-of-sight networks.

Combinatorics, Probability and Computing, 18(1-2):145–163.

Frieze, A. and Tsourakakis, C. E. (2012).

Rainbow connection of sparse random graphs.

the electronic journal of combinatorics, 19(4):P5.

Frieze, A. and Tsourakakis, C. E. (2013).

Some properties of random apollonian networks.

Internet Mathematics.

references XI

Gleich, D. F. and Owen, A. B. (2012).
Moment-based estimation of stochastic kronecker graph parameters.
Internet Mathematics, 8(3):232–256.

Goldberg, A. V. (1984).
Finding a maximum density subgraph.
Technical report.

Gugelmann, L., Panagiotou, K., and Peter, U. (2012).
Random hyperbolic graphs: degree sequence and clustering.
In *Automata, Languages, and Programming*, pages 573–585. Springer.

Håstad, J. (1997).
Clique is hard to approximate within $n^{1-\epsilon}$.
In *Electronic Colloquium on Computational Complexity (ECCC)*.

references XII

Holme, P. and Kim, B. J. (2002).
Growing scale-free networks with tunable clustering.
Physical Review E, 65(2):026107.

Hopcroft, J. and Kannan, R. (2012).
Computer science theory for the information age.

Iasemidis, L. D., Shiau, D.-S., Chaovalitwongse, W. A., Sackellares, J. C., Pardalos, P. M., Principe, J. C., Carney, P. R., Prasad, A., Veeramani, B., and Tsakalis, K. (2003).
Adaptive epileptic seizure prediction system.
IEEE Transactions on Biomedical Engineering, 50(5).

Karypis, G. and Kumar, V. (1998).
A fast and high quality multilevel scheme for partitioning irregular graphs.
SIAM J. Sci. Comput., 20(1):359–392.

references XIII

Khuller, S. and Saha, B. (2009).
On finding dense subgraphs.
In *ICALP*.

Kleinberg, J. M. (2000).
Navigation in a small world.
Nature, 406(6798):845–845.

Kleinberg, J. M., Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. S. (1999a).
The web as a graph: Measurements, models, and methods.
In *Computing and combinatorics*, pages 1–17. Springer.

references XIV

Kleinberg, J. M., Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. S. (1999b).

The Web as a graph: measurements, models and methods.

In *Proceedings of the 5th Annual International Computing and Combinatorics Conference (COCOON)*, volume 1627 of *Lecture Notes in Computer Science*, pages 1–18, Tokyo, Japan. Springer.

Krauthgamer, R., Naor, J. S., and Schwartz, R. (2009).

Partitioning graphs into balanced components.

In *SODA*.

Krivelevich, M. and Sudakov, B. (2013).

The phase transition in random graphs - a simple proof.

references XV

Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., and Upfal, E. (2000).

Stochastic models for the web graph.

In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 57–65, Redondo Beach, CA, USA. IEEE CS Press.

Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. (1999).

Trawling the Web for emerging cyber-communities.

Computer Networks, 31(11–16):1481–1493.

Lakhina, A., Byers, J. W., Crovella, M., and Xie, P. (2003).

Sampling biases in ip topology measurements.

In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 332–341. IEEE.

references XVI

Lattanzi, S. and Sivakumar, D. (2009).

Affiliation networks.

In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 427–434. ACM.

Leskovec, J., Chakrabarti, D., Kleinberg, J., and Faloutsos, C. (2005a).

Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication.

In *Knowledge Discovery in Databases: PKDD 2005*, pages 133–145. Springer.

Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., and Ghahramani, Z. (2010).

Kronecker graphs: An approach to modeling networks.

The Journal of Machine Learning Research, 11:985–1042.

references XVII

Leskovec, J. and Faloutsos, C. (2007).

Scalable modeling of real graphs using kronecker multiplication.

In *Proceedings of the 24th international conference on Machine learning*, pages 497–504. ACM.

Leskovec, J., Kleinberg, J., and Faloutsos, C. (2005b).

Graphs over time: densification laws, shrinking diameters and possible explanations.

In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187, New York, NY, USA. ACM Press.

Leskovec, J., Kleinberg, J., and Faloutsos, C. (2007).

Graph evolution: Densification and shrinking diameters.

ACM Transactions on Knowledge Discovery from Data (TKDD), 1(1):2.

references XVIII

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. (2009).
Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters.
Internet Mathematics, 6(1):29–123.

Li, L., Alderson, D., Doyle, J. C., and Willinger, W. (2005).
Towards a theory of scale-free graphs: Definition, properties, and implications.
Internet Mathematics, 2(4):431–523.

Łuczak, T. and Prałat, P. (2006).
Protean graphs.
Internet Mathematics, 3(1):21–40.

references XIX

Mahdian, M. and Xu, Y. (2007).

Stochastic kronecker graphs.

In *Algorithms and models for the web-graph*, pages 179–186. Springer.

McKay, B. D. and Wormald, N. C. (1997).

The degree sequence of a random graph. i. the models.

Random Structures and Algorithms, 11(2):97–117.

Newman, M. (2004).

Fast algorithm for detecting community structure in networks.

Physical review E, 69(6).

Newman, M. E. J. (2003).

The structure and function of complex networks.

references XX

Newman, M. E. J. and Girvan, M. (2004).
Finding and evaluating community structure in networks.
Physical Review E, 69(2).

Ng, A., Jordan, M., and Weiss, Y. (2001).
On spectral clustering: Analysis and an algorithm.
NIPS.

Ostroumova, L., Ryabchenko, A., and Samosvat, E. (2012).
Generalized preferential attachment: tunable power-law degree distribution and clustering coefficient.
arXiv preprint arXiv:1205.3015.

Penrose, M. (2003).
Random geometric graphs, volume 5.
Oxford University Press Oxford.

references XXI

Pinar, A., Seshadhri, C., and Kolda, T. G. (2011).

The similarity between stochastic kronecker and chung-lu graph models.

arXiv preprint arXiv:1110.4925.

Shi, J. and Malik, J. (2000).

Normalized cuts and image segmentation.

IEEE transactions on Pattern Analysis and Machine Intelligence, 22(8).

Smyth, P. and White, S. (2005).

A spectral clustering approach to finding communities in graphs.

SDM.

references XXII

Sozio, M. and Gionis, A. (2010).

The community-search problem and how to plan a successful cocktail party.

In *KDD*.

Stanton, I. and Kliot, G. (2012).

Streaming graph partitioning for large distributed graphs.

In *KDD*.

Tong, H. and Faloutsos, C. (2006).

Center-piece subgraphs: problem definition and fast solutions.

In *KDD*.

references XXIII

Tsourakakis, C., Bonchi, F., Gionis, A., Gullo, F., and Tsiarli, M. (2013).

Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees.

In *KDD*.

Tsourakakis, C. E. (2008).

Fast counting of triangles in large real networks without counting: Algorithms and laws.

In *ICDM*.

Tsourakakis, C. E., Gkantsidis, C., Radunovic, B., and Vojnovic, M. (2012).

FENNEL: Streaming graph partitioning for massive scale graphs.

Technical report.

references XXIV

Uno, T. (2010).

An efficient algorithm for solving pseudo clique enumeration problem.

Algorithmica, 56(1).

Valiant, L. G. (2005).

Memorization and association on a realistic neural model.

Neural computation, 17(3):527–555.

von Luxburg, U. (2007).

A Tutorial on Spectral Clustering.

arXiv.org.

Zhang, B. and Horvath, S. (2005).

A general framework for weighted gene co-expression network analysis.

Statistical applications in genetics and molecular biology, 4(1):1128.