CMSC5733 Social Computing

04-Graph Mining

Irwin King

The Chinese University of Hong Kong

king@cse.cuhk.edu.hk

Outline

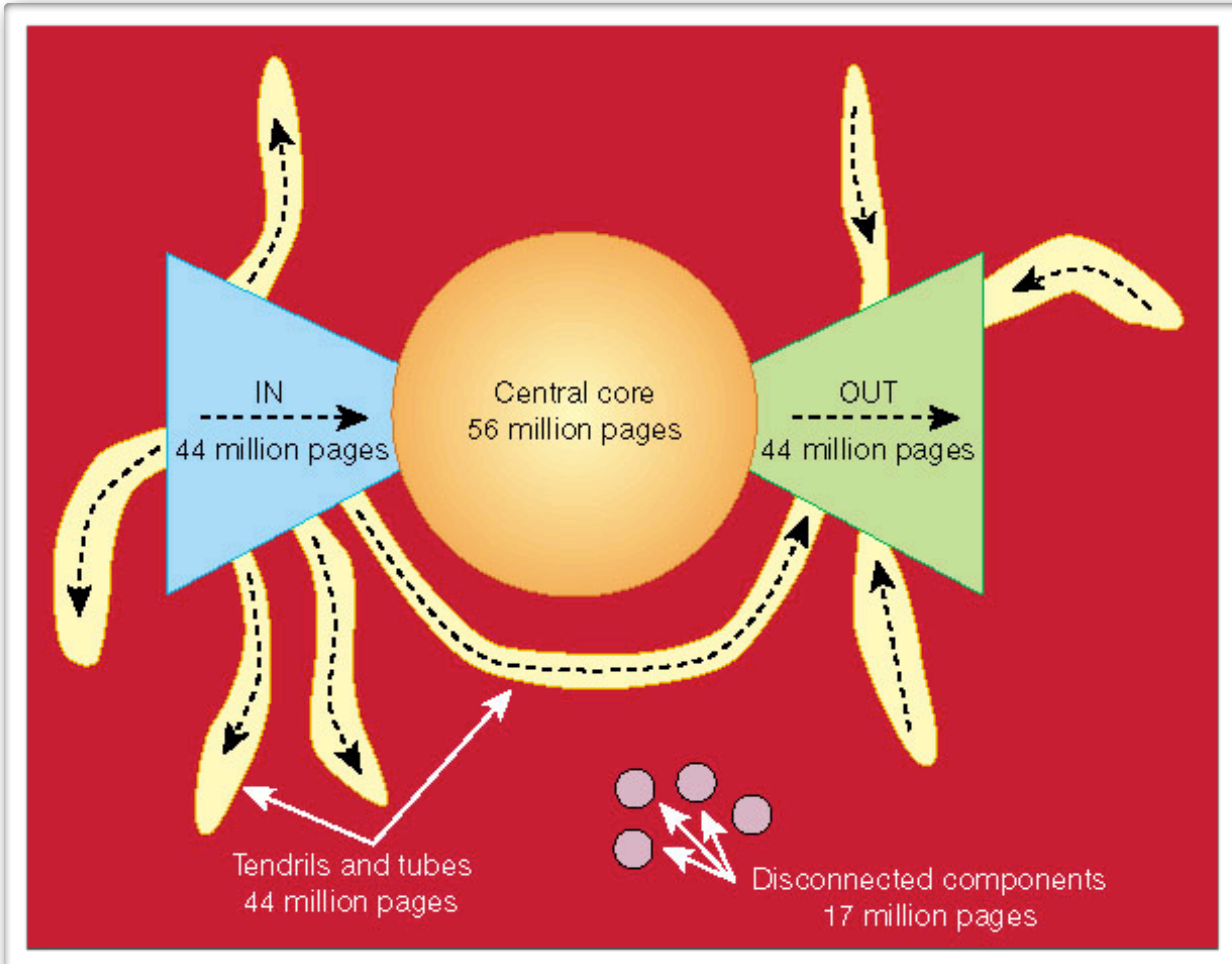
- Graph Characteristics, Patterns, and Structures
- Graph Generation & Information Propagation
- Graph Mining Algorithms

Graph Structures

Graph Patterns

- What are the characteristics of graphs?
- How can we compare graphs?
- What patterns hold for these graphs?
    - Power laws
    - Small diameters
    - Community effects
- How does the Internet graph look like?

What Does the Web Look Like?



- Recursive bowtie structure
- Ease of navigation
- Resilience

Introduction

- Graph mining is simply extraction of information from a massive graph
  - How does any network look like? The visualization of the relationship. One example is to look into how does the Internet or web look like.
  - Once we can characterize something, then we may be able to explore what is unique, abnormal, etc.
  - Are there any characteristics/principles/laws that hold?

Graph Distributions

- Two variables x and y are related by a power law when their scatter plot is linear on a log-log scale:

$$y(x) = cx^{-\gamma} \tag{1}$$

  where c and γ are positive constants.

- The constant γ is often called the **power law exponent**.

- **Power Law Distribution**. A random variable is distributed according to a power law when the probability density function (pdf) is given by

$$p(x) = cx^{-\gamma}, \gamma > 1, x \geq x_{\min} \tag{2}$$

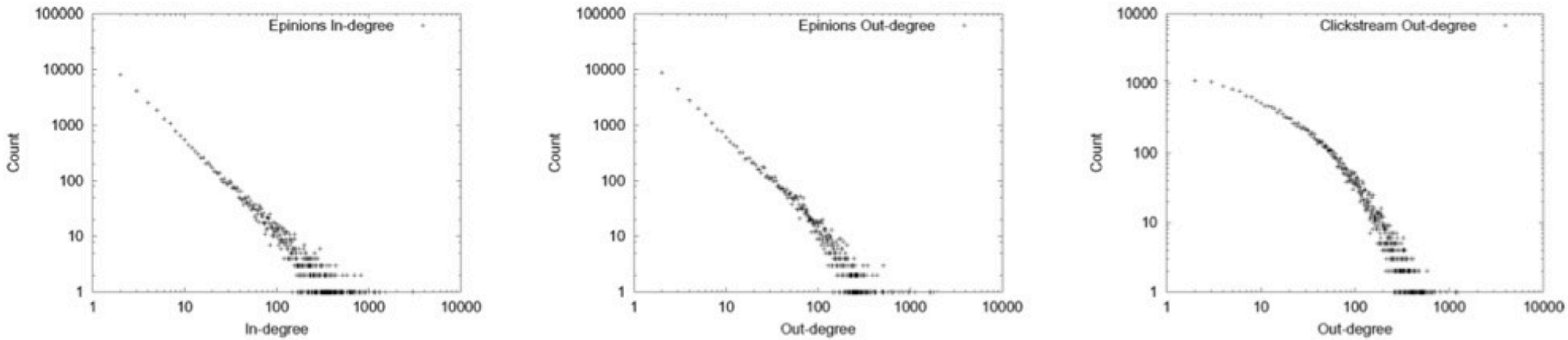
- $\gamma > 1$ ensures that $p(x)$ can be normalized.

- It is unusual to find $\gamma < 1$ in nature.

Degree Distribution

- The **Degree Distribution** of an undirected graph is a plot of the count c_k of nodes with degree k, versus the degree k, typically on a log-log scale.
- Occasionally, the fraction $\frac{c_k}{N}$ is used instead of c_k; however, this merely translates the log-log plot downwards.
- For directed graphs, out-degree and in-degree distributions are defined separately.
- Computational issues:
  1. Creating the scatter plot
  2. Computing the power law exponent
     – Regression models, maximum-likelihood estimation(MLE), non-parametric estimators, etc.
  3. Checking for goodness of fit
     – Correlation coefficient, statistical hypothesis methods, etc.

Examples of Power Law



(a) Epinions In-degree (b) Epinions Out-degree (c) Clickstream Out-degree

- Internet graph (2.1-2.2), Internet router (2.48), in-degree (2.1) and out-degree (2.38-2.72) of the WWW graph, PageRank, citation graph (3), etc.
- Power Law distributions are *heavy-tailed* so they decay more slowly than Gaussian distributions with exponential decay!

Other Distributions

- **Exponential Cutoffs**. Looks like power law over the lower range of values, but decays very fast for higher values. It is defined as,

$$y(x = k) \propto e^{-k/\kappa} k^{-\gamma}$$

  where $e^{-k/\kappa}$ is the exponential cutoff term, and $k^{-\gamma}$ is the power law term.

- The airport network, electric power grid of Souther California are examples of the exponential cutoffs distribution.

- **Longnormals**. Sometimes subsets of a power law graph can deviate significantly. It looks like a truncated parabolas on log-log scale.

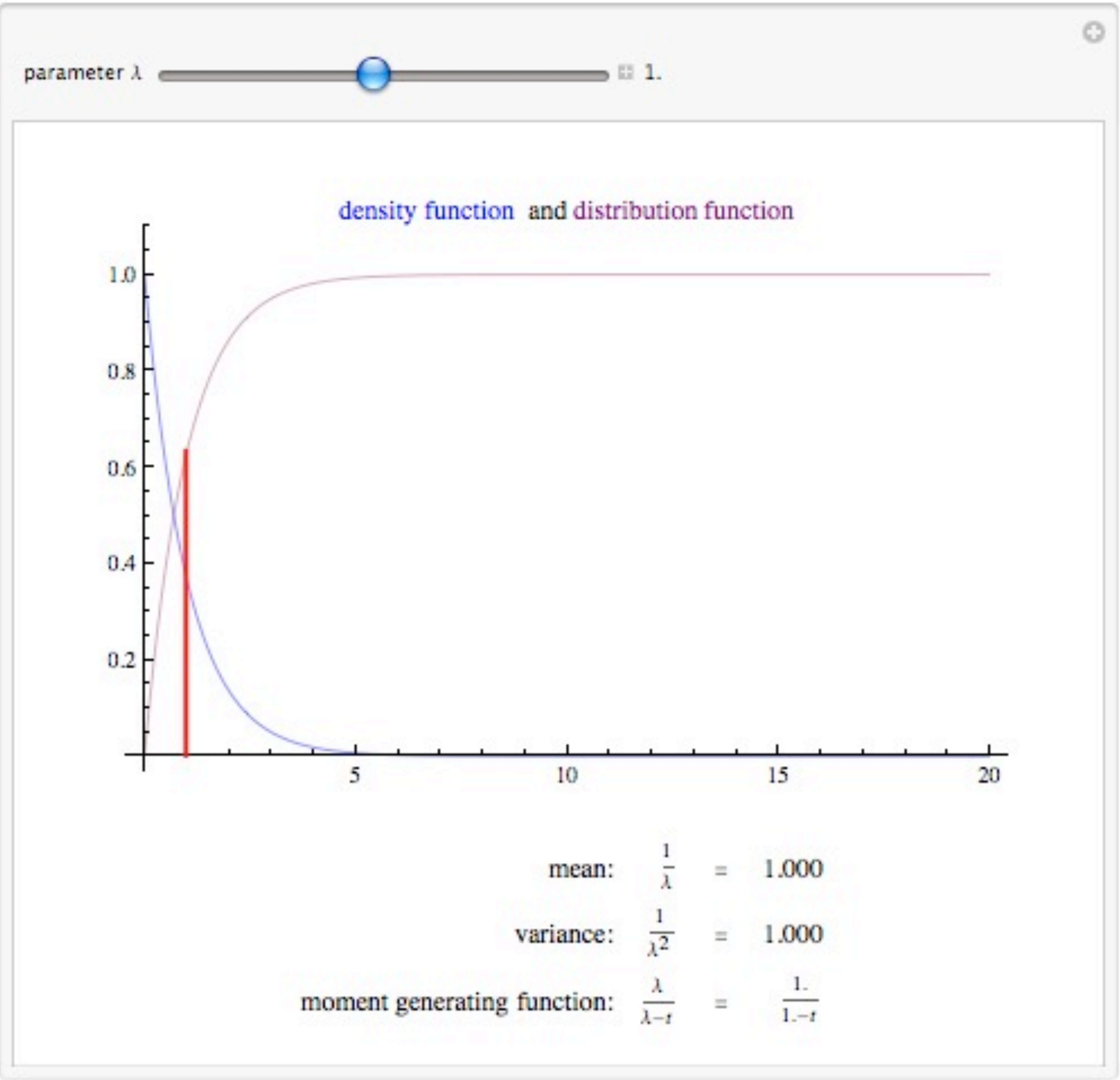
- It has unimodal distributions on the log-log scale and a discrete truncated lognormal (Discrete Gaussian Exponential, DGX) has a good fit.

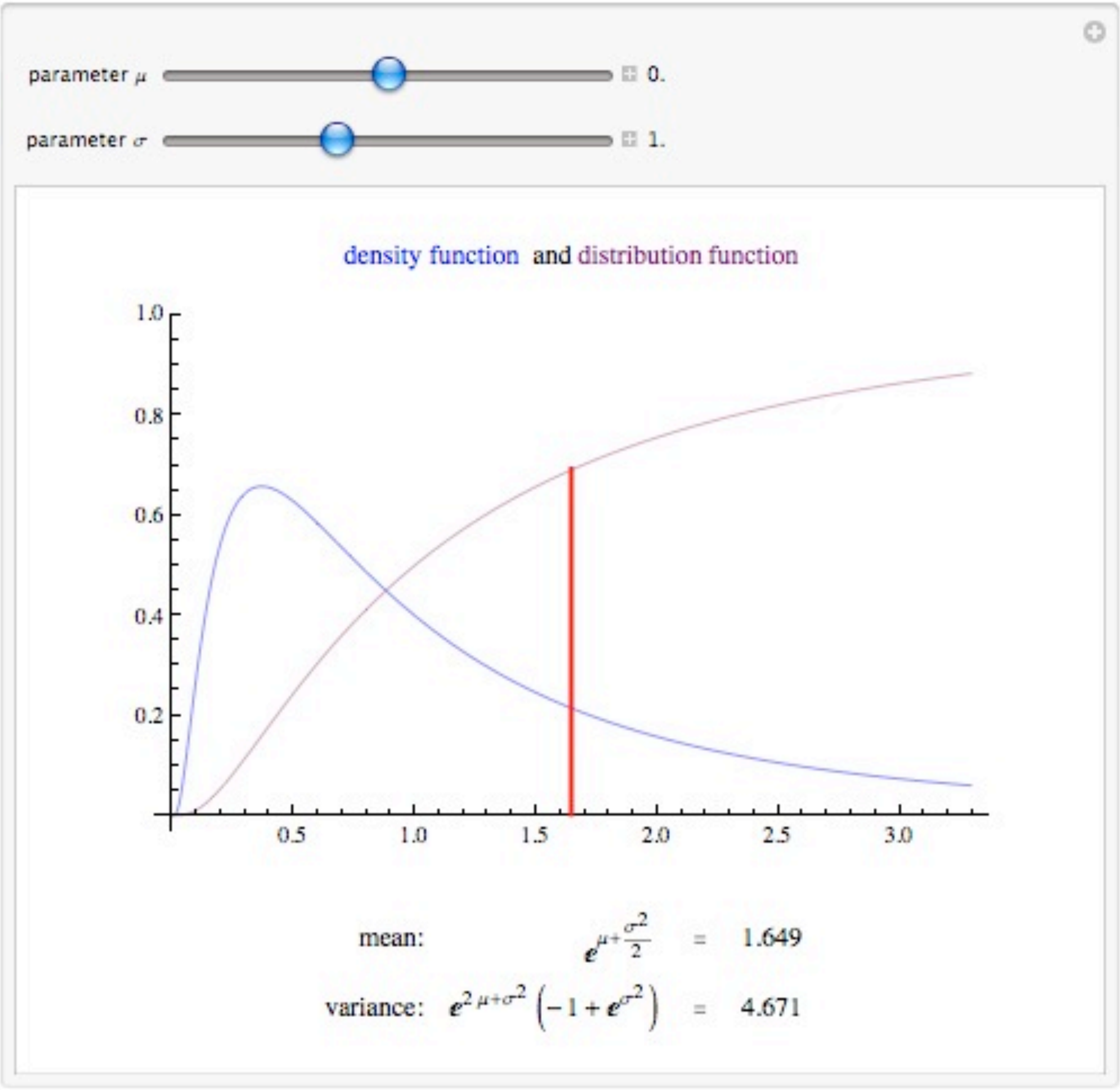
$$y(x = k) = \frac{A(\mu, \sigma)}{k} \exp\left[-\frac{(\ln k - \mu)^2}{2\sigma^2}\right], k = 1, 2, \ldots,$$

  where μ and σ are parameters and $A(\mu, \sigma)$ is a constant.

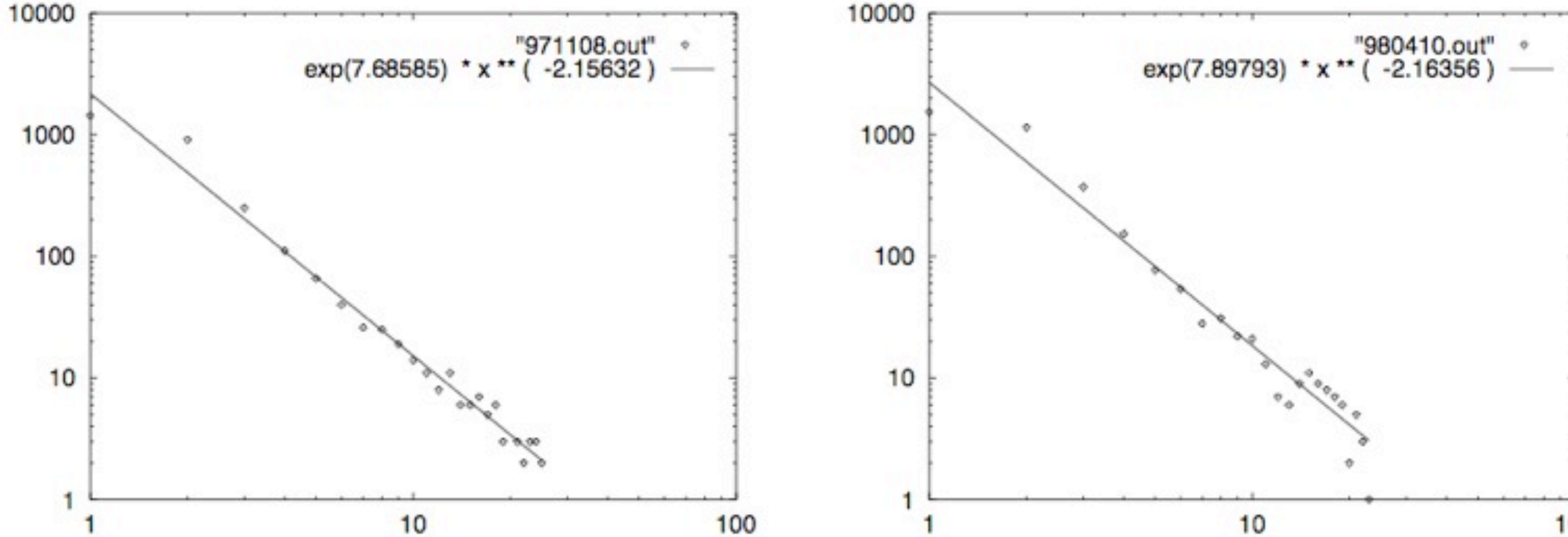
- The topic-based subsets of the WWW, Web clickstream data, sales data in retail chains, file size distributions, and phone usages are some examples of the Longnormals distribution.

Some Examples



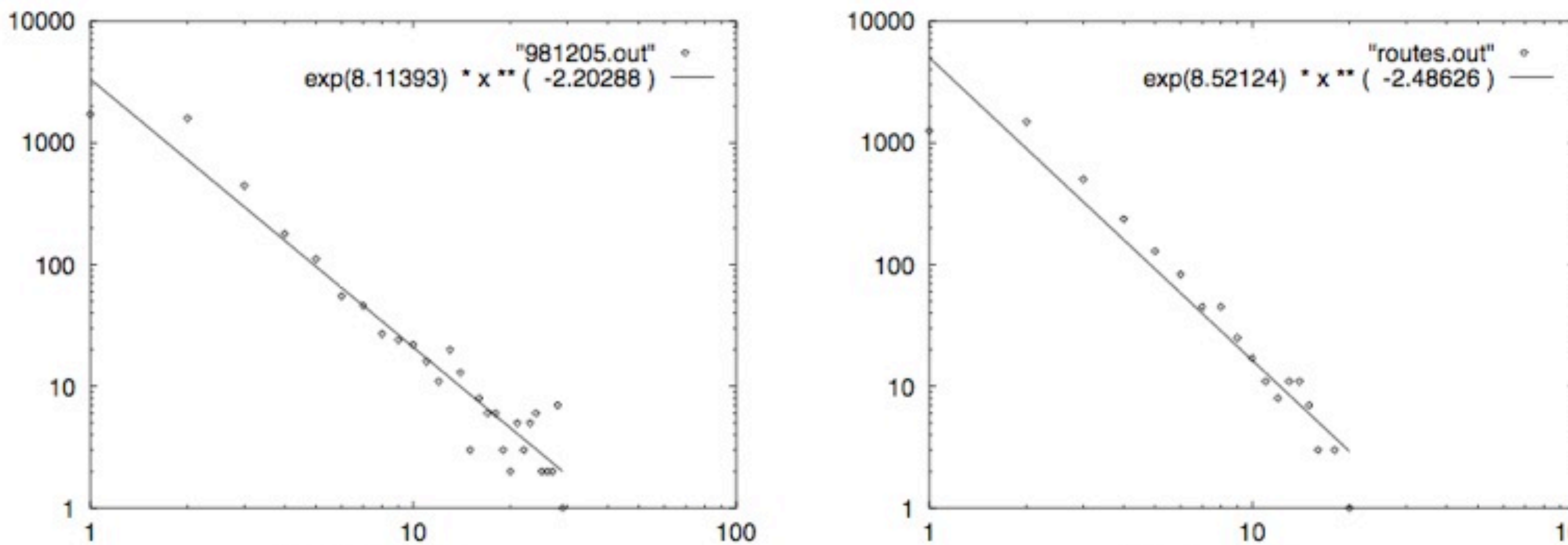


Outdegree Plots



(a) Int-11-97 (b) Int-04-98

Figure 5: The outdegree plots: Log-log plot of frequency f_d versus the outdegree d.



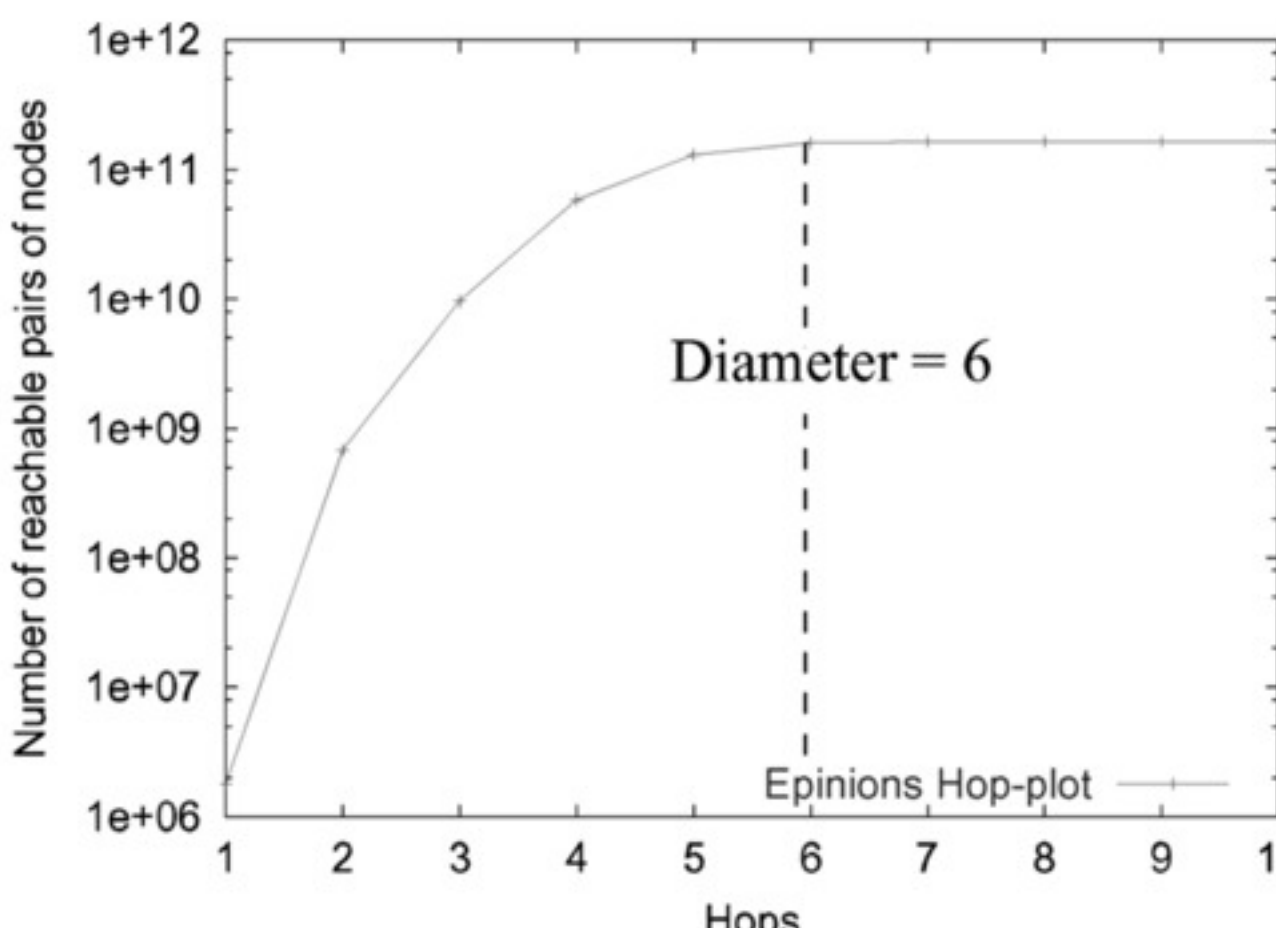
(a) Int-12-98 (b) Rout-95

The Hop Plot

- The **Hop-plot** is the plot of N_h versus h, where $N_h = \sum_u N_h(u)$, u is a node in the graph and $N_h(u)$ is the number of nodes in a neighborhood of h hops.

- The hop-plot can be used to calculate the *effective diameter* (or the eccentricity) of the graph.

- The effective diameter is defined as the minimum number of hops in which some fraction of all connected pairs of nodes can reach each other.



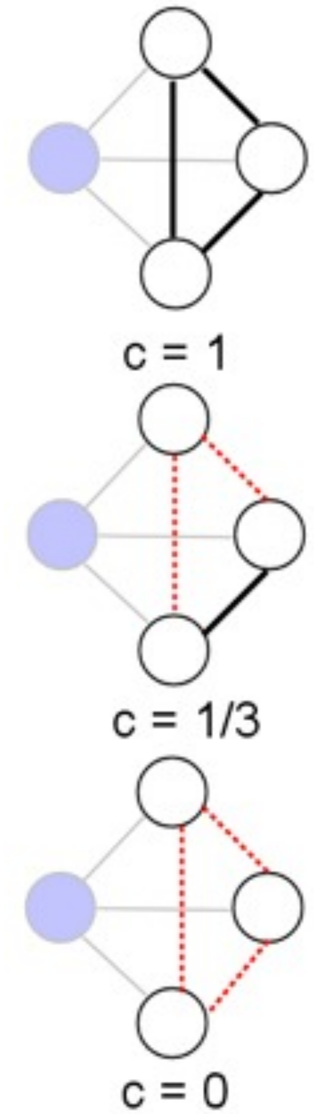
Clustering Coefficient

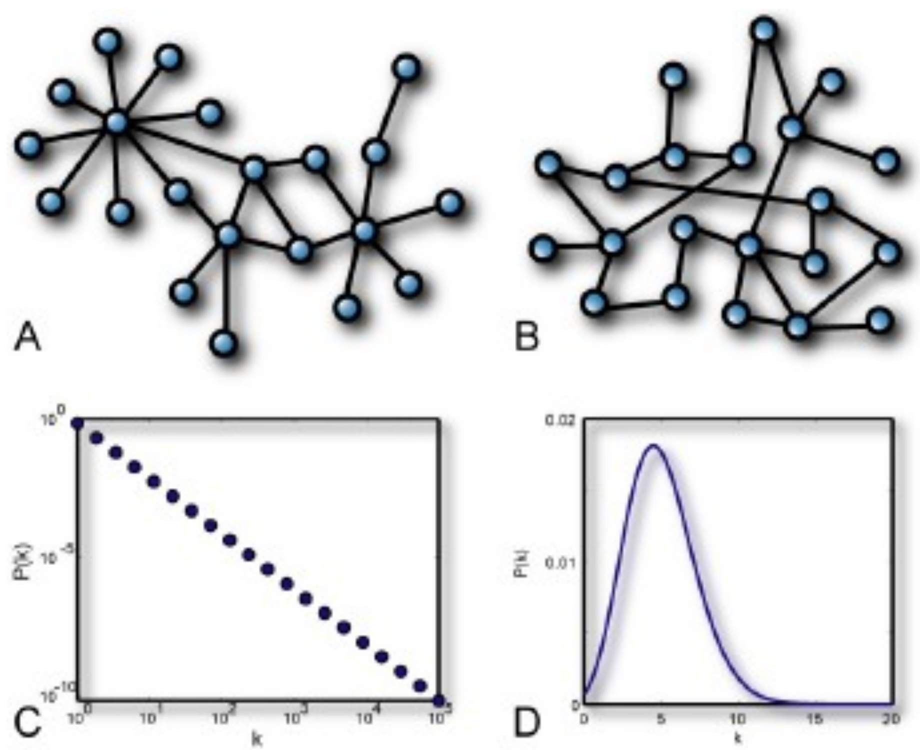
- **Clustering Coefficient**. Given that a node i has k_i neighbors, and there are n_i edges between the neighbors. The clustering coefficient of node i is defined as

$$C_i = \begin{cases} \frac{2n_i}{k_i(k_i-1)} & k_i > 1 \\ 0 & k_i = 0 \text{ or } 1 \end{cases} .$$

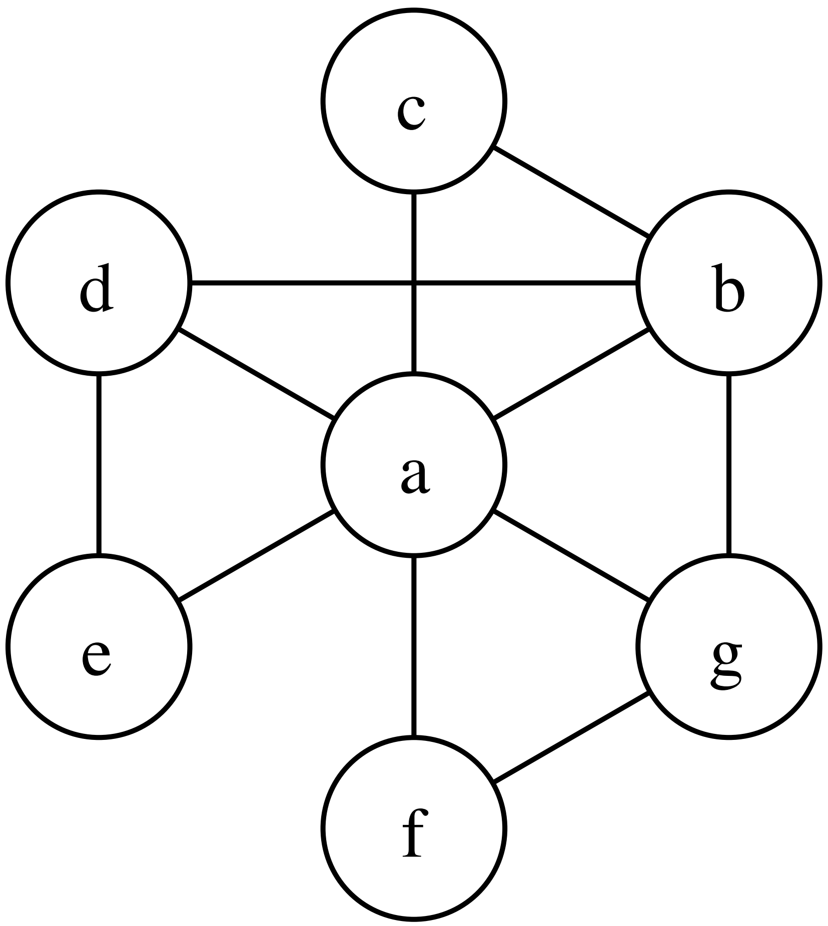
- For a node v with edges (u, v) and (v, w), the **Clustering Coefficient** of v measures the probability of existence of the third edge (u, w).

- The clustering coefficient of the entire graph (Global clustering coefficient) is found by averaging over all nodes in the graph.





An Example of Clustering Coefficient



Node a has 6 neighbors.

These neighbors could have been connected by 15 edges (6 x 5 / 2).

But with only 5 edges ({(c,b), (b,g), (g,f), (d,e), (d,b)}) exist so the local clustering coefficient of node a is 5/15 = 1/3

What is the global clustering coefficient?

Betweenness and Stress Plot

- **Betweenness** is a centrality measure of a vertex within a graph. It is defined as

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

  where σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(v)$ is the number of shortest paths from s to t that pass through a vertex v.

- One can also consider all shortest paths between all pairs of nodes in a graph. The edge betweenness or stress of an edge is the number of these shortest paths that the edge belongs to and is thus a measure of the "load" on that edge.

- **Stress Plot** is a plot of the number of edges s_k with stress k, versus k.

Graph Generation

Introduction

- Allows for simulation studies
- When is a generated graph realistic?
- Selected models
  - Random graph models
  - Preferential attachment models
  - Optimization-based models
  - Geographical models
  - Internet-specific models
  - BRITE, Inet, etc.

Random Graphs

- A **Random Graph** is a graph that is generated by some random process.
- A random graph is obtained by starting with a set of n vertices and adding edges between them at random.
- Different random graph models produce different probability distributions on graphs.
- The Erdös-Rényi model, denoted $G(n, p)$ generates a random graph by having every possible edge occurs independently with probability p.
- Another model, $G(n, M)$ assigns equal probability to all graphs with exactly M edges.
- In the $G(n, M)$ model, a graph is chosen uniformly at random from the collection of all graphs which have n nodes and M edges.
- In the $G(n, p)$ model, a graph is thought to be constructed by connecting nodes randomly. Each edge is included in the graph with probability p, with the presence or absence of any two distinct edges in the graph being independent.

Some Observations

1. If $np < 1$, then a graph in $G(n,p)$ will almost surely have no connected components of size larger than $O(\log n)$.
2. If $np = 1$, then a graph in $G(n,p)$ will almost surely have largest component whose size is of order $n^{2/3}$.
3. If np tends to a constant $c > 1$, then a graph in $G(n,p)$ will almost surely have a unique giant component containing a positive fraction of the vertices. No other component will contain more than $O(\log n)$ vertices.
4. If $p < \frac{(1-\epsilon)\ln n}{n}$, then a graph in $G(n,p)$ will almost surely contain isolated vertices.
5. If $p > \frac{(1+\epsilon)\ln n}{n}$, then a graph in $G(n,p)$ will almost surely have no isolated vertices.

Scale Free Networks

- A **scale-free network** is a network whose degree distribution follows a power law, at least asymptotically, i.e., the fraction $P(k)$ of nodes in the network having k connections to other nodes goes for larger values of k as $P(k) \sim k^{-\gamma}$ where γ is a constant whose value is typically in the range $2 < \gamma < 3$.
- Since it follows the power law, it decays only polynomially as $x \to \infty$, where as the Gaussian distribution has exponential decay.
- Moreover, $y(x)$ in the power law remains unchanged to within a multiplicative factor, i.e., $y(\alpha x) = \beta y(x))$, when x is multiplied by a scaling factor.
- The functional form of the relationship remains the same for all scales.

Preferential Attachment

- Rich get richer!
- A network grows by adding vertices over time.
- The average out-degree of the graph remains at a constant value over time
- Each outgoing edge from the new vertex connects to an old vertex with a probability proportional to the in-degree of the old vertex.

$$P(\text{edge to existing vertex } v) = \frac{k(v) + k_0}{\sum_i (k(i) + k_0)},$$

where $k(i)$ represents the current in-degree of an existing node i, and k_0 is a constant.

Barabasi-Albert Model

- Similar to the Preferential Attachment but for undirected graph
- Two processes
  - Growth--the network adds nodes and edges over time.
  - Preferential Attachment--the probability of connecting to a node is proportional to the current degree of the node.

$$P(\text{edge to existing vertex } v) = \frac{k(v)}{\sum_i k(i)},$$

where $k(i)$ is the degree of node i.

Geographical (Small-World) Models

- Many real-world graphs, e.g., job seeker, six-degrees, etc., seem to have
  - Low average distance between nodes (global property)
  - High clustering coefficients (local property)
- The low average path length was being caused by weak ties joining faraway cliques.

Small World Models

- Two processes:
  - Regular ring lattice (initial set-up)
    - Start with a ring lattice (N, k), a graph with N nodes set in a circle. Each node has k edges to its closest neighbors, with $k/2$ edges on each side.
  - Rewriting (creating weak acquaintance edges)
    - For each node u, each of its edges (u, v) is rewired with probability p to form some different edge (u, w), where node w is chosen uniformly at random.
    - Self-loops and duplicate edges are forbidden.

Information Propagation

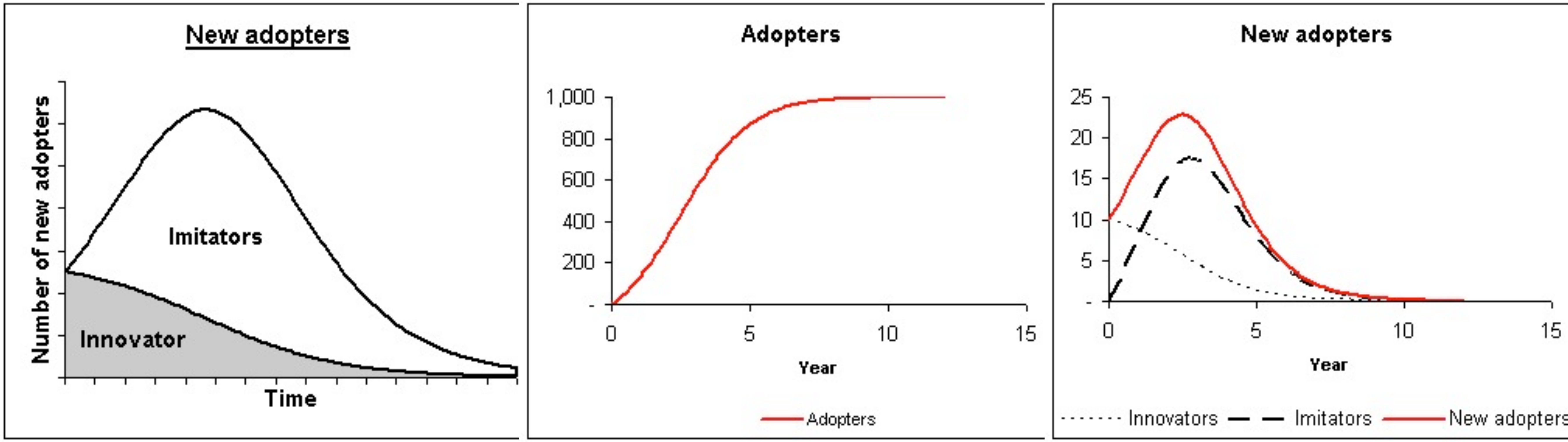
- Propagation Attributes
  - Propagation medium
  - Propagation rate
  - State of the node
  - Connectivity patterns
- Models
  - Threshold models
  - Viral propagation models
  - Diffusion models

Viral Propagation

- SIR Model
  - Susceptible (S), Infective (I), and Removed (R)
  - Each edge (i, j) has a spreading function (birth rate) β_{ij}
  - Each Infective node *u* has a rate of getting cured (death rate) δ_u
  - The spread of infections depends on $\tau = \beta / \delta$
- SIS Model
  - Similar to the SIR model except that once an infective node is cured, it goes back to the susceptible state

Bass Diffusion Model

- The process of how new products get adopted as an interaction between users and potential users



Bass Diffusion Formulation

The **Bass Diffusion Model** is defined as

$$\frac{f(t)}{1 - F(t)} = p + qF(t)$$

where

- $f(t)$ is the rate of change of the installed base fraction
- $F(t)$ is the installed base fraction
- p is the coefficient of innovation
- q is the coefficient of imitation

Sales $S(t)$ is the rate of change of installed base (i.e., adoption) $f(t)$ multiplied by the ultimate market potential m

$$\begin{aligned} S(t) &= mf(t) \\ S(t) &= m\frac{(p+q)^2}{p}\frac{e^{-(p+q)t}}{(1+\frac{q}{p}e^{-(p+q)t})^2} \end{aligned}$$

The time of peak sales t^* is defined as

$$t^* = \frac{\ln q - \ln p}{p + q}$$

Discussions

- Properties to consider
  - Degree distributions
  - Clustering coefficient
  - Community structure
  - Implementation issues
- How do you make friends?
- How can one recommend friends?
- How does information propagate among friends?

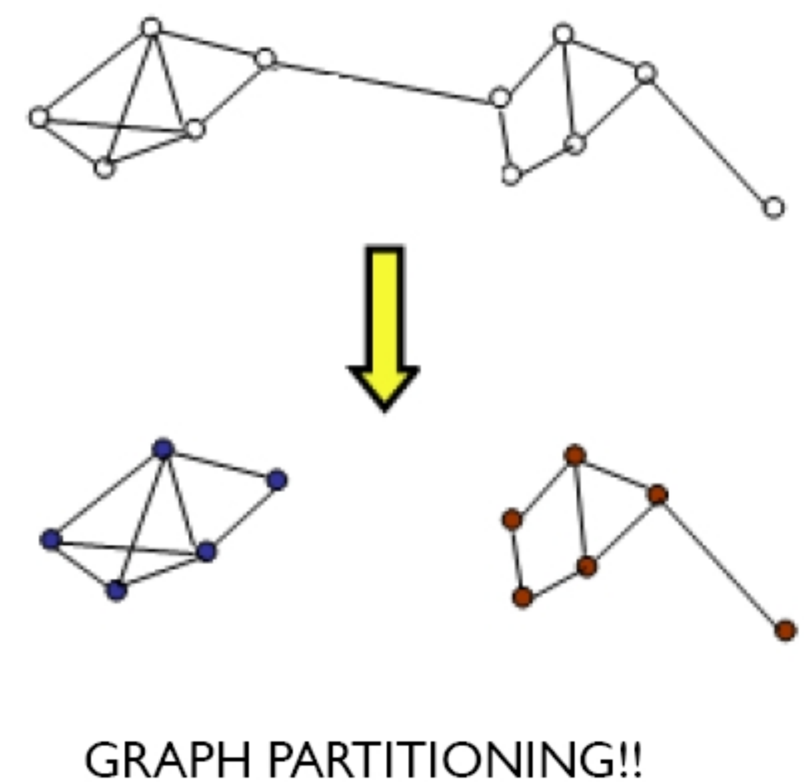
Graph Mining

Clustering

- Finding patterns in data, or grouping similar groups of data-points together into clusters.
- Clustering algorithms for numeric data
  - Lloyd's K-means, EM clustering, spectral clustering etc.
- Traditional definition of a "good" clustering
  - Points assigned to same cluster should be highly similar
  - Points assigned to different clusters should be highly dissimilar

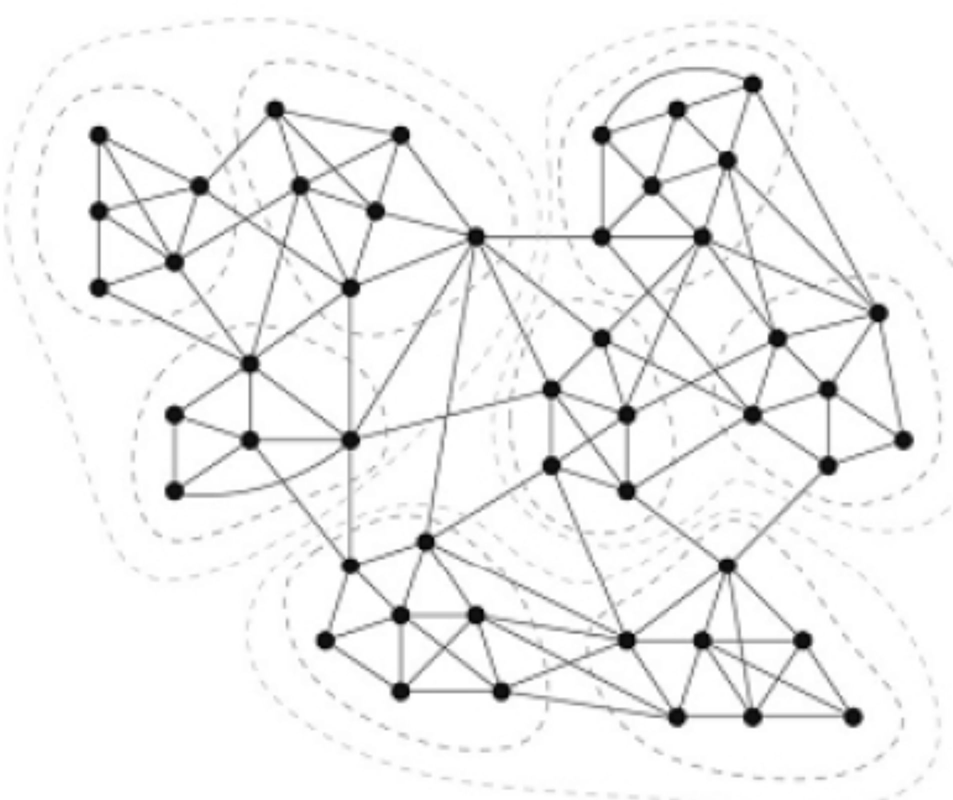
Graph Clustering

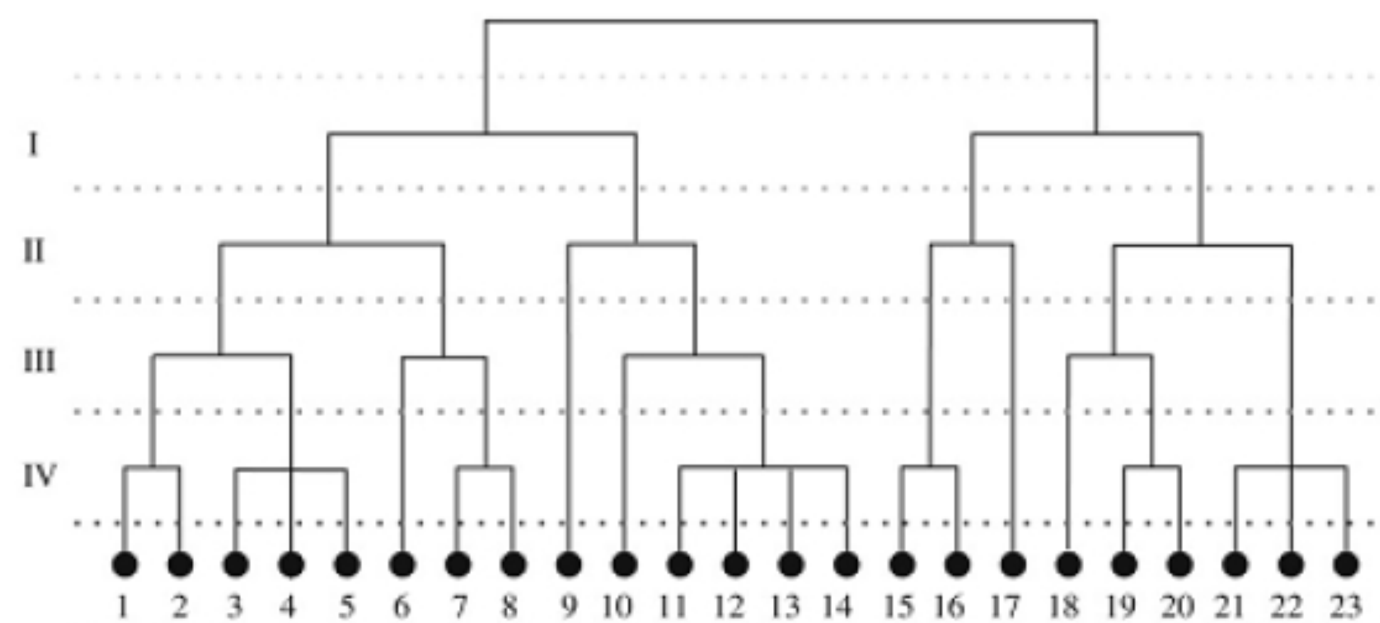
- Graphical representation of data as undirected graphs
- Clustering of vertices on basis of edge structure
- Defining a graph cluster
  - In its loosest sense, a graph cluster is a connected component
  - In its strictest sense, it's a maximal clique of a graph
- Many vertices within each cluster
- Few edges between clusters



Clustering Paradigm

- Hierarchical clustering vs. flat clustering
- Hierarchical:
  - Top down
  - Bottom up



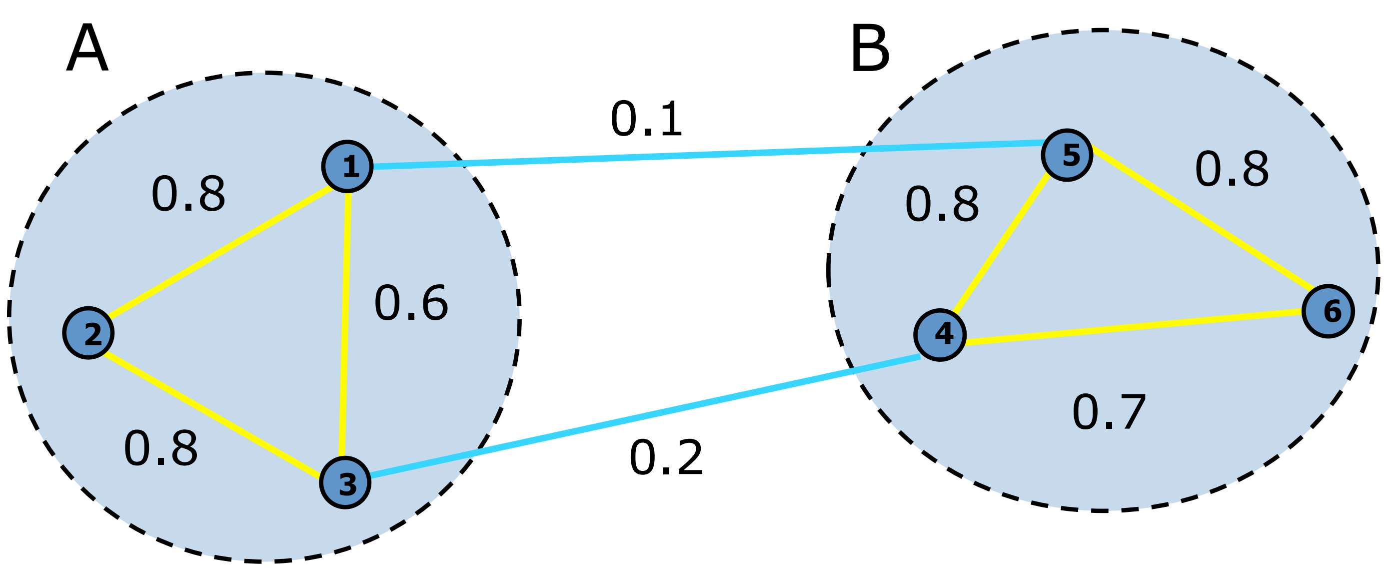


Overview

- Cut based methods
  - Become NP hard with introduction of size constraints
  - Approximation algorithms minimizing graph conductance
- Maximum flow
  - Using results by Golberg and Tarjan
  - Reasonable for small graphs
- Graph spectrum based methods
  - Stable perturbation analysis
  - Good even when graph is not exactly block diagonal
  - Typically, second smallest eigenvalue is taken as graph characteristic
  - Spectrum of graph transition matrix for blind walk

Graph Cuts

- Express partitioning objectives as a function of the "edge cut" of the partition
- *Cut*: Set of edges with only one vertex in a group.
  - We want to find the minimal cut between groups
  - The group that has the minimal cut would be the partition



$$cut(A,B) = \sum_{i \in A, j \in B} w_{ij}$$

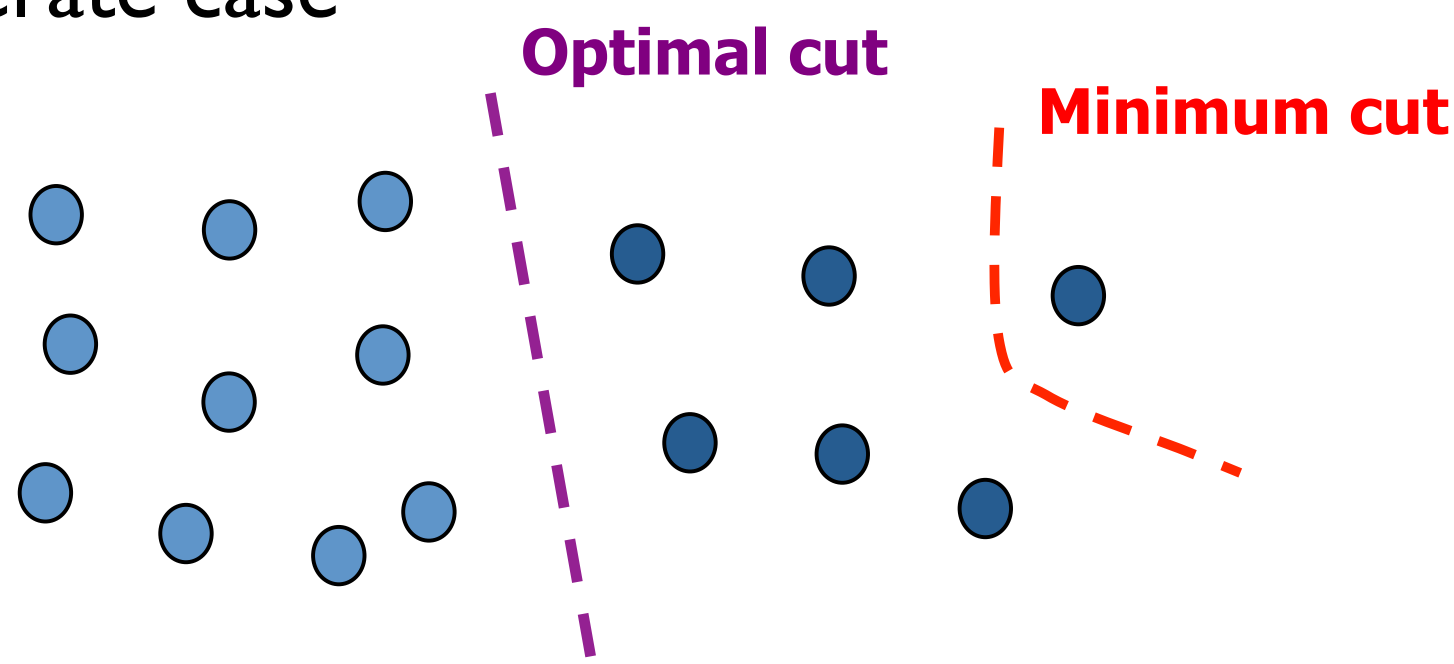
$$cut(A,B) = 0.3$$

Graph Cut Criteria

- Criterion: Minimum-cut
  - Minimize weight of connections between groups

$$\min cut(A, B)$$

- Degenerate case



- Issues
  - Only considers external cluster connections
  - Does not consider internal cluster density

Graph Cut Criteria

- Criterion: Normalized-cut [Shi & Malik,'97]
  - Consider the connectivity between groups relative to the density of each group

$$\min Ncut(A,B) = \frac{cut(A,B)}{vol(A)} + \frac{cut(A,B)}{vol(B)}$$

  - Normalize the association between groups by volume
    - *vol*(*A*): The total weight of the edges originating from group *A*
- Why use this criterion?
  - Minimizing the normalized cut is equivalent to maximizing normalized association
  - Produce more balanced partitions

Summary

- Clustering as a graph partitioning problem
  - Quality of a partition can be determined using graph cut criteria
  - Identifying an optimal partition is NP-hard
- Spectral clustering techniques
  - Efficient approach to calculate near-optimal bi-partitions and *k*-way partitions
  - Based on well-known cut criteria and strong theoretical background

Graph Cuts and Max-Flow/Min-Cut Algorithms

- A **flow network** is defined as a directed graph where an edge has a nonnegative capacity
- A **flow** in *G* is a real-valued (often integer) function that satisfies the following three properties:
  - Capacity Constraint:
    - For all $u, v \in V, f(u,v) \leq c(u,v)$
  - Skew Symmetry
    - For all $u, v \in V, f(u,v) = -f(v,u)$
  - Flow Conservation
    - For all $u \in (V \backslash \{s,t\}), \sum_{v \in V} f(u,v) = 0$

How to Find the Minimum Cut?

- Theorem: In graph *G*, the maximum source-to-sink flow possible is equal to the capacity of the minimum cut in *G*

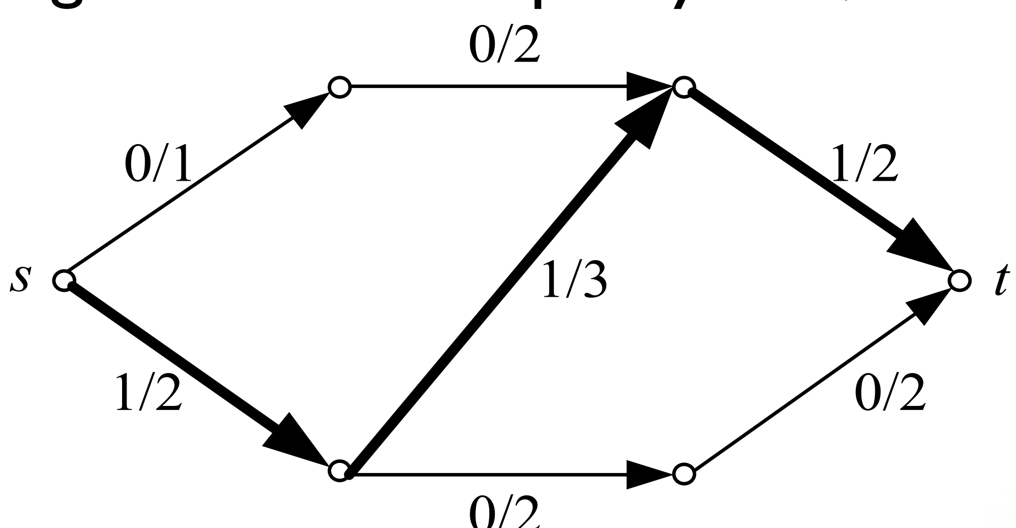
  [L. R. Foulds, Graph Theory Applications, 1992 Springer-Verlag New York Inc., 247-248]

Maximum Flow and Minimum Cut Problem

- Some basic concepts
  - If *f* is a flow, then the net flow across the cut (*S*, *T*) is defined to be *f*(*S*, *T*), which is the sum of all edge capacities from *S* to *T* subtracted by the sum of all edge capacities from *T* to *S*
  - The capacity of the cut (*S*, *T*) is *c*(*S*, *T*), which is the sum of the capacities of all edge from *S* to *T*
  - A minimum cut is a cut whose capacity is the minimum over all cuts of *G*
- Algorithms
  - Ford-Fulkerson Algorithm
  - Push-Relabel Algorithm
  - New Algorithm by Boykov, etc.

Ford-Fulkerson Algorithm

- Main Operation
  - Starting from zero flow, increase the flow gradually by finding a path from *s* to *t* along which more flow can be sent, until a max-flow is achieved
  - The path for flow to be pushed through is called an **augmenting path**
- The Ford-Fulkerson algorithm uses a residual network of flow in order to find the solution
- The residual network is defined as the network of edges containing flow that has already been sent
- For example, in the graph shown below, there is an initial path from the source to the sink, and the middle edge has a total capacity of 3, and a residual capacity of 3-1=2



-

Ford-Fulkerson Algorithm

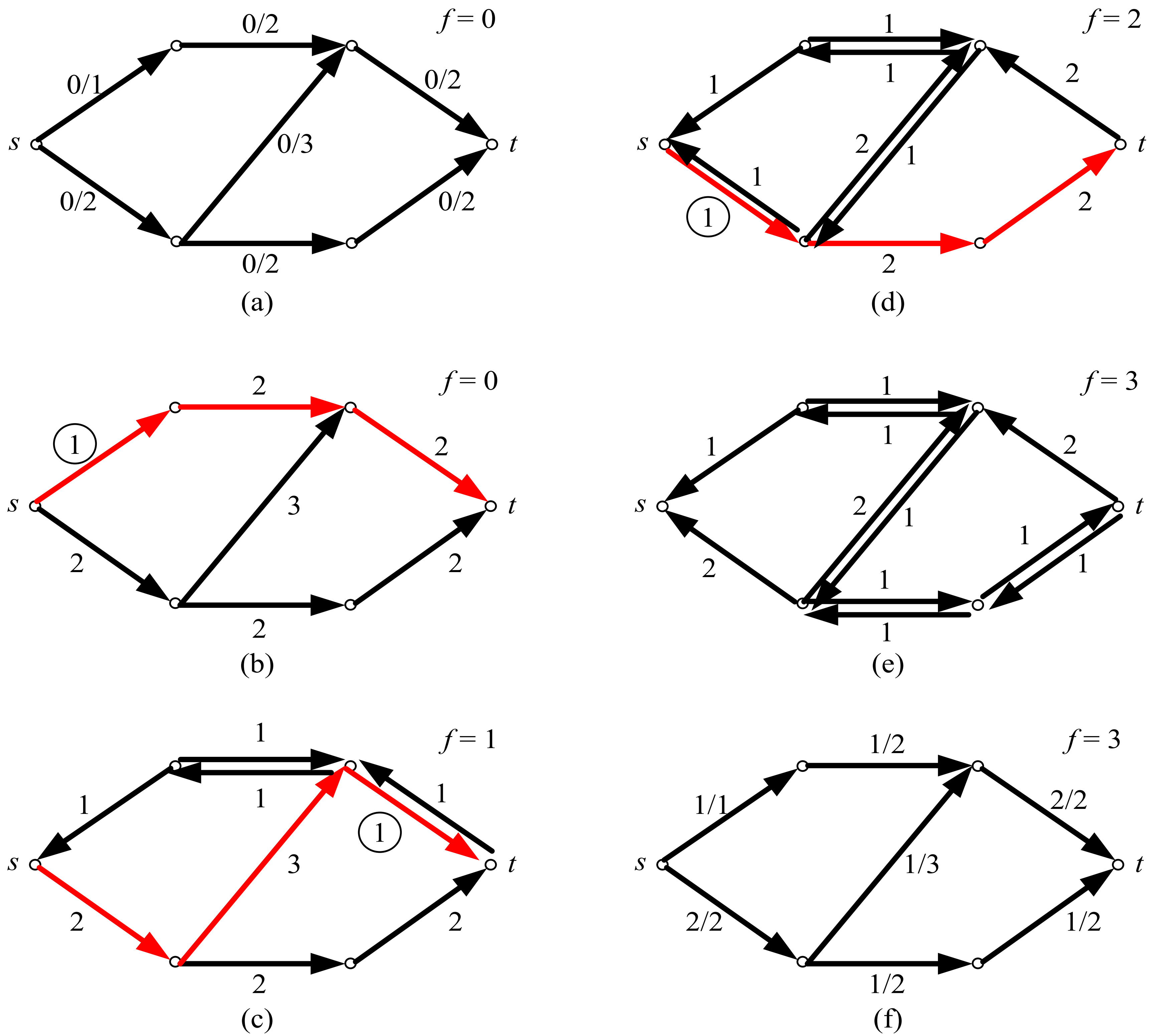
- Assuming there are two vertices, u and v, let $f(u,v)$ denote the flow between them, $c(u,v)$ be the total capacity, $c_f(u,v)$ be the residual capacity, and there should be,

$$c_f(u,v) = c(u,v) - f(u,v)$$

- Given a flow network and a flow f, the residual network of G is $G_f = (V, E_f)$, where $E_f = \{(u,v) \in V \times V : c_f(u,v) > 0\}$
- Given a flow network and a flow f, an augmenting path P is a simple path from s to t in the residual network
- We call the maximum amount by which we can increase the flow on each edge in an augmenting path P the residual capacity of P, given by,

$$c_f(P) = \min\{c_f(u,v) : (u,v) \text{ is on } P\}$$

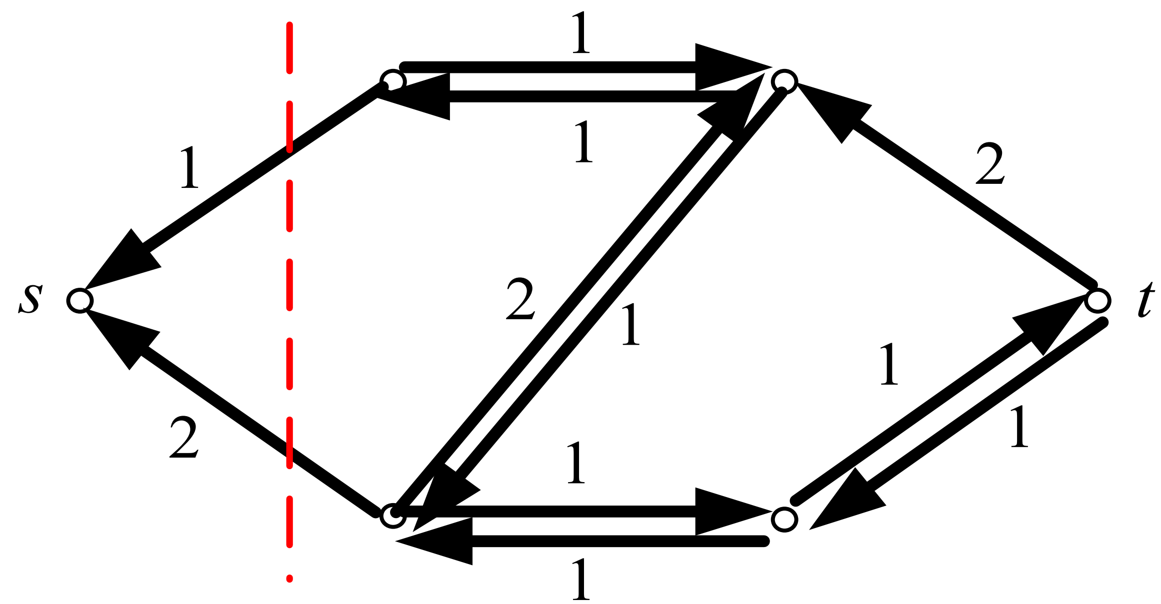
Example


f = 0
0/2
0/1
0/2
0/3
s
t
0/2
0/2
0/2
(a)
f = 2
1
1
1
2
2
1
1
2
2
2
(d)
f = 0
2
1
2
3
2
2
2
(b)
f = 3
1
1
1
2
2
1
1
2
1
1
1
1
(e)
f = 1
1
1
1
1
1
2
3
2
2
(c)
f = 3
1/2
1/1
2/2
1/3
2/2
1/2
1/2
(f)

Finding the Min-Cut

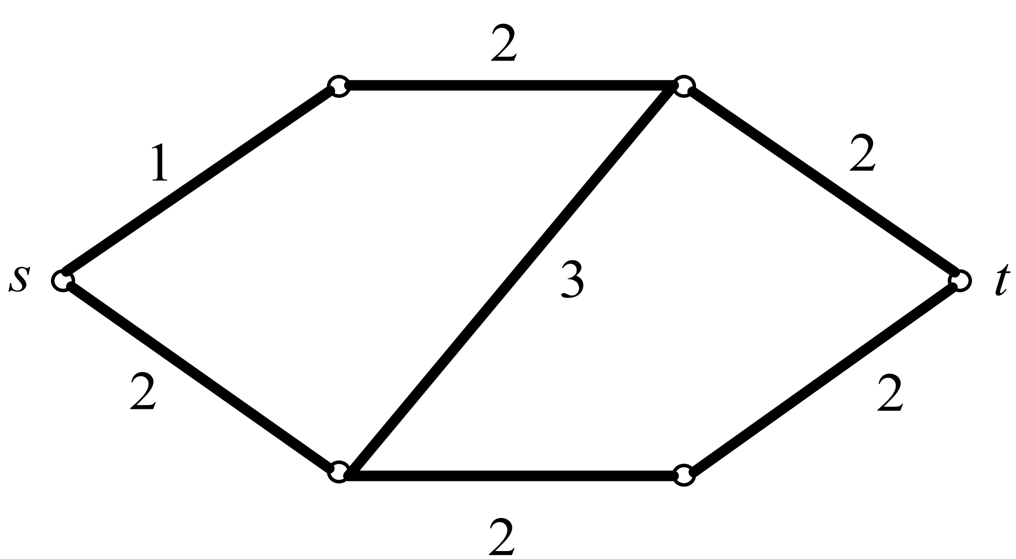
- After the max-flow is found, the minimum cut is determined by

$$\begin{aligned} S &= \{\text{All vertices reachable from } s\} \\ T &= G \backslash S \end{aligned}$$

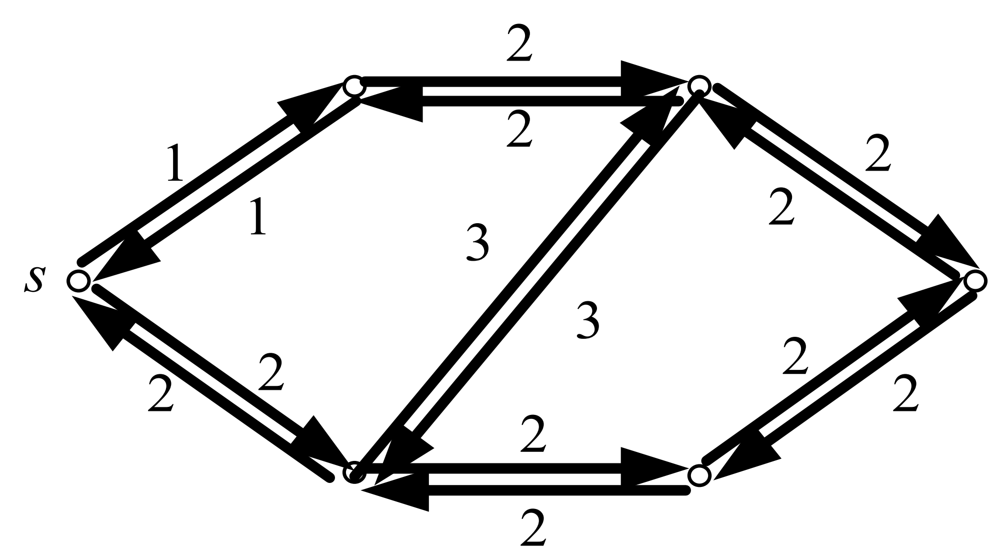


Special Case

- As in some applications only undirected graph is constructed, when we want to find the min-cut, we assign two edges with the same capacity to take the place of the original undirected edge



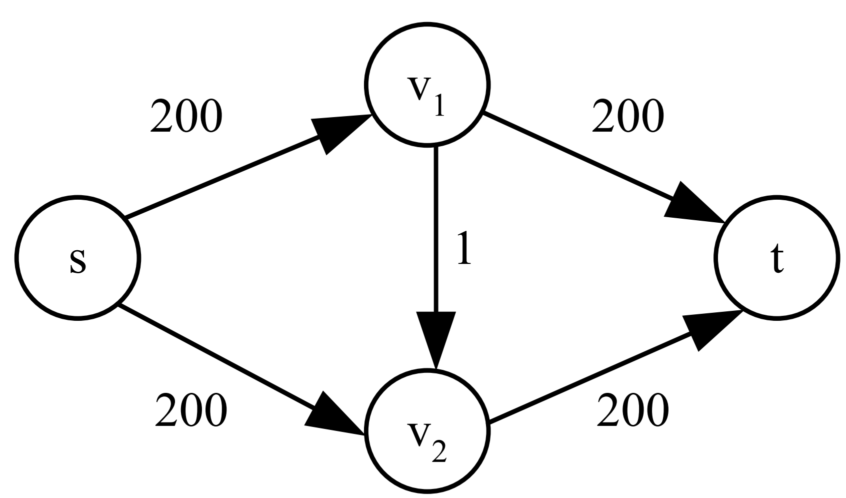
(a)



(b)

Ford-Fulkerson Algorithm Analysis

- The running time of the algorithm depends on how the augmenting path is determined
  - If the searching for augmenting path is realized by a breadth-first search, the algorithm runs in polynomial time of $O(E\ |f_{max}|\ )$
- Under some extreme cases the efficiency of the algorithm can be reduced drastically
  - One example is shown in the figure below, applying Ford-Fulkerson algorithm needs 400 iterations to get the max flow of 400



  -

References

- Deepayan Chakrabarti and Christos Faloutsos, Graph mining: Laws, generators, and algorithms, ACM Computing Surveys (CSUR) 38, 1, Article No. 2 (2006).
- Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Ra- jagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the Web. In Proc. 9th International World Wide Web Conference, pages 309–320, 2000.
- Wikipedia, NetworkX, etc.
- Rahul Bajaj and Manu Bansal, "Detection of communities in social networks"

CSCI5070 Advanced Topics in Social Computing

04-Link Analysis

Irwin King

The Chinese University of Hong Kong

king@cse.cuhk.edu.hk

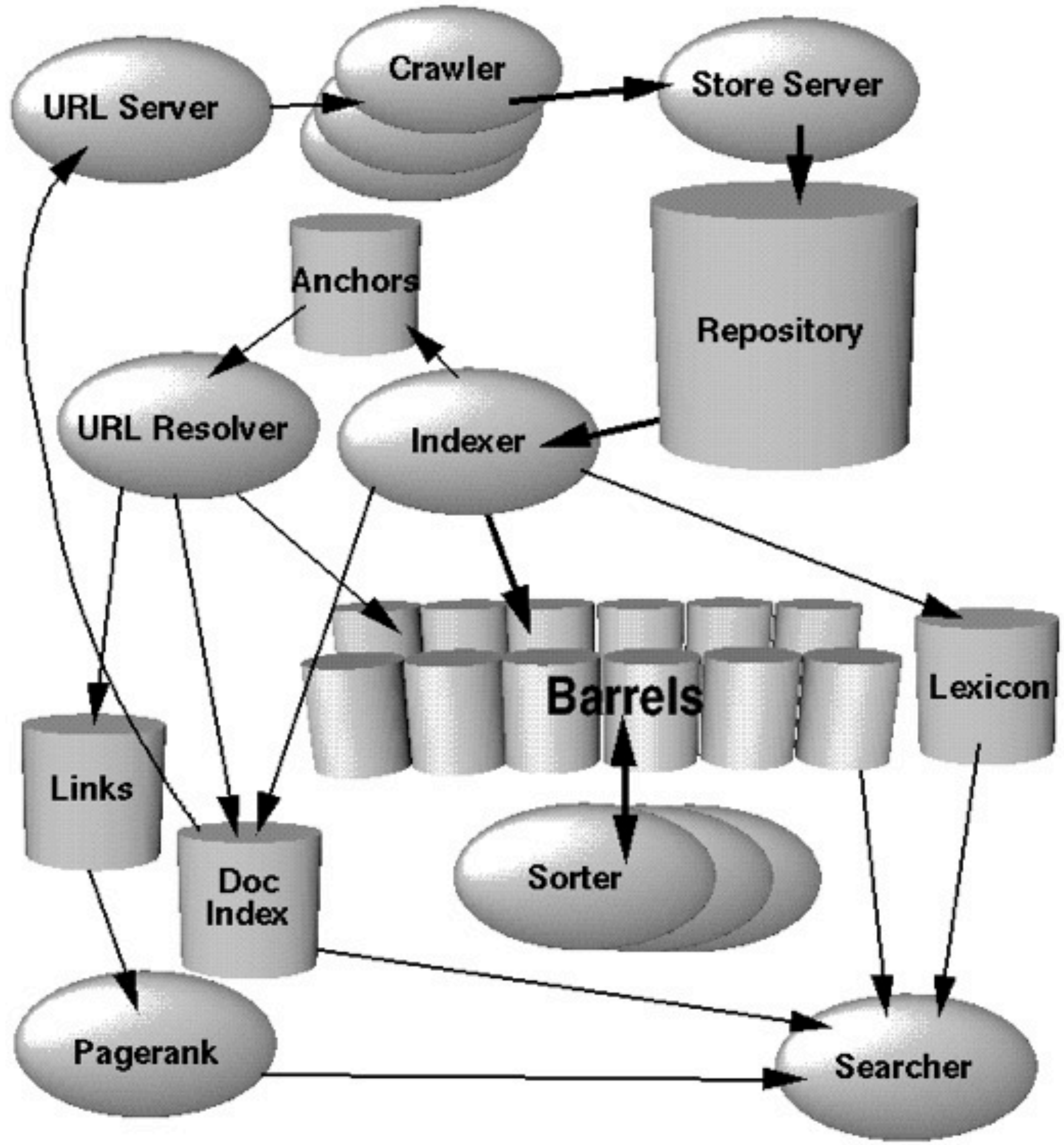
Traditional Information Retrieval

- Content matching against the query
  - Occurrence of query words
  - Location of query words
  - Document weighting
- Not much of ranking
- Science Citation Index and Impact Factor

Challenges of Web Search

- Voluminous
- Dynamic (generated deep web)
- Self-organized
- Hyperlinked
- Quality of Information
- Accessibility

Information Retrieval and Search Engine



Crawler

- Page Repository
- Indexing Module
- Indices

- Query Module
- Ranking Module

Information Retrieval Basics

- Vector Space Model
- Relevance Scoring and Relevance Feedback
- Meta-search Engines
- Precision vs. Recall

The InDegree Algorithm

- A simple heuristic
- Rank the pages according to popularity (indegree) of the page
- Issues?

The PageRank Algorithm

- Hyperlinked documents are different!
  - Similar to academic papers
  - In-links = authorities
  - Out-links = citations
  - Citations give better approximation of the quality of pages

Define PageRank

The PageRank calculation is defined as follows. We assume page A has pages $T_1, \cdots, T_n$ which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. $C(A)$ is defined as the number of links going out of page A. The PageRank of a page A is given as follows:

$$PR(A) = (1-d) + d(PR(T_1)/C(T_1) + \cdots + PR(T_n)/C(T_n)). \qquad (1)$$

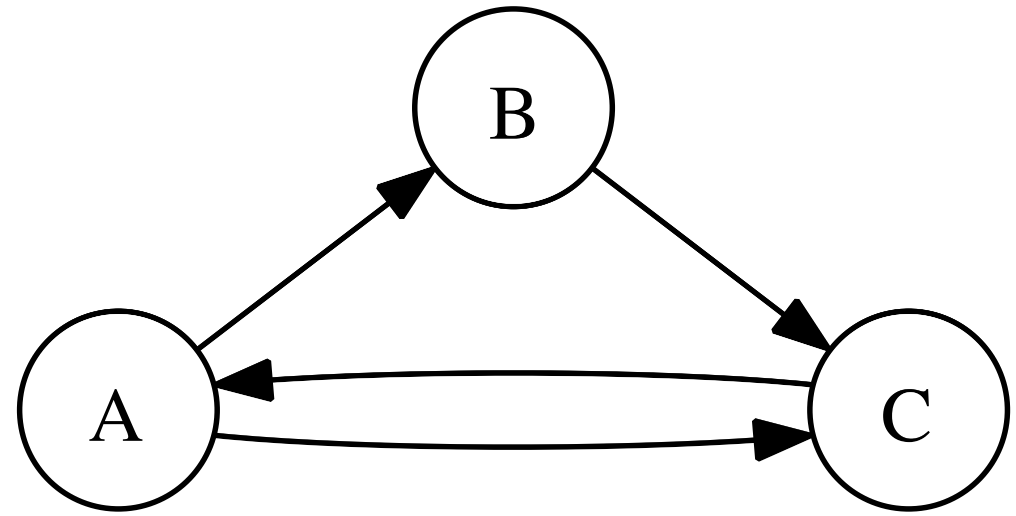
$$PR(A) = (1-d) + d\sum_{i}^{n} \frac{PR(T_i)}{C(T_i)}.$$

- PageRanks form a probability distribution over web pages, so the sum of all web pages' PageRanks will be one
- It can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web

Assumptions

- A "random surfer" who is given a web page at random
- The surfer keeps clicking on links, never hitting "back"
- The surfer gets bored and starts on another random page
- The probability that the random surfer visits a page is its PageRank
- The *d* damping factor is the probability at each page the Surfer will get bored and request another random page.
- Instead of a global *d*, one may consider a page damping factor d_i for each individual page or a group of pages

Examples

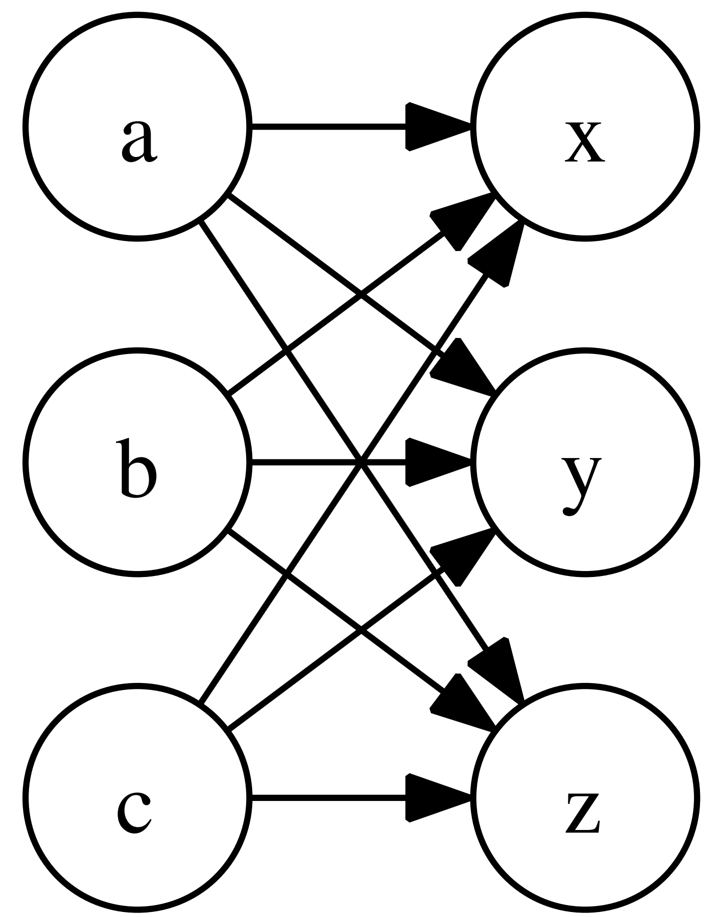


$$
\begin{aligned}
d &= 0.5 && (1)\\
PR(A) &= 0.5 + 0.5(PR(A)/2) && (2)\\
PR(C) &= 0.5 + 0.5(PR(A)/2 + PR(B)) && (3)
\end{aligned}
$$

$$
\begin{aligned}
PR(A) &= 14/13 = 1.07692308 && (4)\\
PR(B) &= 10/13 = 0.76923077 && (5)\\
PR(C) &= 15/13 = 1.15384615 && (6)
\end{aligned}
$$

Kleinberg's Algorithm

- Web page importance should depend on the search query being performed
- Each page should have a separate "authority" rating (based on the links going to the page) that captures the quality of the page as a resource itself
- Each page should also have a "hub" rating (based on the links going from the page) that captures the quality of the pages as a pointer to useful resources



Hubs Authorities

Define HITS Algorithm

- The HITS (Hyperlink Induced Topic Distillation) algorithm computes lists of hubs and authorities for WWW search topics
- Start with a search topic, specified by one or more query terms
    - Sampling Stage--constructs a focused collection of several thousand Web pages likely to be rich in relevant authorities
    - Weight-propagation Stage-- determines numerical estimates of hub and authority weights by an iterative procedure
- The pages with the highest weights are returned as hubs and authorities for the search topic

The HITS Algorithm

Let the Web be a digraph $G = (V, E)$. Given a subgraph $S \subseteq V$ with $u, v \in S$ and $(u, v) \in E$. The authority and hub weights are updated as follows.

1. If a page is pointed to by many good hubs, we would like to increase its authority weight.

$$x_p = \sum_{q \text{ such that } q \to p} y_q, \tag{1}$$

   where the notation $q \to p$ indicates taht q links to p.

2. If a page points to many good authorities, we increase its hub weight

$$y_p = \sum_{q \text{ such that } p \to q} x_q. \tag{2}$$

The above can be rewritten in a matrix notation as

$$x \leftarrow A^T y \leftarrow A^T A x = (A^T A)x \tag{3}$$

and

$$y \leftarrow Ax \leftarrow AA^T y = (AA^T)y \tag{4}$$

The HITS Pseudocode

- It is executed at query time, not at indexing time
- The hub and authority scores assigned to a page are query-specific.
- It computes two scores per document, hub and authority, as opposed to a single score.
- It is processed on a small subset of 'relevant' documents, not all documents as was the case with PageRank.

```
G := set of pages
for each page p in G do
  p.auth = 1 // p.auth is the authority score of the page p
  p.hub = 1 // p.hub is the hub score of the page p
function HubsAndAuthorities(G)
  for step from 1 to k do // run the algorithm for k steps
    for each page p in G do  // update all authority values first
      for each page q in p.incomingNeighbors do // p.incomingNeighbors is the set of pages that link to p
        p.auth += q.hub
    for each page p in G do  // then update all hub values
      for each page r in p.outgoingNeighbors do // p.outgoingNeighbors is the set of pages that p links to
        p.hub += r.auth
```

Conclusion

- Information propagation is heavily related to the network structure
- In addition to contents, links are powerful indicators to express the importance of an object