Mining the graph structures of the web

Aristides Gionis

Yahoo! Research, Barcelona, Spain, and
University of Helsinki, Finland

Summer School on Algorithmic Data Analysis (SADA07)
May 28 – June 1, 2007
Helsinki, Finland

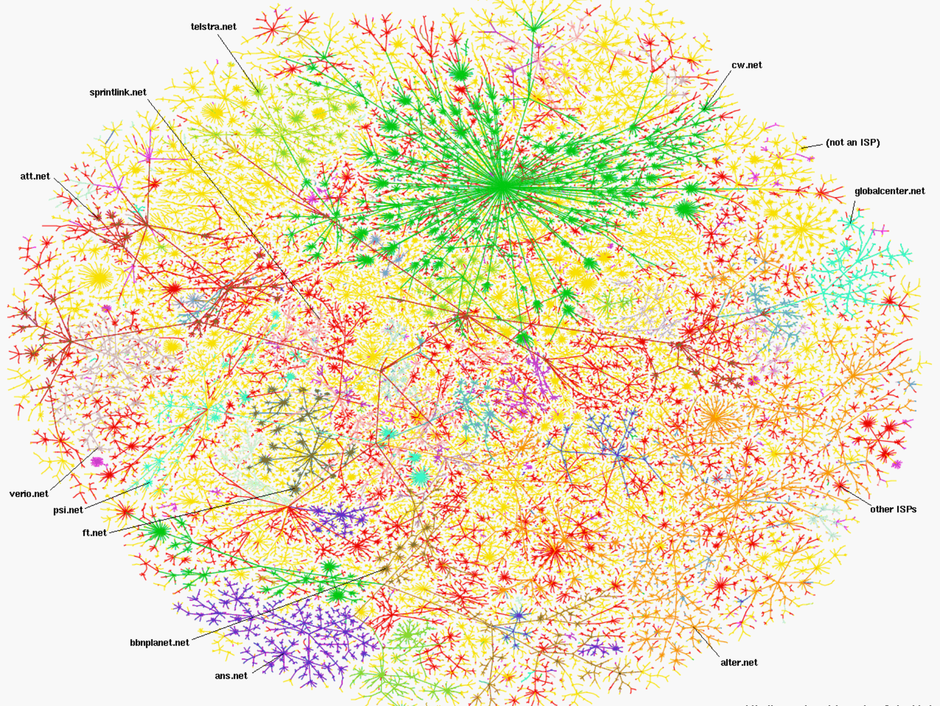
Graphs in the web

A large wealth of data in the web can be represented as graphs

- Rich amounts of information
- Complex interactions among the entities they represent

To extract the information represented in those graphs need

- Understanding of the generating processes
- Analysis of graphs at different levels
- Efficient data mining algorithms



Burch/Cheswick map of the Internet
showing the major ISPs. Data collected 28 June 1999

http://www.cheswick.com/map/index.html

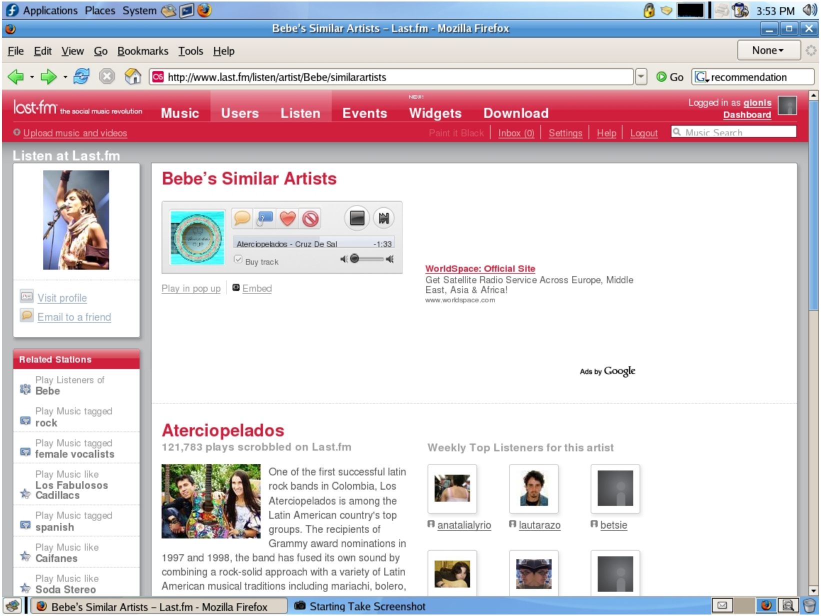
Graphs in the web

- Internet graph
- Web graph
- Blogs
  - Collaborative topical discussions
- Social networks
  - friendship networks, buddy lists, orkut, 360^o
- Photo/video sharing and tagging
  - Flickr, You Tube
- Yahoo! answers
- Query logs

How to take advantage

- Information dissemination
- Retrieve information for tasks otherwise "too difficult"
- Recommendations, suggestions
- Personalization

Listen and explore music as a member of a community



Find a photo of a 'Dali painting' in Flickr

Graph datasets are universal

- Protein interaction networks
- Gene regulation networks
- Gene co-expression networks
- Neural networks
- Food webs
- Citation graphs
- Collaboration graphs (scientists, actors)
- Word co-occurrence graphs

Agenda

Thu 31/5: Tutorial on mining graphs:
models and algorithms

Fri 1/6: Applications:
Spam detection and reputation prediction

Basic notation

- Graph $G = (V, E)$
- V a set of n vertices
- $E \subseteq V \times V$ a set of m edges
- Directed or undirected graphs
- $N(u) = \{v \mid (u, v) \in E\}$ neighbors of u
- $d(u) = |N(u)|$ degree of u
- In-degree and out-degree in the directed case

Basic notation

- $u = x_0, x_1, \ldots, x_{k-1}, x_k = v$ path of length k from u to v, if $(x_i, x_{i+1}) \in E$
- u and v are connected if there is a path from u to v
- Connected component: a subset of vertices each pair of which are connected
- $d(u, v)$: shortest path from u to v
- $D_G = \max_{u,v} d(u, v)$: diameter of the graph

Extensions

- Weights on the vertices and/or the edges
- Types on the vertices and/or the edges
- Feature vectors, e.g., text

Properties of graphs at different levels

Diverse collections of graphs arising from different phenomena

Are there any typical patterns?

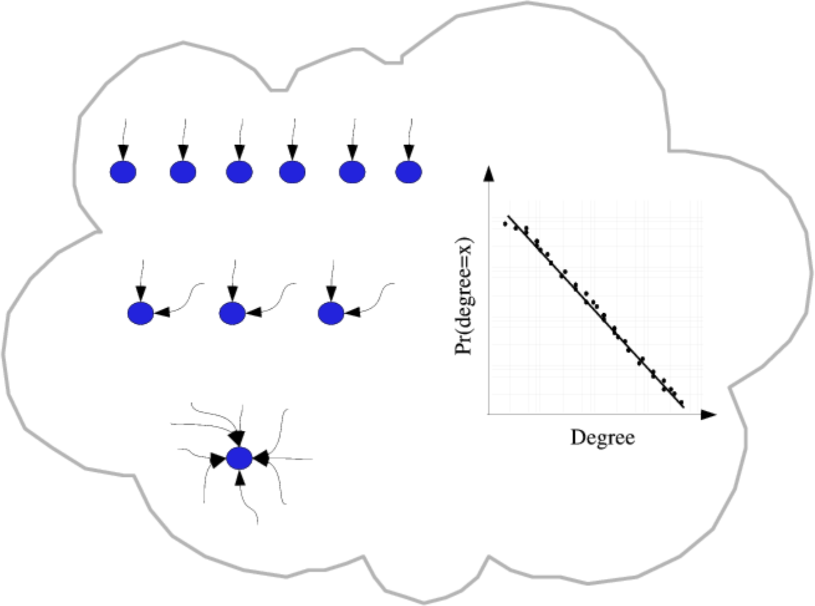
At which level should we look for commonalities?

- Degree distribution — microscopic
- Communities — mesoscopic
- Small diameters — macroscopic

Degree distribution

- Consider C_k the number of vertices u with degree $d(u) = k$. Then
$$C_k = ck^{-\gamma},$$
with $\gamma > 1$, or
$$\ln C_k = \ln c - \gamma \ln k$$
- So, 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)

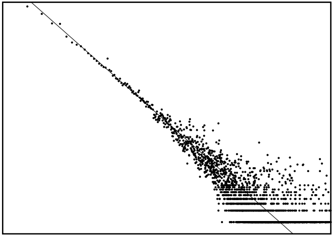
Degree distribution



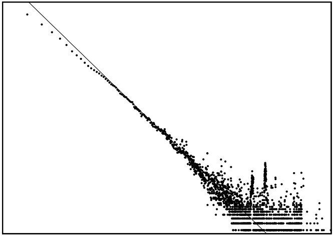
Degree distribution

Indegree distributions of Web graphs within national domains

Greece



Spain

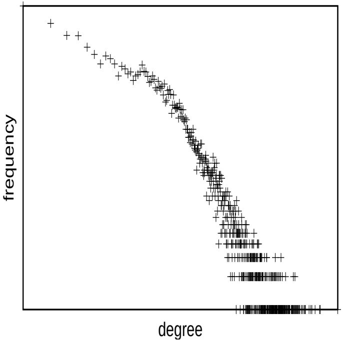


[Baeza-Yates and Castillo, 2005]

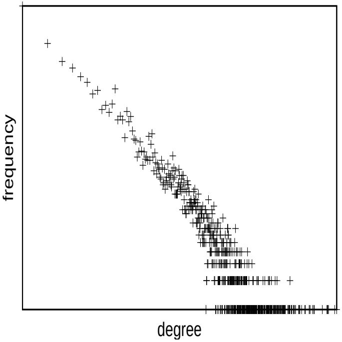
Degree distribution

...and more "straight" lines

In-degrees of UK hostgraph



Out-degrees of UK hostgraph



Community structure

- Intuitively a subset of vertices that are more connected to each other than to other vertices in the graph
- A proposed measure is *clustering coefficient*

$$C_1 = \frac{3 \times \text{ number of triangles in the network}}{\text{number of connected triples of vertices}}$$

- 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

Community structure

- 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*

$$C_2 = \frac{1}{n} \sum_i C_i$$

- Community structure is captured by large values of clustering coefficient

Small diameter

Diameter of many real graphs is small (e.g., $D = 6$ is famous)

Proposed measures

- *Hop-plots:* plot of $|N_h(u)|$, the number of neighbors of u at distance at most h, as a function of h
  [M. Faloutsos, 1999] conjectured that it grows exponentially and considered *hop exponent*
- *Effective diameter:* upper bound of the shortest path of 90% of the pairs of vertices
- *Average diameter:* average of the shortest paths over all pairs of vertices
- *Characteristic path length:* median of the shortest paths over all pairs of vertices

Measurements on real graphs

Graph	n	m	α	C_1	C_2	ℓ
film actors	449 913	25 516 482	2.3	0.20	0.78	3.48
Internet	10 697	31 992	2.5	0.03	0.39	3.31
protein interactions	2 115	2 240	2.4	0.07	0.07	6.80

[Newman, 2003b]

Random graphs

- Erdös-Rényi random graphs have been used as point of reference
- The basic random graph model:
- n : the number of vertices
- $0 \leq p \leq 1$
- 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 $p^m(1-p)^{\binom{n}{2}-m}$
- $z = np$

Random graphs

- Do they satisfy properties similar with those of real graphs?
- Typical distance $d = \frac{\ln n}{\ln z}$ ✔
  - Number of vertices at distance l is $\simeq z^l$, set $z^d \simeq n$
- Poisson degree distribution ✗

$$p_k = \binom{n}{k} p^k (1-p)^{n-k} \simeq \frac{z^k e^{-z}}{k}$$

  - highly concentrated around the mean ($z = np$)
  - probability of very high degree nodes is exponentially small
- Clustering coefficient $C = p$ ✗
  - probability that two neighbors of a vertex are connected is independent of the local structure

Other properties

- Degree correlations
- Distribution of size of connected components
- Resilience
- Eigenvalues
- Distribution of motifs

Properties of evolving graphs

- [Leskovec et al., 2005] discovered two interesting and counter-intuitive phenomena
- Densification power law

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

- Diameter is shrinking

Next...

- Delve deeper into the above properties of graphs
  - Power laws on degree distribution
  - Communities
  - Small diameters
- Generative models and algorithms

Power law distributions

- *"A Brief History of Generative Models for Power Law and Lognormal Distributions"* [Mitzenmacher, 2004]
- A random variable X has *power law distribution*, if

$$\Pr[X \geq x] \sim cx^{-\alpha} \qquad \text{for } c > 0, \text{ and } \alpha > 0.$$

- Random variable X has *Pareto distribution*, if

$$\Pr[X \geq x] = (\frac{x}{k})^{-\alpha} \qquad \text{for } \alpha > 0, \text{ and } k > 0, \text{ where } X \geq k.$$

- Density function of Pareto

$$f(x) = \alpha k^{\alpha} x^{-(\alpha+1)}$$

Scale-free distributions

- Or scaling distributions.
  Since
  $$\Pr[X \geq x] = cx^{-\alpha}$$
  then
  $$\Pr[X \geq x | X \geq w] = c_1 x^{-\alpha}$$
  Thus the conditional distribution $\Pr[X \geq x | X \geq w]$ is identical to $\Pr[X \geq x]$, except from a change in scale

Signature of a power law

- From $\Pr[X \geq x] = (\frac{x}{k})^{-\alpha}$ we get

$$\ln(\Pr[X \geq x]) = -\alpha(\ln x - \ln k)$$

  So, a straight line on a log-log plot (slope $-\alpha$)
- Similarly for the density function (slope $-\alpha - 1$)
- Usually $0 \leq \alpha \leq 2$
- if $\alpha \leq 2$ infinite variance
- if $\alpha \leq 1$ infinite mean

A process that generates power law

Preferential attachment

- The main idea is that "the rich get richer"
- First studied by [Yule, 1925]
  - to suggest a model of why the number of species in genera follows a power-law
- Generalized by [Simon, 1955]
  - applications in distribution of word frequencies, population of cities, income, etc.
- Revisited in the 90s as a basis for Web-graph models
  - [Barabási and Albert, 1999, Broder et al., 2000, Kleinberg et al., 1999]

Preferential attachment

The basic theme

- Start with a single vertex, with a link to itself
- At each time step a new vertex u appears with outdegree 1 and gets connected to an existing vertex v
- With probability $\alpha < 1$, vertex v is chosen uniformly at random
- With probability $1 - \alpha$, vertex v is chosen with probability proportional to its degree
- Process leads to power law for the indegree distribution, with exponent $\frac{2-\alpha}{1-\alpha}$

Lognormal distribution

- Random variable X has *lognormal distribution* if $Y = \ln X$ has normal distribution. Since

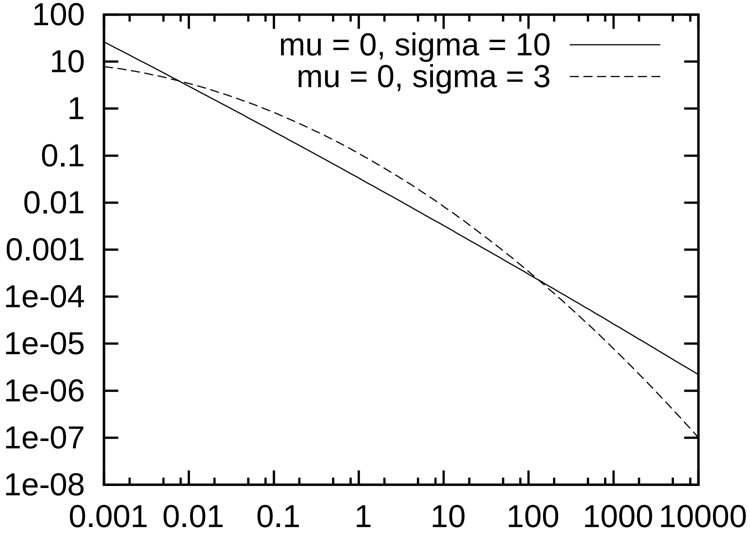
  $f(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-\mu)^2/2\sigma^2}$, it is $f(x) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-(\ln x-\mu)^2/2\sigma^2}$.

- Always finite mean and variance
- But it also appears a straight line on a log-log plot

$$\begin{aligned} \ln f(x) &= \ln x - \ln \sqrt{2\pi}\sigma - \frac{(\ln x - \mu)^2}{2\sigma^2} \\ &= -\frac{(\ln x)^2}{2\sigma^2} + (\frac{\mu}{\sigma^2} - 1) \ln x - \ln \sqrt{2\pi}\sigma - \frac{\mu^2}{2\sigma^2} \end{aligned}$$

So, if σ^2 is large, then quadratic term is small for a large range of values of x

Lognormal distribution



Multiplicative models

- Let two independent random variables Y_1 and Y_2 have normal distribution with means μ_1 and μ_2 and variances σ_1^2 and σ_2^2, resp.
- Then $Y = Y_1 + Y_2$ has normal distribution, too, with mean $\mu_1 + \mu_2$ and variance $\sigma_1^2 + \sigma_2^2$
- So the *product* of two lognormally distributed independent random variables follows a lognormal distribution

Multiplicative models

- Assume a generative process

$$X_j = F_j X_{j-1},$$

e.g., the size of a population might grow or shrink according to a random variable F_j. Then

$$\ln X_j = \ln X_0 + \sum_{k=1}^{j} \ln F_k$$

- If $(\ln F_k)$ are i.i.d. with mean μ and finite variance σ^2, then by Central Limit Theorem, for large values of j, X_j can be approximated by a lognormal
- Proposed to model the growth of sites of the Web, as well as the growth of user traffic on Web sites [Huberman and Adamic, 1999]

Power law or lognormal?

- Distribution of income
- Start with some income X_0
- At time t with probability $1/3$ double the income, with probability $2/3$ cut the income in half
- Then, income distribution is lognormal

Power law or lognormal?

- Assume now a "reflective barrier":
- At X_0 maintain the same income with prob. $2/3$
- Call "having income $X = X_0 2^{k-1}$" as "being in state k"
- Equilibrium probability of being in state k is $1/2^k$
- Probability of being in state $\geq k$ is $1/2^{k-1}$

$$\Pr[X \geq X_0 2^{k-1}] = 1/2^{k-1}, \text{ or}$$

$$\Pr[X \geq x] = \frac{X_0}{x}$$

a power law!

A look back at the data..

Graph	n ($\times 1000$)	m ($\times 1000$)	α	C_1	C_2	ℓ
film actors	449	25 516	2.3	0.20	0.78	3.48
internet	10	31	2.5	0.03	0.39	3.31
protein interactions	2	2	2.4	0.07	0.07	6.80
word co-occurrence	460	17 000	2.8		0.44	
telephone call graph	47 000	80 000	2.1			
www altavista	203 549	2 130 000	2.1/2.7			
sexual contacts	2		3.2			

[Newman, 2003b]

Clustering coefficient

$$C = \frac{3 \times \text{ number of triangles in the network}}{\text{number of connected triples of vertices}}$$

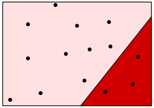
- How to compute it?
- How to compute the number of triangles in a graph?
- Assume that the graph is very large, stored in disk
- [Buriol et al., 2006]
- Count triangles, when graph is seen as a data stream
- Two models:
  - edges are stored in any order
  - edges in order — all edges incident to one vertex are stored sequentially

Counting triangles

- Brute-force algorithm is checking every triple of vertices
- Obtain an approximation by sampling triples
- Let T be the set of all triples and
  T_i the set of triples that have i edges, $i = 0, 1, 2, 3$
- By Chernoff bound, to get an ϵ-approximation, with probability $1 - \delta$, the number of samples should be

$$N \geq O\left(\frac{|T|}{|T_3|}\frac{1}{\epsilon^2}\log\frac{1}{\delta}\right)$$

but $|T|$ can be very large compared to $|T_3|$



Counting triangles — incidence stream model

SAMPLETRIANGLE [Buriol et al., 2006]

1st Pass

Count the number of paths of length 2 in the stream

2nd Pass

Uniformly choose one path (a, u, b)

3rd Pass

if $((a, b) \in E)$ $\beta = 1$ else $\beta = 0$

return β

We have $\mathrm{E}[\beta] = \frac{3|T_3|}{|T_2|+3|T_3|}$, with $|T_2| + 3|T_3| = \sum_u \frac{d_u(d_u-1)}{2}$, so

$$|T_3| = \mathrm{E}[\beta] \sum_u \frac{d_u(d_u-1)}{6}$$

and space needed is $O((1 + \frac{|T_2|}{|T_3|})\frac{1}{\epsilon^2} \log \frac{1}{\delta})$

Counting triangles

The previous idea can be also applied to

- Count triangles when edges are stored in arbitrary order
- Obtain one-pass algorithm
- Count other minors

Diameter

- How to compute the diameter of a graph?
- Matrix multiplication in $O(n^{2.376})$ time,
  but $O(n^2)$ space
- BFS from a vertex takes $O(n+m)$ time,
  but need to do it from every vertex, so $O(mn)$
- Resort to approximations again

Approximating the diameter

- [Palmer et al., 2002], see also [Cohen, 1997]
- Define:

  Individual neighborhood function

  $$N(u,h) = |\{v \mid d(u,v) \le h\}|$$

  Neighborhood function

  $$N(h) = |\{(u,v) \mid d(u,v) \le h\}| = \sum_u N(u,h)$$

- $N(h)$ can be used to obtain diameter, effective diameter, etc.

Approximating the diameter

- Define: $M(u,h) = \{v \mid d(u,v) \leq h\}$, e.g., $M(u,0) = \{u\}$
- Algorithm based on the idea that
  $x \in M(u,h)$ if $(u,v) \in E$ and $x \in M(v,h-1)$

ANF [Palmer et al., 2002]

$M(u,0) = \{u\}$ for all $u \in V$
for each distance h **do**
  $M(u,h) = M(u,h-1)$ for all $u \in V$
  for each edge (u,v) **do**
    $M(u,h) = M(u,h) \cup M(v,h-1)$

- Keep $M(u,h)$ in memory, make a passes over the edges
- How to maintain $M(u,h)$?

Approximating the diameter

- How to maintain $M(u, h)$ that it counts *distinct* vertices?
- The problem of counting distinct elements in data streams
- ANF uses the sketching algorithm of [Flajolet and Martin, 1985] with $O(\log n)$ space (but other counting algorithms can be used [Bar-Yossef et al., 2002])
- What if the $M(u, h)$ sketches do not fit in memory?
- Split $M(u, h)$ sketches into in-memory blocks, load one block at the time, and process edges from that block

Conclusions

Real graphs coming from applications and generated from different processes have many commonalities

- Power law distribution of the degree sequences
- Communities
- Small diameters
- Power law distribution of size of connected components
- Resilience
- Eigenvalues

Finding communities

- A set of related Web pages
- A group of scientists collaborating with each other
- A set of blog posts discussing a specific topic
- A set of related queries

- Formulated as a *graph clustering* problem

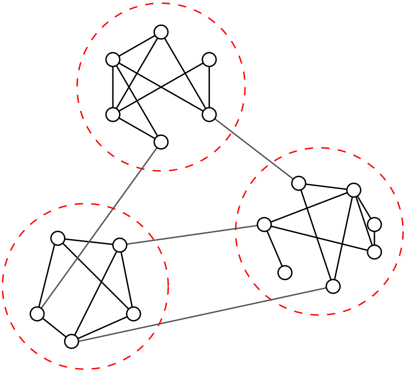
Graph clustering

- Graph $G = (V, E)$
- Edge (u, v) denotes similarity between u and v
  - weighted edges can be used to denote degree of similarity

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

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

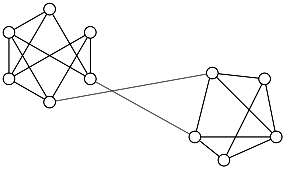
Graph clustering



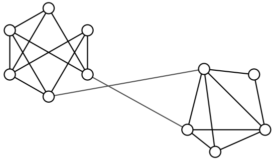
Measuring connectivity

- *minimum cut*: The minimum number of edges whose removal disconnects the graph
- $c(S) = \min_{S \subseteq V} |\{(u, v) \in E \mid u \in S \text{ and } v \in V - S\}$

G_1

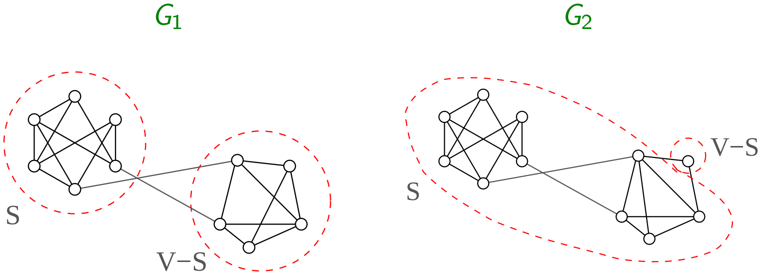


G_2



Measuring connectivity

- *minimum cut*: The minimum number of edges whose removal disconnects the graph
- $c(S) = \min_{S \subseteq V} |\{(u,v) \in V \mid u \in S \text{ and } v \in V - S\}$



Graph expansion

- Normalize the cut by the size of the smallest component
- Define *cut ratio*

$$\alpha(G, S) = \frac{c(S)}{\min\{|S|, |V - S|\}}$$

- And *graph expansion*

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

- Other similar normalized criteria have been proposed
- Related to the eigenvalues of the adjacency matrix of the graph, thus with the *expansion* properties of the graph

Spectral analysis

- Let A be the adjacency matrix of the graph G
- Define the *Laplacian matrix* of A as

$$L = D - A,$$

- $D = \mathrm{diag}(d_1, \ldots, d_n)$, a *diagonal* matrix
- d_i the degree of vertex i

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

- L is symmetric positive semidefinite
- The smallest eigenvalue of L is $\lambda_1 = 0$, with corresponding eigenvector $\mathbf{w}_1 = (1, 1, \ldots, 1)^T$

Spectral analysis

- For the second smallest eigenvector λ_2 of L

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

- Corresponding eigenvector $\mathbf{w}_2$ is called *Fielder vector*
- The ordering according to the values of $\mathbf{w}_2$ will group similar (connected) vertices together
- Physical interpretation: The stable state of springs placed on the edges of the graph, when graph is forced to 1 dimension

Spectral partition

- Partition the nodes according to the ordering induced by the Fielder vector

- Some partitioning rules:
- *Bisection:* s is the median value in $\mathbf{w}_2$
- *Cut ratio:* find the partition that minimizes α
- *Sign:* Separate positive and negative values
- *Gap:* Separate according to the largest gap in the values of $\mathbf{w}_2$

- Spectral partition works very well in practice
- However, not scalable

Spectral algorithms

- [Kannan et al., 2004]: Use *conductance* instead of *graph expansion* (weight vertices by their degree)
- Bicriterion: Find a clustering in which all clusters have large conductance and the number of across-cluster edges is small
- Apply spectral partition to cluster the graph recursively
- Polylogarithmic quality guarantees

- [Cheng et al., 2006]: Enhance previous algorithm by a merging post-processing phase:
- Merge using dynamic programming in order to find a tree-respecting clustering that optimizes a given objective function

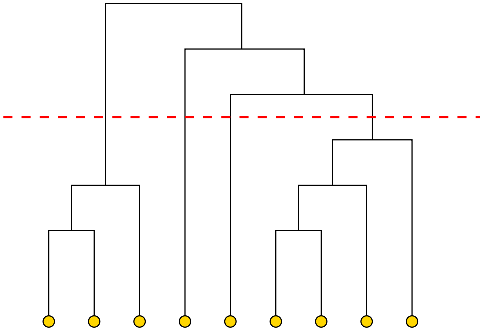
http://eigencluster.csail.mit.edu/

METIS graph partition

- Popular family of algorithms and software [Karypis and Kumar, 1998]
- 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

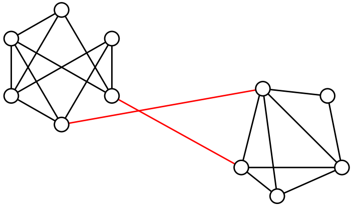
Top down algorithms

- [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



Top down algorithms

- Select edge to remove based on *"betweeness"*



Three definitions

- *Shortest-path betweeness:* Number of shortest paths that the edge belongs to
- *Random-walk betweeness:* Expected number of paths for a random walk from u to v
- *Current-flow betweeness:* Resistance derived from considering the graph as an electric circuit

Top down algorithms — overview

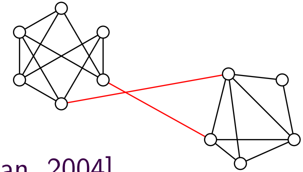
TopDown_0 [Newman and Girvan, 2004]

1. Compute betweeness value of all edges

2. Remove the edge with the highest betweeness

3. Repeat until no edges left

Problem with "ties":



TopDown [Newman and Girvan, 2004]

1. Compute betweeness value of all edges

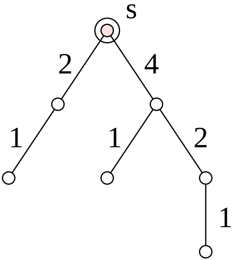
2. Remove the edge with the highest betweeness

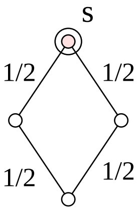
3. Recompute betweeness value of all remaining edges

4. Repeat until no edges left

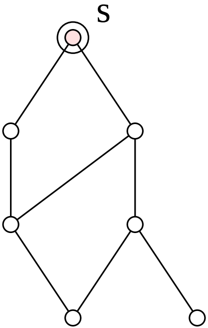
Shortest-path betweeness

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

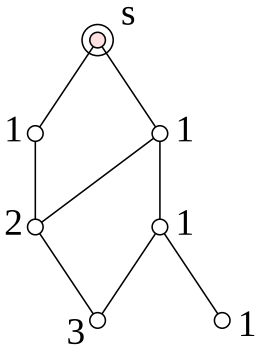




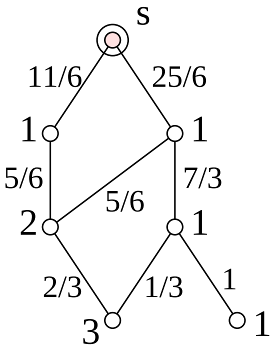
Shortest-path betweeness



Shortest-path betweeness

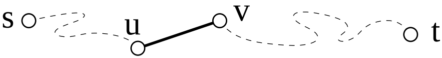


Shortest-path betweeness



Overall time of TOPDOWN is $O(m^2 n)$

Random-walk betweeness

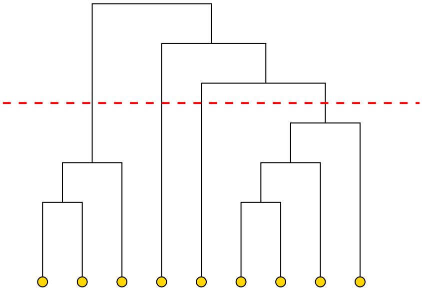


- Stochastic matrix of random walk is $M = D^{-1} \cdot A$
- with $D = \mathrm{diag}(d_1, \ldots, d_n)$, so row i divided by d_i
- Let M_t be M after removing the t-th row and the t-th column
- and $\mathbf{s}$ be the vector with 1 at position s and 0 elsewhere
- Probability distribution over vertices at time n is $\mathbf{s} \cdot M_t^n$
- Expected number of visits at each vertex is $\sum_n \mathbf{s} \cdot M_t^n = \mathbf{s} \cdot (1 - M_t)^{-1}$
- $c_u = \mathrm{E}[\#\text{ times passing from } u \text{ to } v] = \left(\mathbf{s} \cdot (1 - M_t)^{-1}\right)_u \cdot \frac{1}{d_u}$
- $\mathbf{c} = \mathbf{s} \cdot (1 - M_t)^{-1} \cdot D^{-1} = \mathbf{s} \cdot (D_t - A_t)^{-1}$
- Define *random-walk betweeness* at (u, v) as $|c_u - c_v|$

Random-walk betweeness

- *Random-walk betweeness* at (u, v) is $|c_u - c_v|$
- with $\mathbf{c} = \mathbf{s} \cdot (D_t - A_t)^{-1}$
- The choice of vertex t does not matter
- Required one matrix inversion $O(n^3)$ and additional $O(nm)$ time to calculate the betweeness values on all edges
- In total $O(n^3 m)$ time with recalculation
- Not scalable
- *Current-flow betweeness* is equivalent!
- According to [Newman and Girvan, 2004] shortest-path betweeness works the best

Top down



- How to select where to cut the cluster hierarchy?
- How to decide if a given clustering is a good one?

Modularity

- [Newman and Girvan, 2004] suggested notion of *modularity*
- Given a clustering of G
- Let E be a cluster×cluster ($k \times k$) matrix, where
- E_{ij} is the fraction of edges from cluster i to cluster j, and
- $A_i = \sum_j E_{ij}$
- Define modularity as

$$Q = \sum_i (E_{ii} - A_i^2) = \mathrm{Tr}(E) - ||E^2||$$

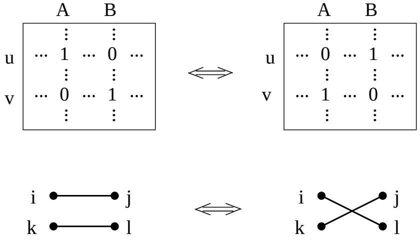
- Values:
  0 random structure, 1 strong community structure,
  typical [0.3..0.7], can be negative, too
- Q measure is not monotone with k

Optimizing modularity

- [Newman, 2003a] proposed an *agglomerative* algorithm for optimizing modularity directly
- [White and Smyth, 2005] proposed two *spectral* algorithms
- Comparable results, but spectral is much faster
- Still not scalable
- Can we do better? Faster algorithms? Approximation guarantees?
- Maximizing modularity is **NP**-hard [Brandes et al., 2006]

Modularity and swap randomization

- Assessing results of data mining algorithms via swap randomization [Gionis et al., 2006]
- Compare the result of a data mining algorithm on data $\mathcal{D}$ with the result obtained by the same algorithm on data $\mathcal{D}'$ that has the same margins as $\mathcal{D}$



- Same idea used by [Milo et al., 2004] to find significant motifs in biological networks

Modularity and swap randomization

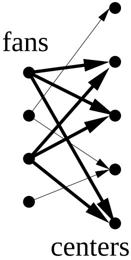
- Recall: $Q = \sum_i (E_{ii} - A_i^2)$,
  where E_{ij} is the fraction of edges from cluster i to cluster j,
  and $A_i = \sum_j E_{ij}$
- Appears to take account the total number of edges out of clusters, not the degrees of individual vertices
- Fix the degree of each vertex u to d_u
- Under independence, the probability of having an edge within cluster i is

$$\left(\sum_{u \in C_i} \frac{d_u}{2m}\right)\left(\sum_{v \in C_i} \frac{d_v}{2m}\right) = \left(\sum_{u \in C_i} \frac{d_u}{2m}\right)^2 = \left(\sum_j E_{ij}\right)^2 = A_i^2$$

Scaling up

- How to find communities on a large graph, say, the Web?
- *Web communities are characterized by dense directed bipartite graphs* [Kumar et al., 1999]
- Idea similar to *hubs* and *authorities*
- Example: Pages of sport cars (Lotus, Ferrari, Lamborghini) and enthusiastic fans
- *Bipartite cores:* Complete bipartite cliques contained in a community
- Support from random graph theory: If $G = (U, V, E)$ is a dense bipartite graph, then w.h.p. there is a $K_{i,j}$, for some i and j

Detecting communities by trawling



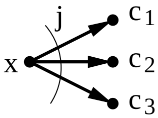
Many pruning phases

1. Heuristic pruning (quality consideration)
   - fans should point to at least 6 different hosts
   - canters should be pointed by at most 50 fans
2. Degree-based pruning
   - for a fan to participate in a $K_{i,j}$ it should have out-degree at least j
   - for a center to participate in a $K_{i,j}$ it should have in-degree at least i
   - prune iteratively fans and centers
   - can be done efficient by sorting edges
   - sort edges by src to prune fans
   - sort edges by dst to prune centers

Detecting communities by trawling

3. Inclusion-exclusion pruning

- either a core is output or a vertex is pruned



$$|N(c_1) \cap N(c_2) \cap N(c_3)| \geq i$$

- computation can be organized so that pruning is done with successive passes on the data

4. A-priori pruning

- cores satisfy monotonicity
- if (X, Y) is a $K_{i,j}$ then every (X', Y) with $X' \subseteq X$ a $K_{i',j}$
- a-priori algorithm: start with $(1, j)$, $(2, j)$, ...
- most computationally demanding phase, but the graph is already heavily pruned

Conclusions

- Finding communities in graphs:
- What is the right objective?
- Designing scalable algorithms is challenging
- How to evaluate the results?

Acknowledgments

The following people have contributed directly or indirectly to some of the content in this presentation

- Ricardo Baeza-Yates
- Carlos “Chato” Castillo
- Panayiotis Tsaparas
- . . .

Baeza-Yates, R. and Castillo, C. (2005).

Link analysis in national Web domains.

In Beigbeder, M. and Yee, W. G., editors, *Workshop on Open Source Web Information Retrieval (OSWIR)*, pages 15–18, Compiegne, France.

Bar-Yossef, Z., Jayram, T. S., Kumar, R., Sivakumar, D., and Trevisan, L. (2002).

Counting distinct elements in a data stream.

In *Proceedings of the 6th International Workshop on Randomization and Approximation Techniques (RANDOM)*, pages 1–10, Cambridge, Ma, USA. Springer-Verlag.

Barabási, A. L. and Albert, R. (1999).

Emergence of scaling in random networks.

Science, 286(5439):509–512.

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.

Buriol, L. S., Frahling, G., Leonardi, S., Marchetti-Spaccamela, A., and Sohler, C. (2006).
Counting triangles in data streams.
In *PODS '06: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 253–262, New York, NY, USA. ACM Press.

Cheng, D., Kannan, R., Vempala, S., and Wang, G. (2006).

A divide-and-merge methodology for clustering.

ACM Trans. Database Syst., 31(4):1499–1525.

Cohen, E. (1997).

Size-estimation framework with applications to transitive closure and reachability.

Journal of Computer and System Sciences, 55(3):441–453.

Flajolet, P. and Martin, N. G. (1985).

Probabilistic counting algorithms for data base applications.

Journal of Computer and System Sciences, 31(2):182–209.

Gionis, A., Mannila, H., Mielikäinen, T., and Tsaparas, P. (2006).

Assessing data mining results via swap randomization.

In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 167–176, New York, NY, USA. ACM Press.

Huberman, B. A. and Adamic, L. A. (1999).
Growth dynamics of the world-wide web.
Nature, 399.

Kannan, R., Vempala, S., and Vetta, A. (2004).
On clusterings: Good, bad and spectral.
J. ACM, 51(3):497–515.

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.

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

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

Trawling the Web for emerging cyber-communities.

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

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

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.

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

On power-law relationships of the internet topology.

In *SIGCOMM*.

Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M., and Alon, U. (2004).

Superfamilies of evolved and designed networks.

Science, 303(5663):1538–1542.

Mitzenmacher, M. (2004).

A brief history of generative models for power law and lognormal distributions.

Internet Mathematics, 1(2):226–251.

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

Fast algorithm for detecting community structure in networks.

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

The structure and function of complex networks.

Newman, M. E. J. and Girvan, M. (2004).

Finding and evaluating community structure in networks.

Physical Review E, 69(2).

Palmer, C. R., Gibbons, P. B., and Faloutsos, C. (2002).

ANF: a fast and scalable tool for data mining in massive graphs.

In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 81–90, New York, NY, USA. ACM Press.

Simon, H. A. (1955).

On a class of skew distribution functions.

Biometrika, 42(3/4):425.

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

A spectral clustering approach to finding communities in graph.

In *SDM*.

Yule, G. U. (1925).

A mathematical theory of evolution based on the conclusions of Dr. J. C. Willis.

Philosophical transactions of the Royal Society of London, 213:21–87.