Big Data BUS 41201

Week 6: Networks

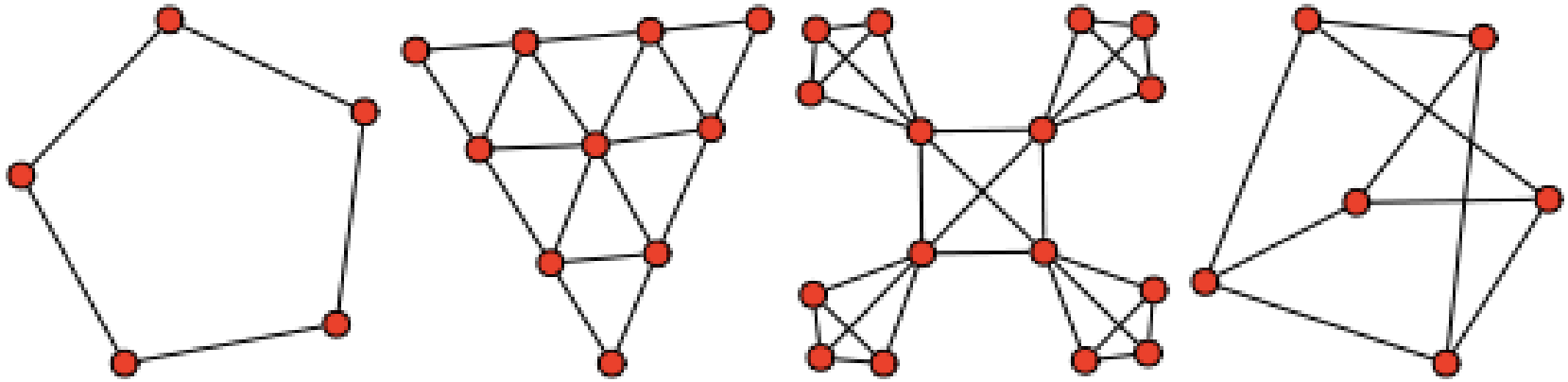
Veronika Ročková

University of Chicago Booth School of Business

http://faculty.chicagobooth.edu/veronika.rockova/

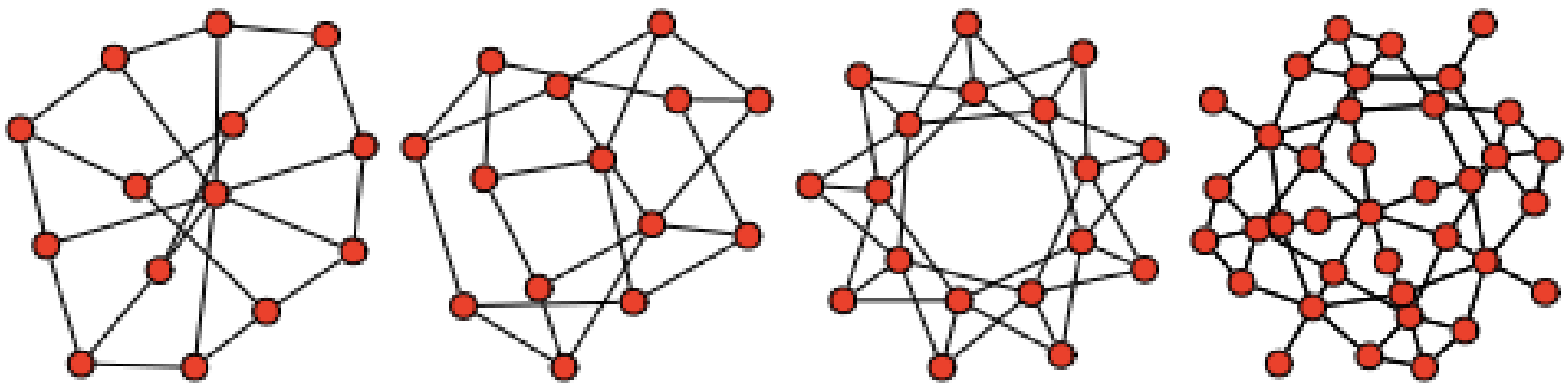
Networks

- ✓Network data examples
- ✓Graph representations
- ✓Summarization: nodes and edges, direction
- ✓Measuring connectivity and betweenness
- ✓Community detection
- ✓Market basket analysis
- ✓Page Rank for relevance ordering

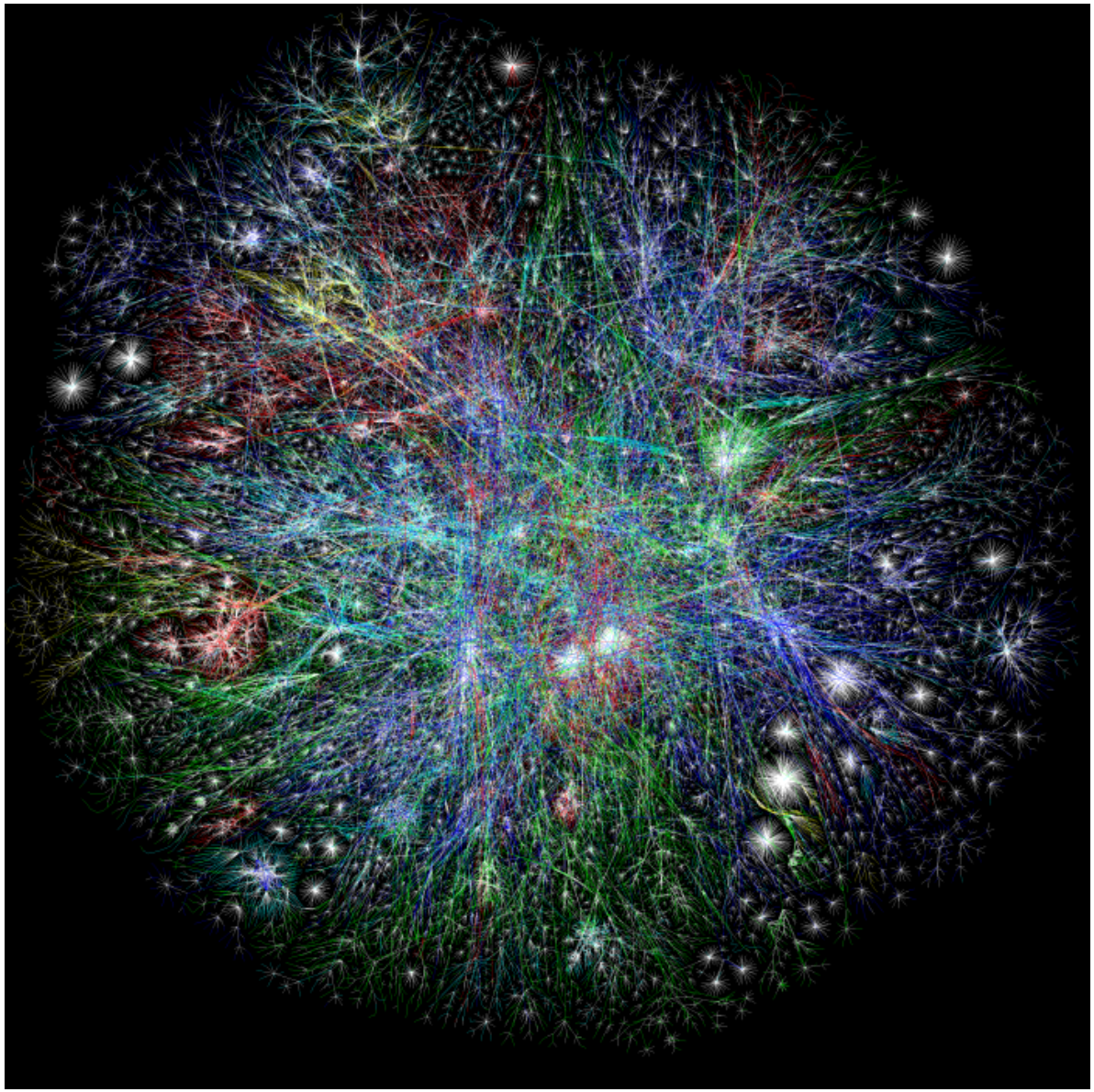


The network has nodes (vertices), such as a website or worker,

and edges are the (directed or undirected) links between nodes.



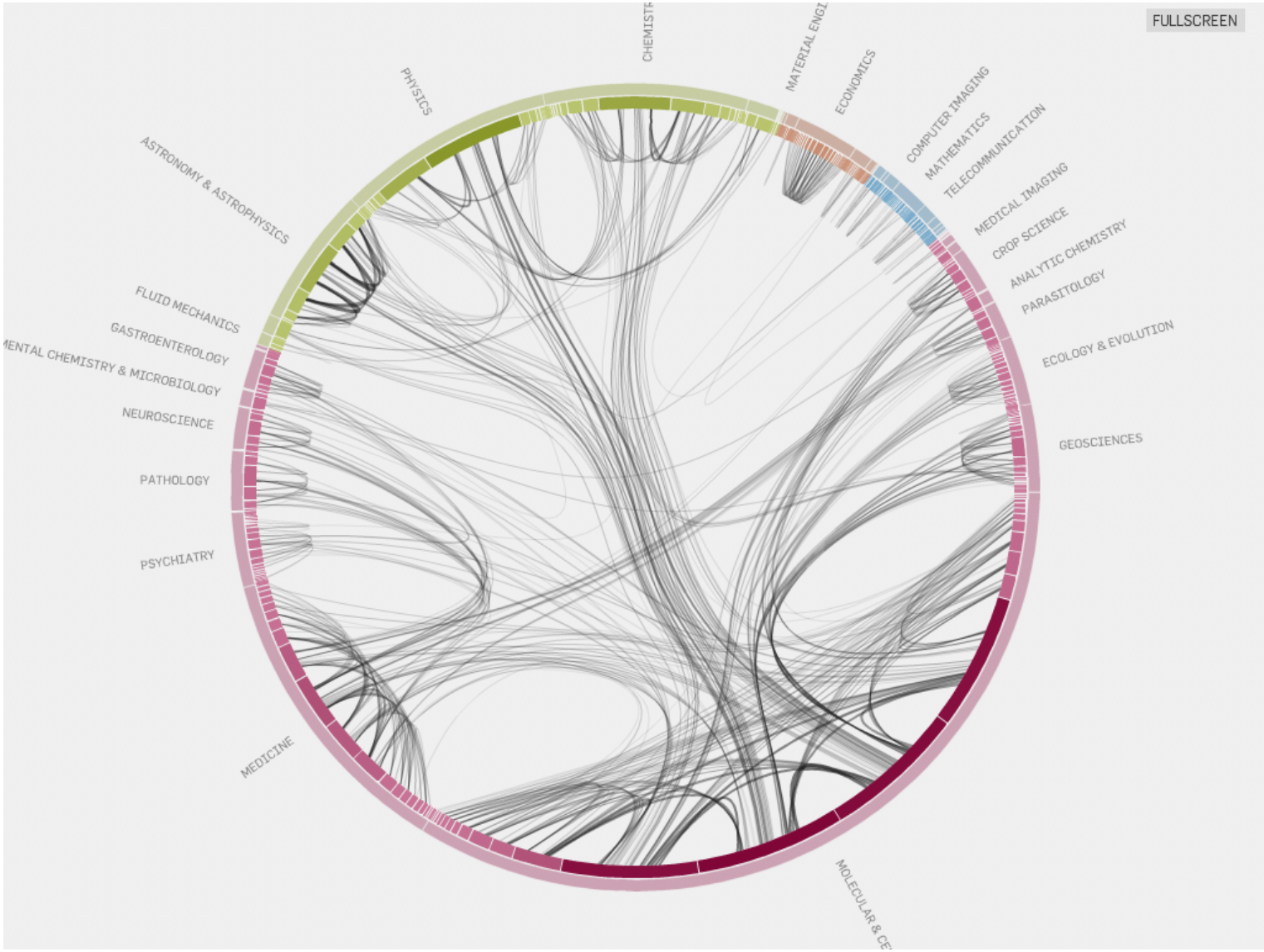
Internet — 50 billion Webpages



www.opte.org www.worldwidewebsize.com

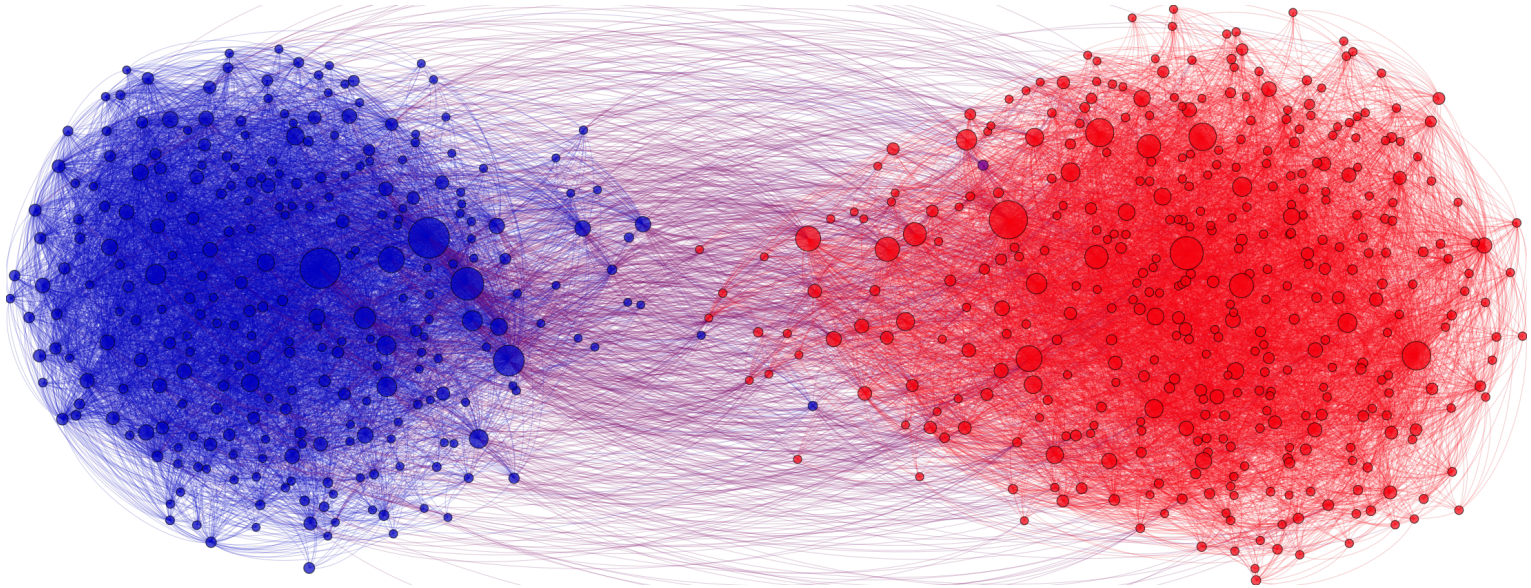
Facebook — 1.2 Billion Users

Citation Network — 250 Million Articles



well-formed.eigenfactor.org

Media networks



Connections between political blogs (Adamic, Glance, 2005)

Organizational networks

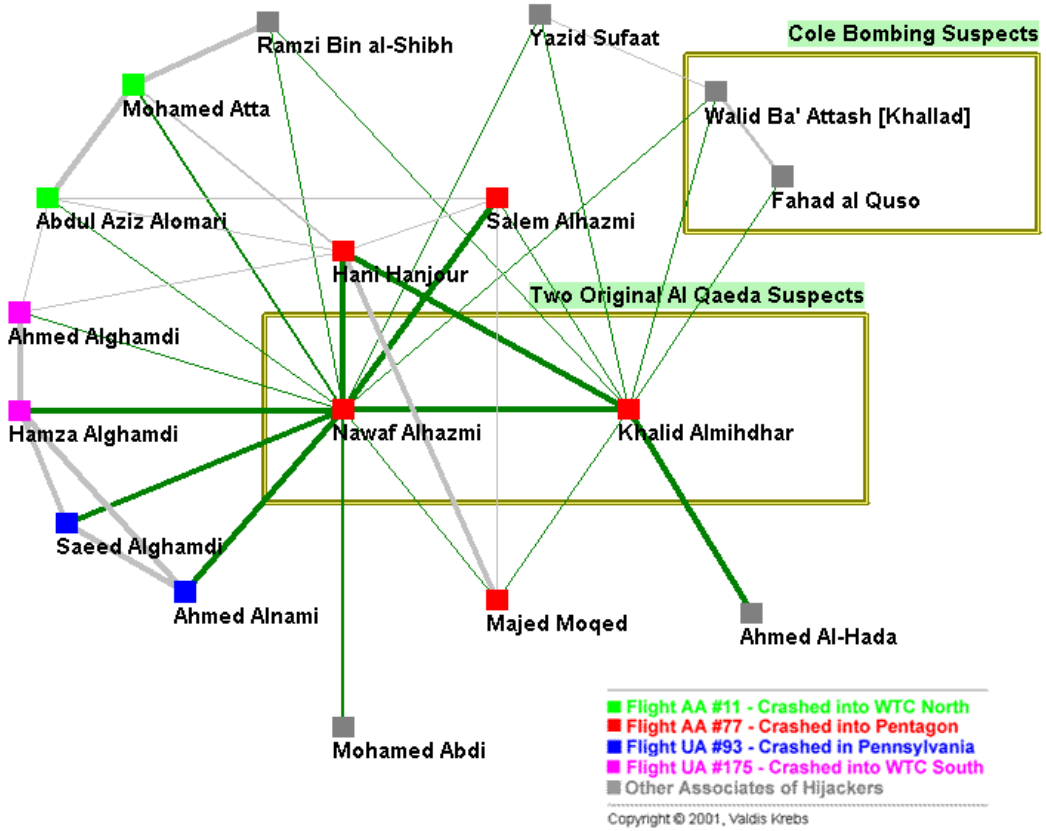


Figure 2 - All nodes within 1 step [direct link] of original suspects

9/11 terrorist network (Krebs, 2002)

Many more examples

who follows whom?

who calls whom?

who buys what?

Network Analysis

A rich set of tools to help us understand complex relationships, learn about behaviors, preferences and trends.

We'll begin by summarizing important properties.

In particular, we'll focus on measures of network connectivity.

Each node has connectivity statistics

Degree: How many other nodes are you connected to?

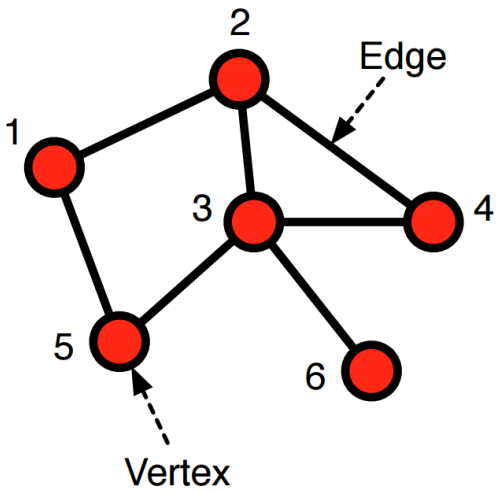
Betweenness: How many node-to-node paths go through you?

You can also make a lot of cool illustrations for graphs.
However, how to effectively visualize large networks is an open problem.

Basics: How to represent a network?

We will use graphs, which consist of vertices and edges.

Visually



Adjacency matrix

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

Adjacency list

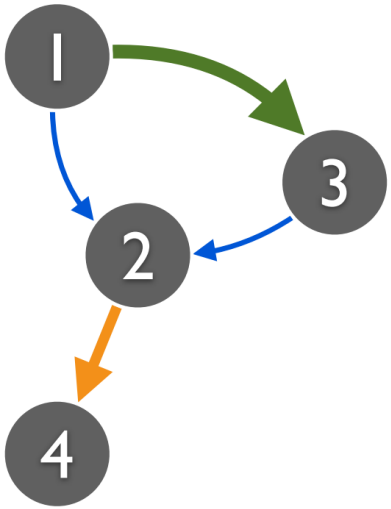
1	2	5		
2	3	1	4	
3	2	5	4	6
4	2	3		
5	1	3		
6	3			

Edge list

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

More exotic networks

Visually



Adjacency matrix

Target node

Source node	1	2	3	4
1	0	**1**	**3**	0
2	0	0	0	**2**
3	0	**1**	0	0
4	0	0	0	0

Adjacency list

1	2, 3
2	4
3	2

Edge list

1, 2, 1
1, 3, 3
2, 4, 2
3, 2, 1

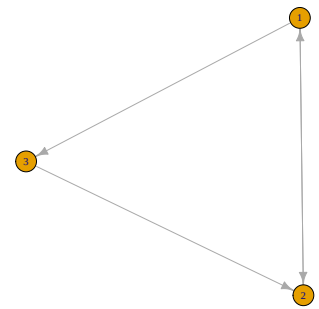
Network Graphs in R

`igraph` is a toolbox for visualizing and summarizing graphs.
It has front-ends for `R` and `Python`. Others: Gephi, Pajek, etc.

Unlike most R packages, `igraph` is well documented.
Type `help(igraph)` to get started.

For most applications, you'll read graphs from an edgelist:

1	2
1	3
2	1
3	2



```
edgemat <- as.matrix(read.table("edgelist.txt"))
graph <- graph.edgelist(edgemat)
```

Descriptive statistics of networks

Degree of a node

Number of edges connected to a node $d_i = \sum_j A_{ij}$

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

Degree distribution

k	$\Pr(k)$
1	1/6
2	3/6
3	1/6
4	1/6

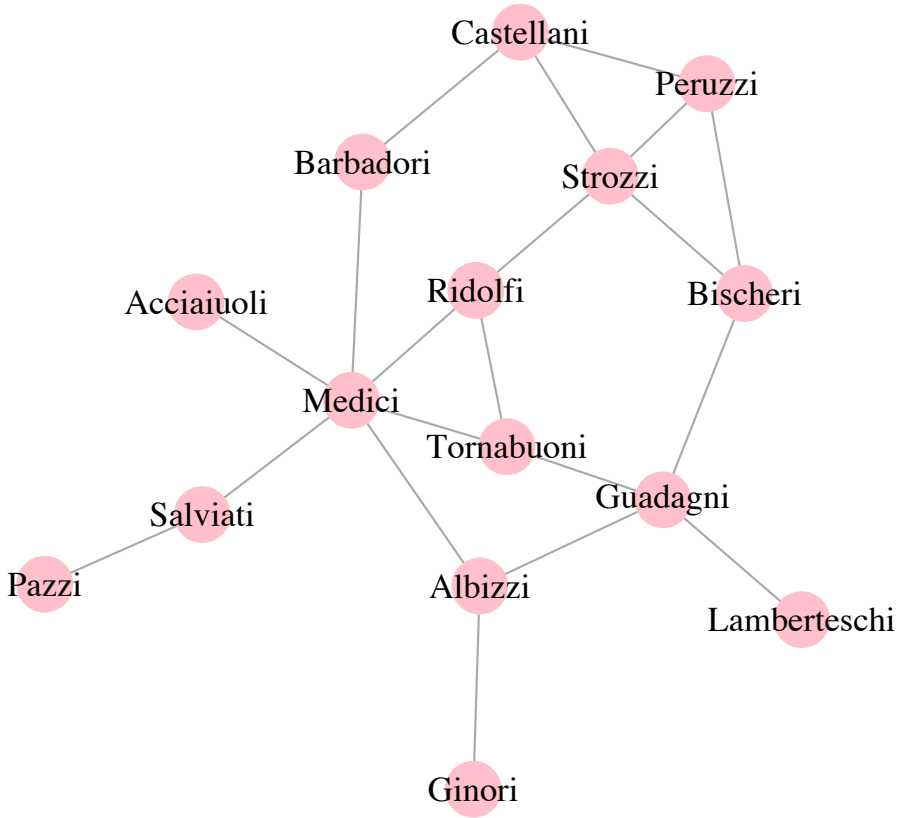
Marriage and Power

Early Renaissance Florence was ruled by an oligarchy of powerful families.
Padget and Ansell: *"Robust Action and the Rise of Medici"*

By the 15th century, the Medicis emerged supreme, & Medici Bank became the largest in Europe.

Political ties were established via marriage.
How did Medici win?

Marriage in Florence: 1250-1450



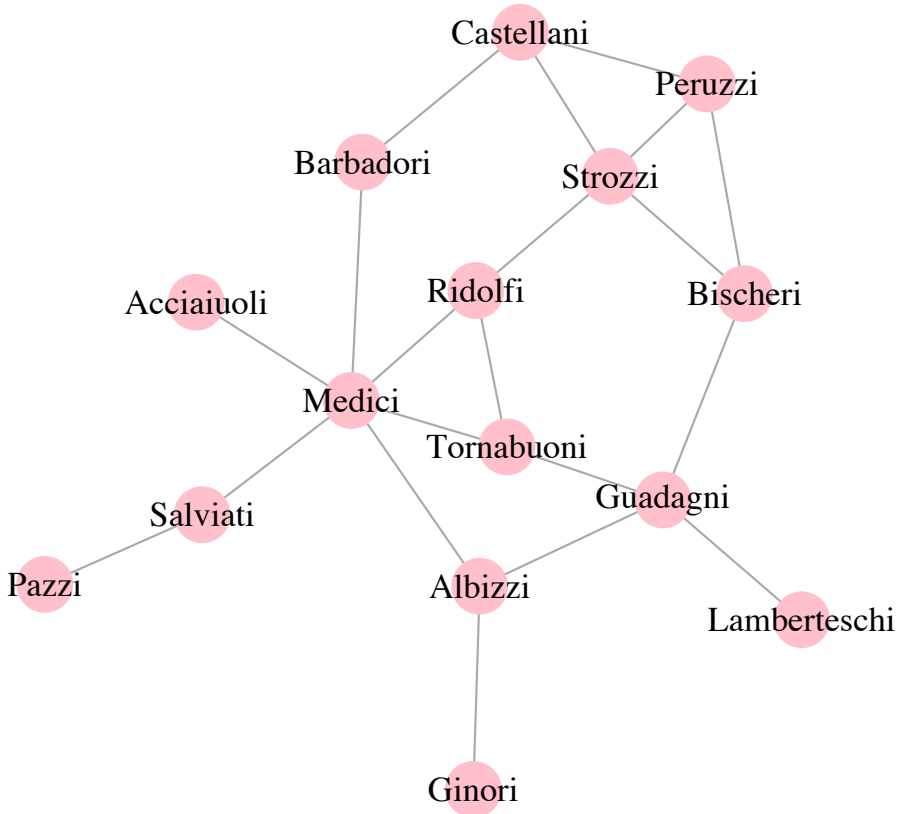
Network links can be used to measure "social capital".

A node's degree is its number of edges.

```
> sort(degree(marriage))
Ginori ... Strozzi Medici
     1           4      6
```

Medicis are connected!

Medici Network: Closer Look



The Medici family very centralized (a star shaped network)

$\rightsquigarrow$ *Medici relatives were connected almost solely through Medici's themselves.*

The rival Strozzi's network is far denser (competing claims for network leadership).

Deeper network structure with betweenness

A node is important if it lies on many shortest-paths so it is essential in passing information through the network.

An alternative to degree, betweenness measure the proportion of shortest paths containing a given node.

Shortest path: fewest steps from i to j (direction matters).

$$\text{betweenness}(k) = \sum_{i,j:i\neq j,k\notin\{i,j\}} \frac{s_{ij}(k)}{s_{ij}}$$

- $\rightsquigarrow$ $s_{ij}(k)$ number of shortest paths from i to j that go through k
- $\rightsquigarrow$ s_{ij} number of shortest paths from i to j

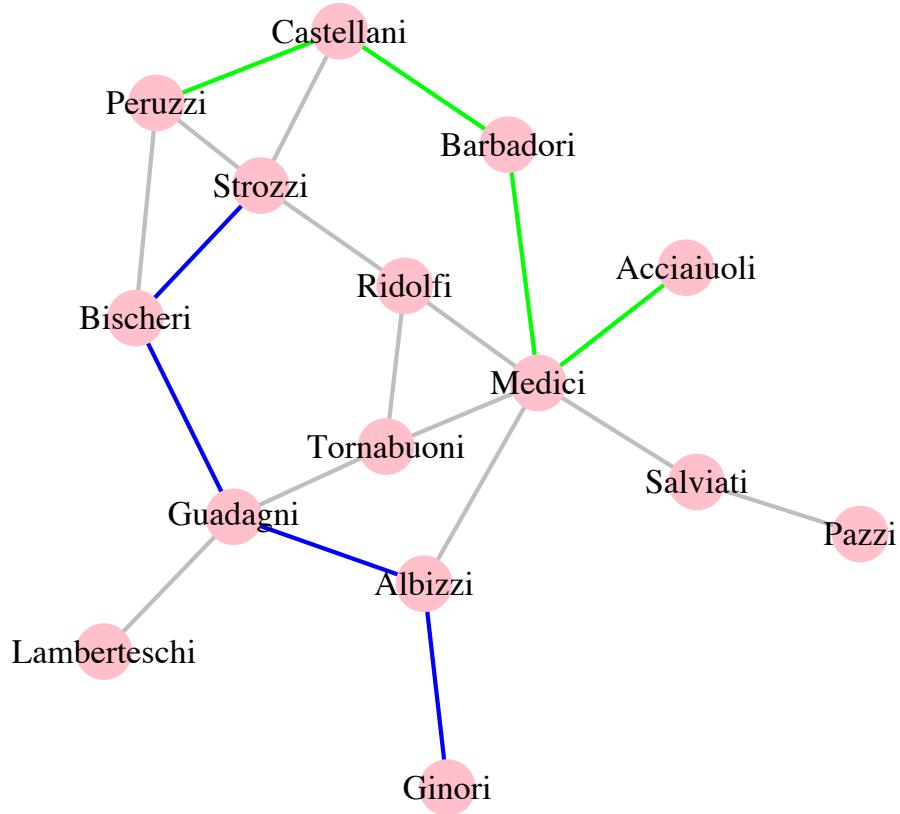
Measures how much influence a node has over connections between others, i.e. how often a node serves as the "bridge".

Betweenness vs Degree

Medicis have the highest degree, but only by a factor of 3/2 over the Strozzis.

But their betweenness is 5 times higher!

Betweenness measures *deep graph connectivity*, rather than just counting neighbors.




```
> sort(betweenness(marriage))
 Ginori   ...   Strozzi   Medici
    0.0             9.3     47.5
```

Structural Holes

A structural hole is a *low-degree* node in an organization chart with *high betweenness*.

Social Capital exists where people have an advantage because of their location in a social structure (brokerage opportunities).

Holes can act like bottlenecks in companies, and lead to unexpected employees having excess power and influence.

But if you're the employee, it's a fast track to promotion.

"People who stand near the holes in a social structure are at higher risk of having good ideas."

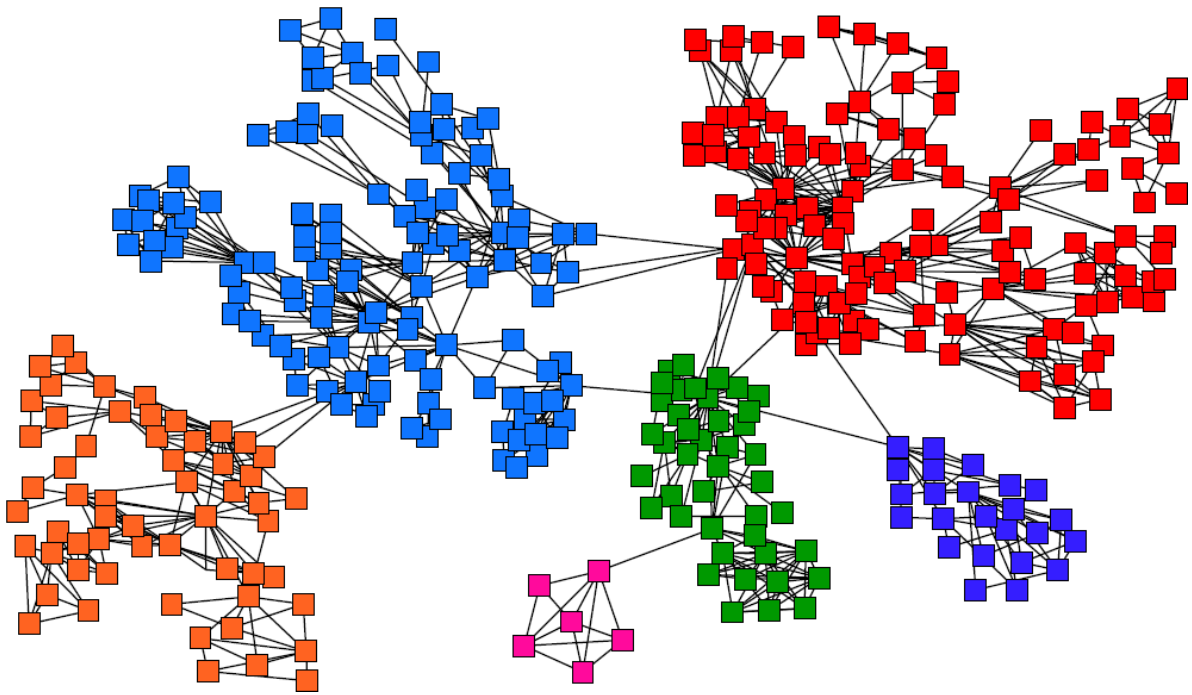
Burt: *Structural Holes and Good Ideas*, AJS 2004.

`igraph` has `constraint` for finding structural holes.

Community Detection

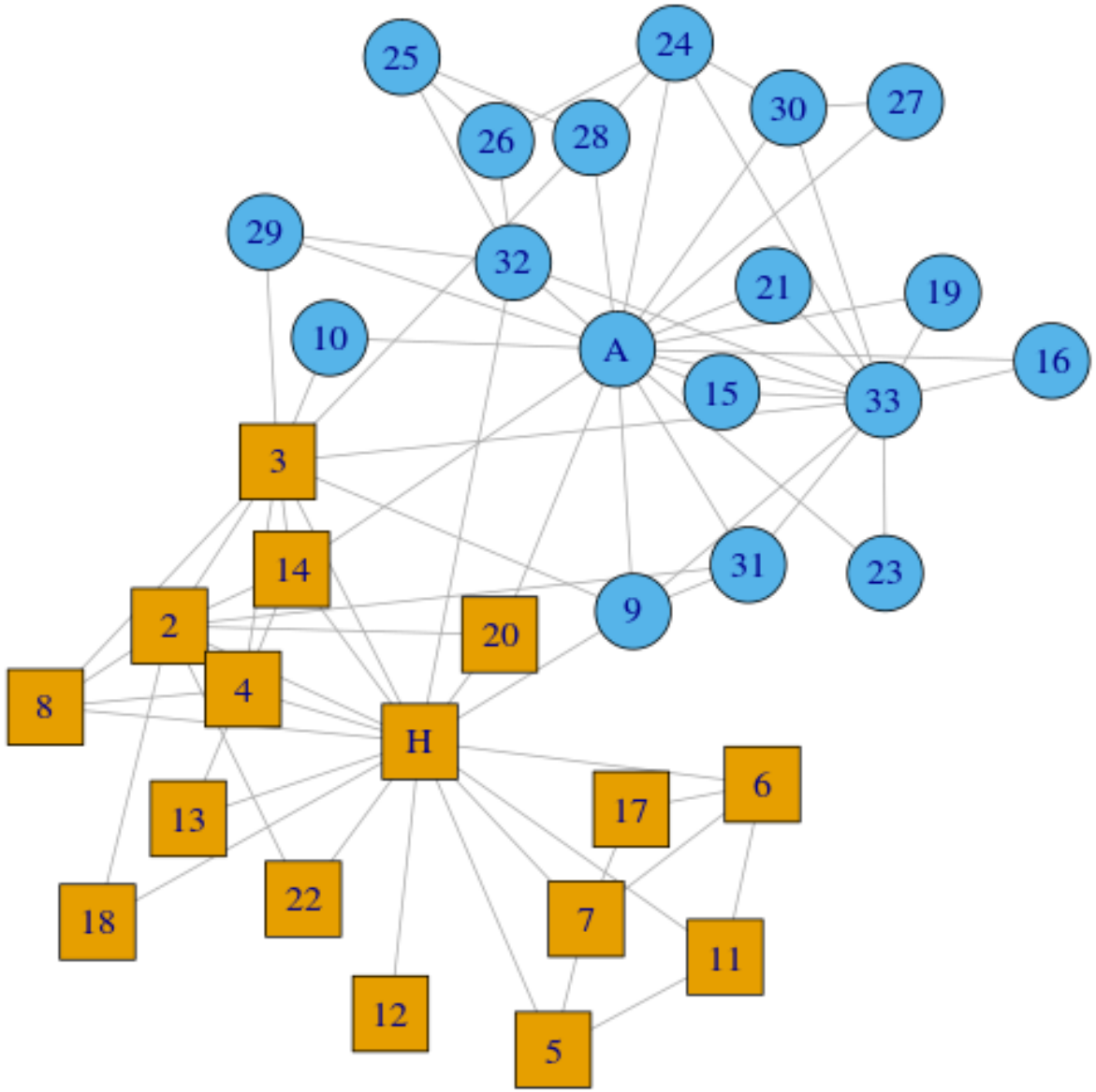
Networks are often organized into **communities**: groups of nodes that are *densely connected inside* but only *loosely connected across.*

The goal is to identify meaningful communities in an automated way.



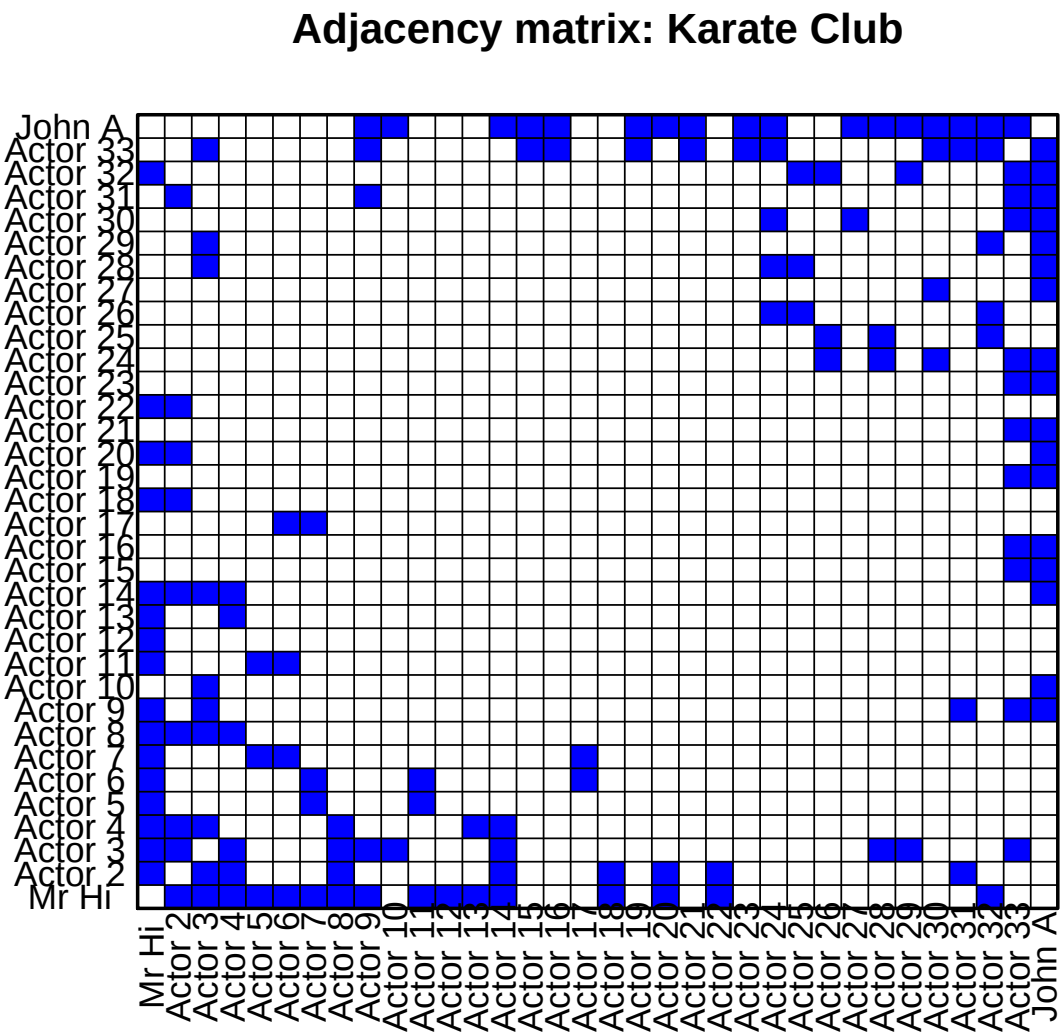
Karate Club Example

Zachary's karate club network (H: Instructor, A: Club president)

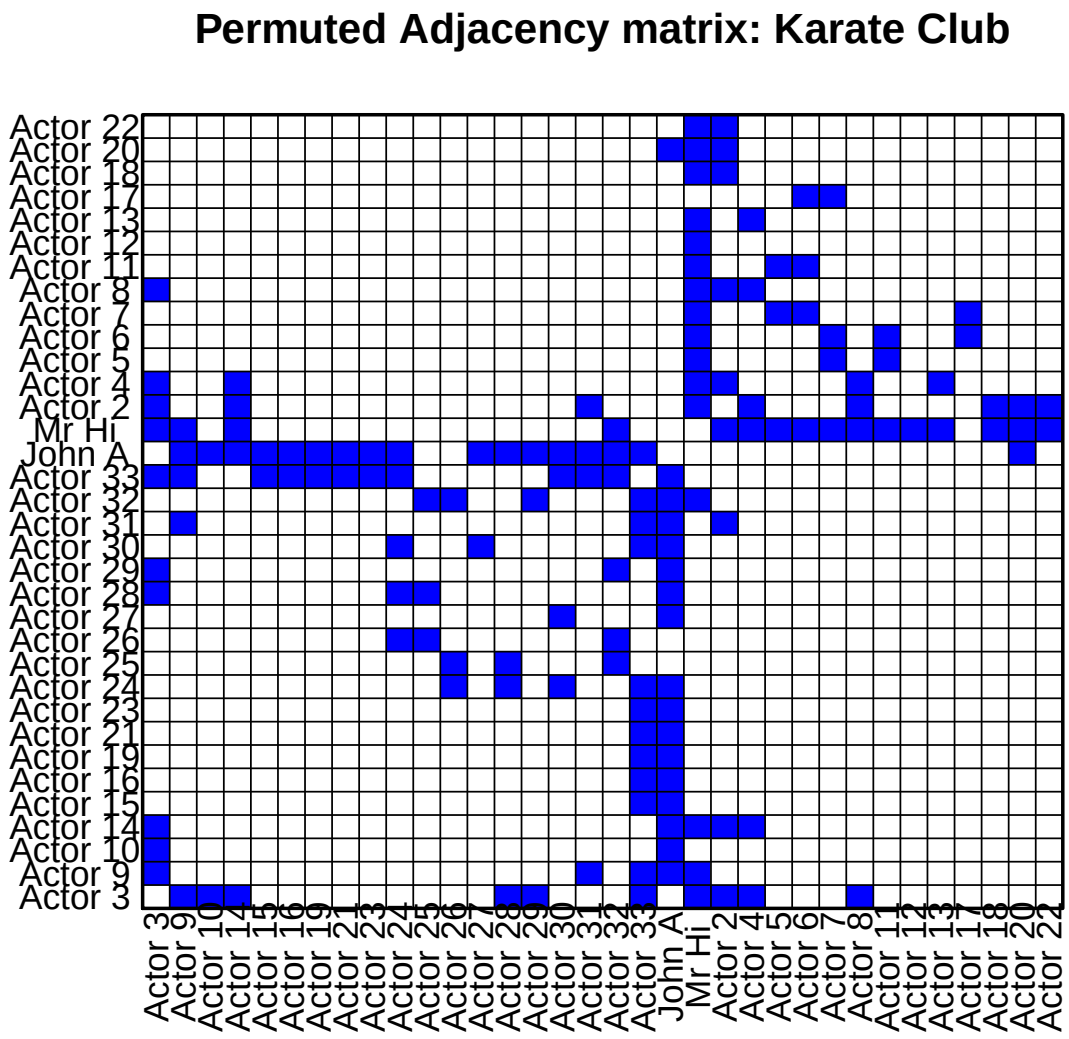


http://networkkarate.tumblr.com/

Karate club: Adjacency matrix



Karate club: Reordered adjacency matrix



Edge Betweenness and Community Detection

The *edge betweenness* is a measure of traffic flow through an edge rather than a node.

It is the proportion of shortest paths between pairs of nodes that run through *the edge*, summed over all possible node pairs.

If a network contains communities that are only loosely connected, then all shortest paths between different communities must go along only a few edges.

Community detection can be achieved by removing *high-betweenness edges* in a graph $\rightsquigarrow$ *Girvan-Newman algorithm*

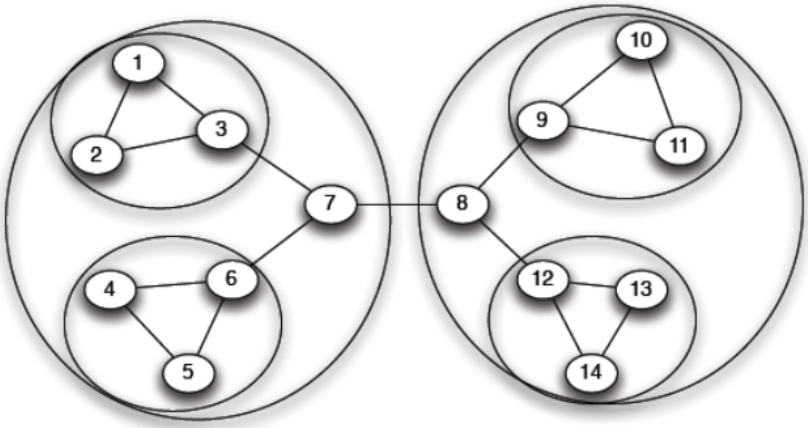
(1) *Calculate betweenness of all edges in a graph.*

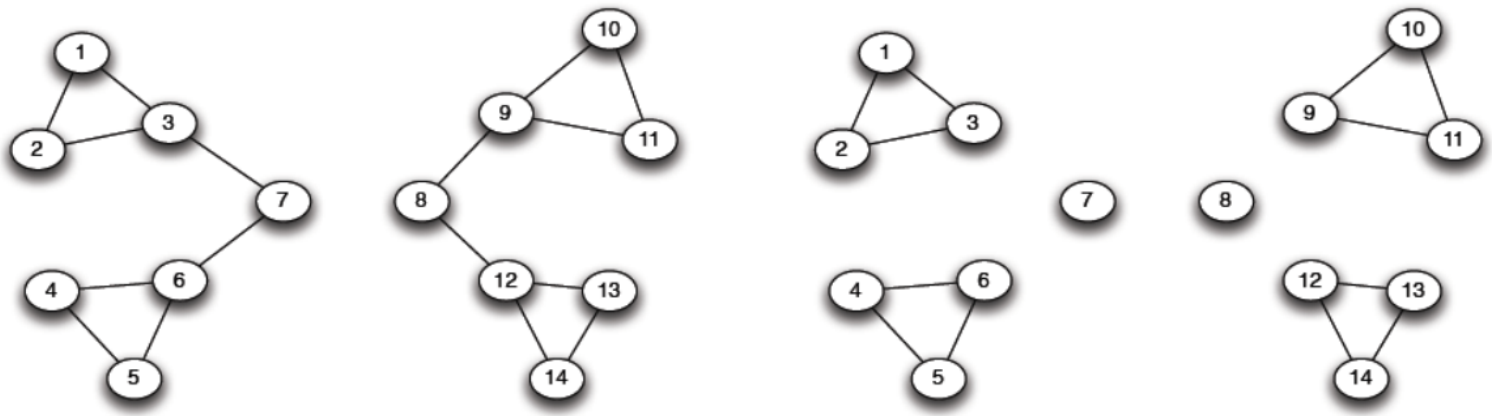
(2) *Remove the edge with the highest betweenness.*

(3) *Repeat (1) until no edges remain.*

Community Detection: Edge Betweenness

Successively remove edges of highest betweenness, breaking up the network into separate components





(a) *Step 1* (b) *Step 2*

Collaborative Filtering

A common question in data mining:
what do one person's choices say about anothers?
As amazon says: "people who buy this book also bought..."

These types of tasks are referred to as 'collaborative filtering':
using shared choices to predict preferences.

It's a big field, with many tools

- logistic regression of each product on to all other choices.
- principal componenets analysis: underlying taste factors.

Many of the tools from this class apply (projects?).

But as an easy start, there are good fast algorithms
for discovering low dimensional *association rules*.

Association Rules

Frequent itemset mining is a popular tool for discovering purchasing patterns from large commercial databases.

Consider $\mathbf{X} = \{X_1, \ldots, X_p\}$ a set of *binary attributes* called *items*.

Let $T = \{T_1, \ldots, T_n\}$ be a *database of transactions*, where each transaction T_i is associated with an item set S_i (subset of variables in $\mathbf{X}$ for which $X_j = 1$).

For two item sets S_i and S_j, we define an implication

$$S_i \Rightarrow S_j$$

as an *association rule* when S_j occurs more frequently when S_i occurs.

The LHS item S_i is the *antecendent* and the RHS item S_j is the *consequent*.

Association Rules: Toy Example

The goal is to find interesting item co-ocurrences and build association rules.

Assume $\mathbf{X} = \{\texttt{milk, bread, butter, beer}\}$.

Database of transactions

Transaction ID	Items
1	milk,bread
2	bread, butter
3	beer
4	milk, bread, butter
5	bread, butter

An example of rule can be

$$\{\texttt{milk, bread}\} \Rightarrow \texttt{butter}.$$

Association Rules

There are various measures of informativeness among item sets and association rules.

Support of item set S $\texttt{supp}(S)$ is the proportion of transactions that contain S

$$\texttt{supp}(\{\textit{milk}, \textit{bread}\}) = 2/5$$

It is an estimate of the probability of purchasing {milk, bread}

Confidence of a rule $S_i \Rightarrow S_j$ is the proportion of S_i transactions that include also S_j:

$$\texttt{conf}\big(S_i \Rightarrow S_j\big) = \frac{\texttt{supp}\big(S_i \text{ and } S_j\big)}{\texttt{supp}(S_i)}$$

$$\texttt{conf}\big(\{\textit{milk}, \textit{bread}\} \Rightarrow \{\textit{butter}\}\big) = \frac{\texttt{supp}(\{\textit{milk}, \textit{bread}, \textit{butter}\})}{\texttt{supp}(\{\textit{milk}, \textit{bread}\})} = 1/2$$

It is an estimate of the **conditional** probability of purchasing butter **given** {milk,bread}

Association Rules

Association rules (AR's) should have enough of support and confidence. To narrow down focus to even fewer and even more interesting AR's, one can use *lift*.

Lift of a rule $S_i \Rightarrow S_j$ measures the increase in support of S_j when S_i occurs

$$\texttt{lift}\big(S_i \Rightarrow S_j\big) = \frac{\texttt{supp}\big(S_i \text{ and } S_j\big)}{\texttt{supp}(S_i)\texttt{supp}(S_j)} = \frac{\texttt{conf}\big(S_i \Rightarrow S_j\big)}{\texttt{supp}(S_j)}$$

It can be regarded as a ratio of a **conditional** probability of purchasing `butter`, **given** $\{\texttt{milk,bread}\}$, and an **unconditional** probability of purchasing `butter`.

$$\texttt{lift}\big(\{\textit{milk}, \textit{bread}\} \Rightarrow \{\texttt{butter}\}\big) = 1/2 \times 5/3$$

Greater lift values indicate stronger associations.

Support, Association, and Lift

Ex: when you buy chips, you need beer to wash them down.

Suppose that beer is purchased 10% of the time in general, but 50% of the time when the consumer grabs chips.

- The *support* for 'beer' is 10%
- The *confidence* of this rule is 50%.
- It's *lift* is 5: 50% is 5 times higher than 10%.

Given this information, you could put some chips by the beer.

Generally, association rules with high lift are most useful.

Low support does not preclude high confidence or high lift.

Chips $\Rightarrow$ Beer is high support, but low lift if everybody always buys beer.

Vodka $\Rightarrow$ Caviar is low support, but high lift if people caviar for their parties.

Finding Association Rules with R

There's no deep theory around ARules. We just scan the high-lift or high-confidence rules to find interesting rules.

To find confidence and lift, just count the number of times RHS and LHS happen, and how often they happen together.

$$\text{supp(event)} = \frac{\text{number of times event occurs}}{\text{number of observations}}$$

However, counting all possible combinations can take forever.
Apriori: algorithm for finding rules over a support threshold.

The `apriori` function is available in the `arules` package.

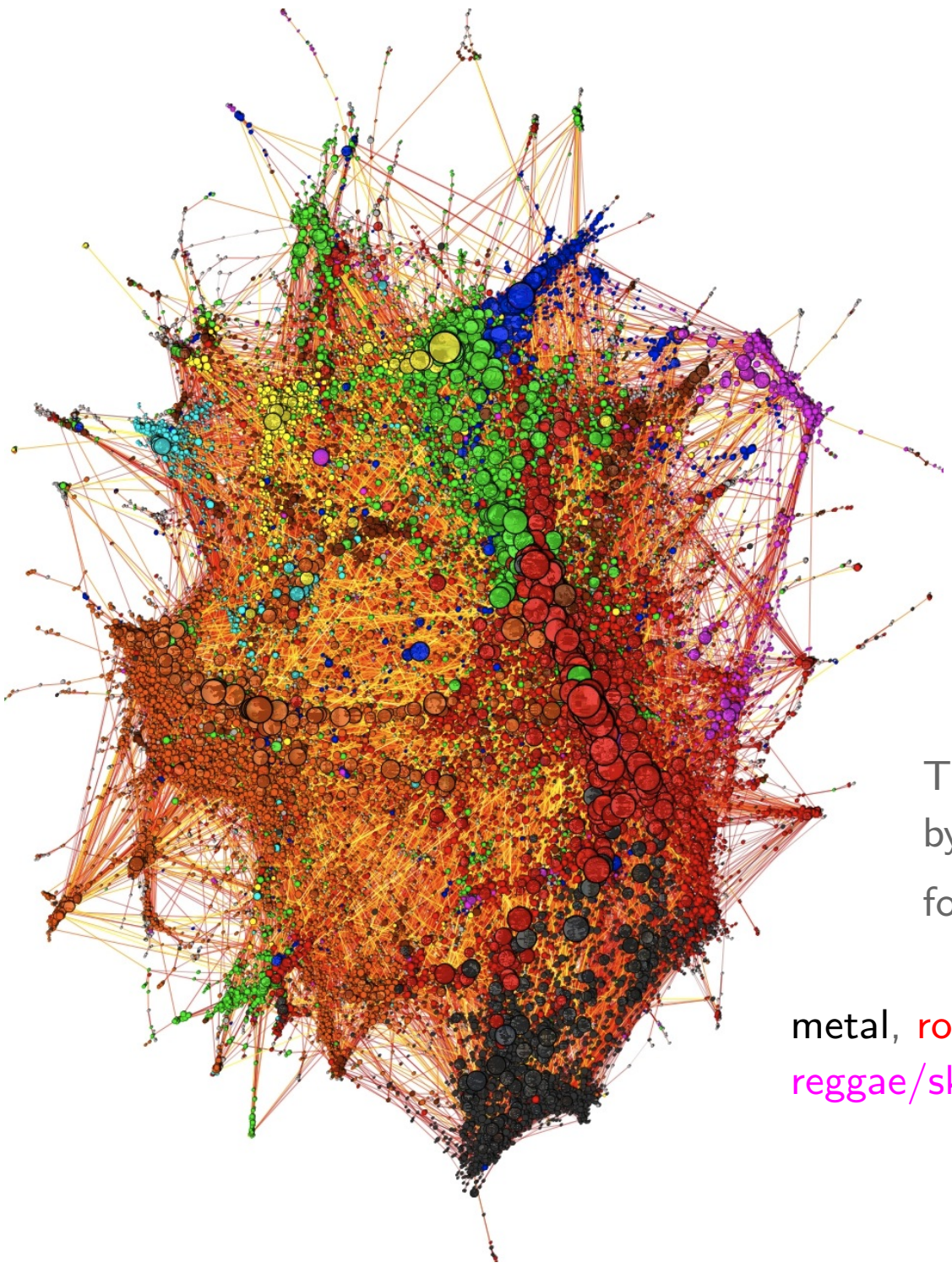
You need to get the data in a certain format,
but after this it is straightforward to use.

Binary Incidence Matrix

Convenient representation of transactions with binary incidence matrix.

Transaction	X_1 milk	X_2 bread	X_3 butter	X_4 beer
T_1	1	1	0	0
T_2	0	1	1	0
T_3	0	0	0	1
T_4	1	1	1	0
T_5	0	1	1	0

The number of purchase opportunities as well as products can be huge. Coding as a sparse matrix is extremely helpful.



Last.fm Artist Plays

Online radio keeps track of everything you play, for recommending music & focused marketing.

This 'network' shows artists sized by play count, with lines (edges) for shared users.

metal, rock, pop, jazz, electronica, hip-hop
reggae/ska, classical, folk/country/world.

Association rules for Music Taste

lhs		rhs	support	confidence	lift
t.i.	=>	kanye west	0.0104	0.5672	8.8544
pink floyd, the doors	=>	led zeppelin	0.0106	0.5387	6.8020
beyonce	=>	rihanna	0.0139	0.4686	10.8810
morrissey	=>	the smiths	0.0112	0.4655	8.8961
megadeth	=>	iron maiden	0.0132	0.4307	7.2677
jimi hendrix	=>	the doors	0.0120	0.3062	5.3170
nelly furtado	=>	madonna	0.0100	0.2750	5.0374
bright eyes	=>	the shins	0.0102	0.2698	5.4623
elliott smith	=>	modest mouse	0.0109	0.2679	5.1732
britney spears	=>	lady gaga	0.0120	0.2612	7.7292
ramones	=>	the clash	0.0104	0.2586	5.9052
franz ferdinand	=>	kaiser chiefs	0.0132	0.2224	7.1153

Example: Given a new user that listens to a lot of Morrissey, we're 46% positive that they'll also like the Smiths;

This is 9 times higher than if we didn't know about Morrissey.

From association to networks

Graphs can be a useful way to summarize all sorts of data.
We can define networks using any measure of connectivity.

For example, an association network:

> Say there's an edge between `lhs` and `rhs` if `support` and `confidence` are greater than some thresholds.

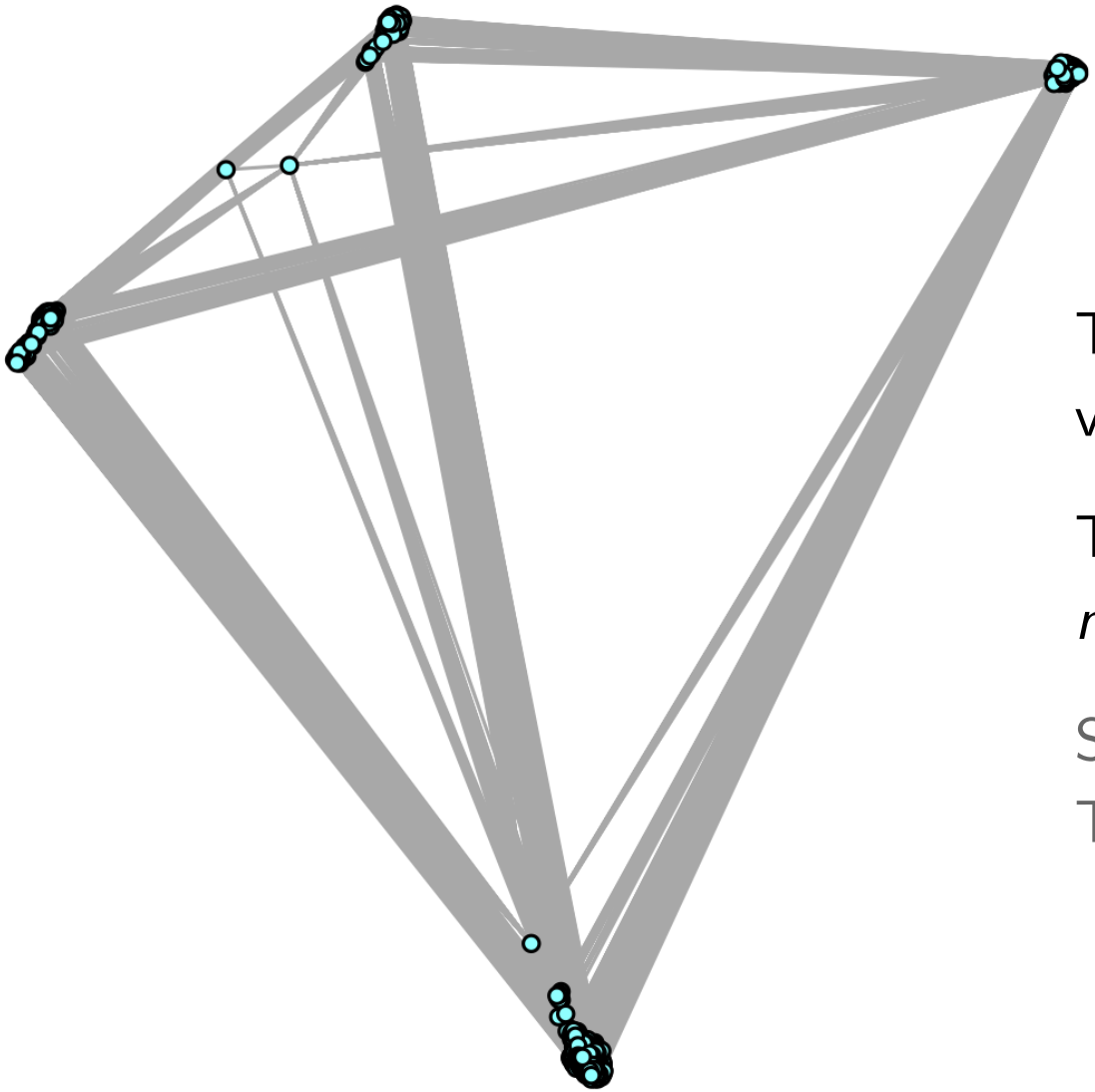
If we just look at any shared membership in a playlist, we get our monster graph from the beginning.

For example, in the `lastfm.R` code we use rules from

```
apriori(playtrans,
  parameter=list(support=.001, confidence=.1, maxlen=2))
```

to define a network with 1k nodes and 36k edges.

0.1% support and 10% confidence lastfm network



The network has four very distinct cliques.

These look something like *metal*, *hip-hop*, *alt*, *pop*.

See code for plotting. There's lots you can do.

Search for "California"

Search is one great example of network analysis;

Consider ranking sites for the query "California".

$\rightsquigarrow$ Take 200 pages with heavy traffic and high term-frequency for "California".

$\rightsquigarrow$ Follow links to build a neighborhood.

We're left with about 10,000 sites, with links, to rank.

```
caedges <- read.csv("CaliforniaEdges.csv")
casites <- scan("CaliforniaNodes.txt", "character")
edgemat <- cbind(casites[caedges$from], casites[caedges$to])
```

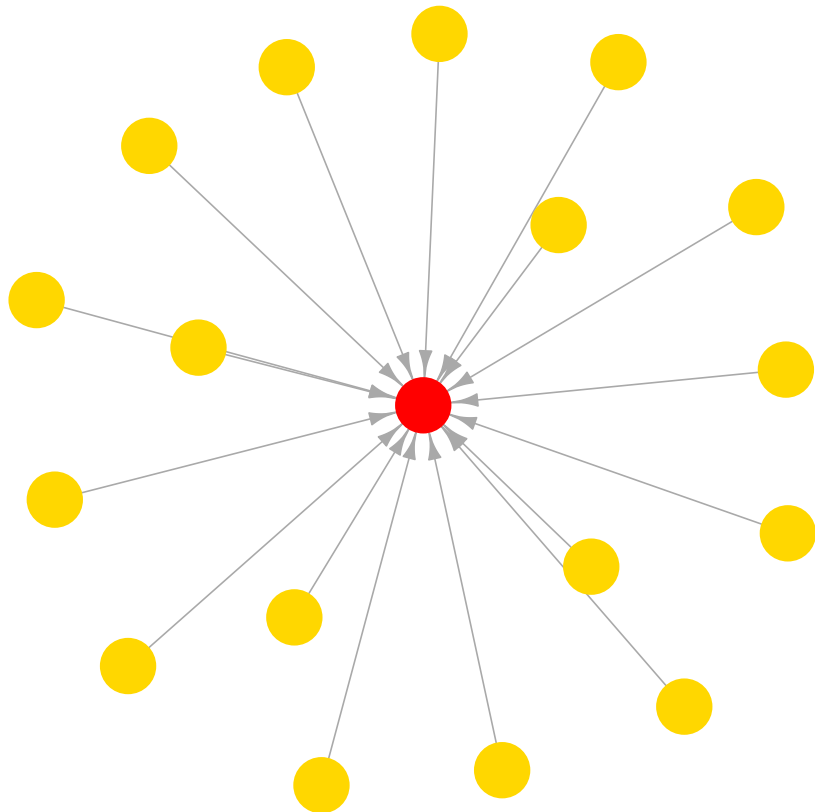
The query provides a very large directed network

Look at neighborhoods to get a workable plot.

```
latimes <- graph.neighborhood(calink, order=1, ...)
```

LA Times network

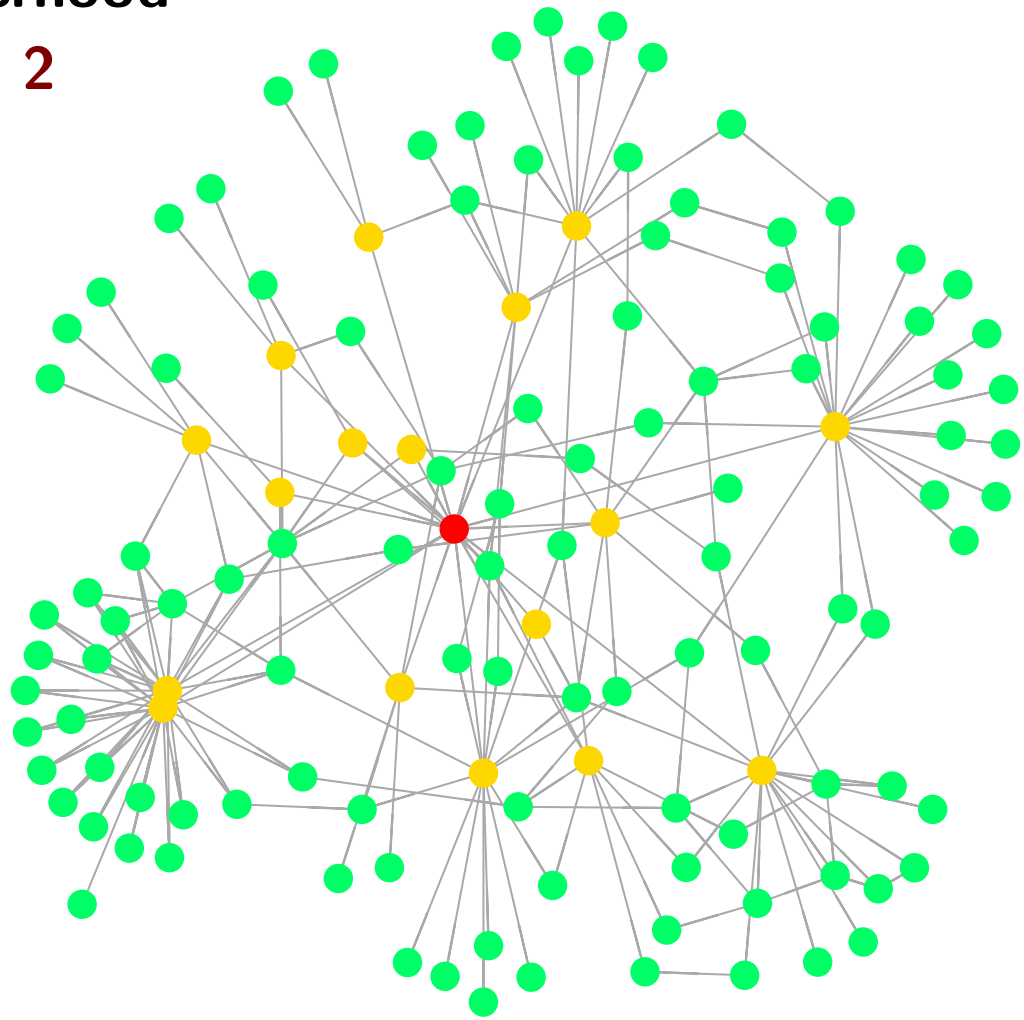
Neighborhood order is the number of included steps away from the node.



At order=1, we just have sites pointing to `latimes.com`.

LA Times Neighborhood

Expanded to order 2



Just going one extra step creates a much bigger network.

Yellow points are the first-order connections from before.

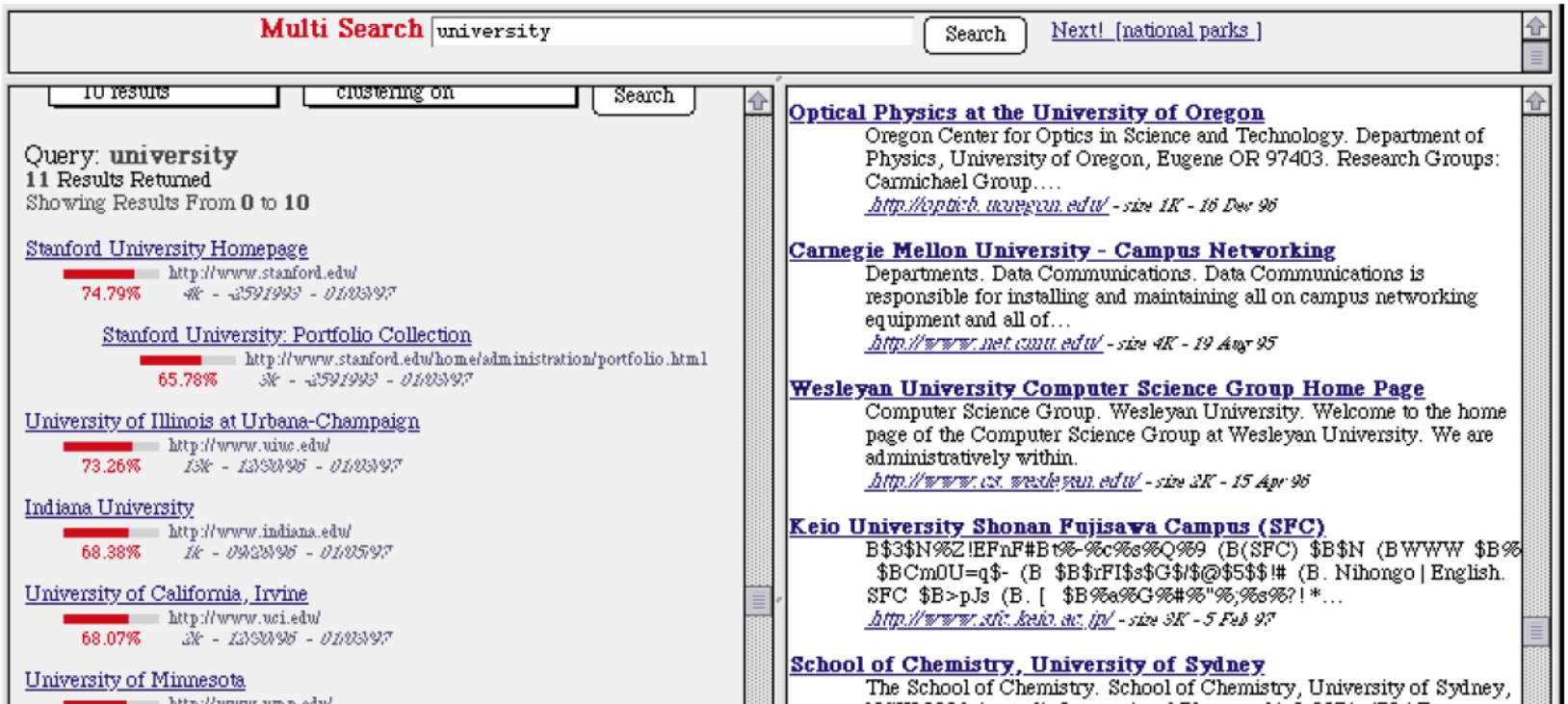
Google's PageRank Algorithm

Google has been pretty successful. Page Rank is a key ingredient.

Suppose we have N webpages and we want to rank them in terms of their likely relevance to the websurfer.

PageRank algorithm, famously invented by Larry Page and Sergei Brin (founders of Google), assigns a score to each webpage

Paper: The Anatomy of a Large-Scale Hypertextual Web Search Engine



PageRank Algorithm

Page Rank labels a site more important *if many sites link to it*.

However, we don't want to treat all linking webpages equally.

Instead, we weight the links from different webpages

(1) Webpages that link to i, and have *high PageRank* scores themselves, should be given more weight

(2) Webpages that link to i, but *link to a lot of other webpages* in general, should be given less weight

The rank r_i of i^{th} page depends on all the other ranks r_j of pages that point to it $\rightsquigarrow$ circular argument.

PageRank Algorithm

The circularity can be exploited in recursive calculations.

$$r_i = \sum_{j=1}^{N} \frac{e_{ij}}{c_j} r_j$$

- r_i is the page rank,
- e_{ij} is a binary edge indicator, and
- $c_j = \sum_{i=1}^{N} e_{ij}$ is the number of nodes pointed at by node j.

Start with an initial vector $\mathbf{r}^{(0)}$ and update $\mathbf{r}^{(k)} = \mathbf{A}\mathbf{r}^{(k-1)}$ until convergence, where $\mathbf{A}$ is a sparse weighted adjacency matrix.

☺ Very efficient and simple calculation.

Page rank of "california" search response

We can run PageRank to organize our list of sites.

```
> search <- page.rank(calink)$vector
> casites[order(search, decreasing=TRUE)]
 [1] "http://www.calgold.com/"
 [2] "http://www.sancarlos-homes.com/info.asp"
 [3] "http://spectacle.berkeley.edu/"
 [4] "http://www.graddiv.ucr.edu/"
 [5] "http://www.chico-homes.com/info.asp"
 [6] "http://www.webb.pvt.k12.ca.us/~webb/WSCPrograms.html"
 [7] "http://www.berkeley.edu/"
 [8] "http://www.calfund.org/"
 [9] "http://www.ca-probate.com/"
[10] "http://www.ppconline.com/"
```

I don't think this alone would have made google famous!

Nodes need to be weighted by traffic and by successful clicks.