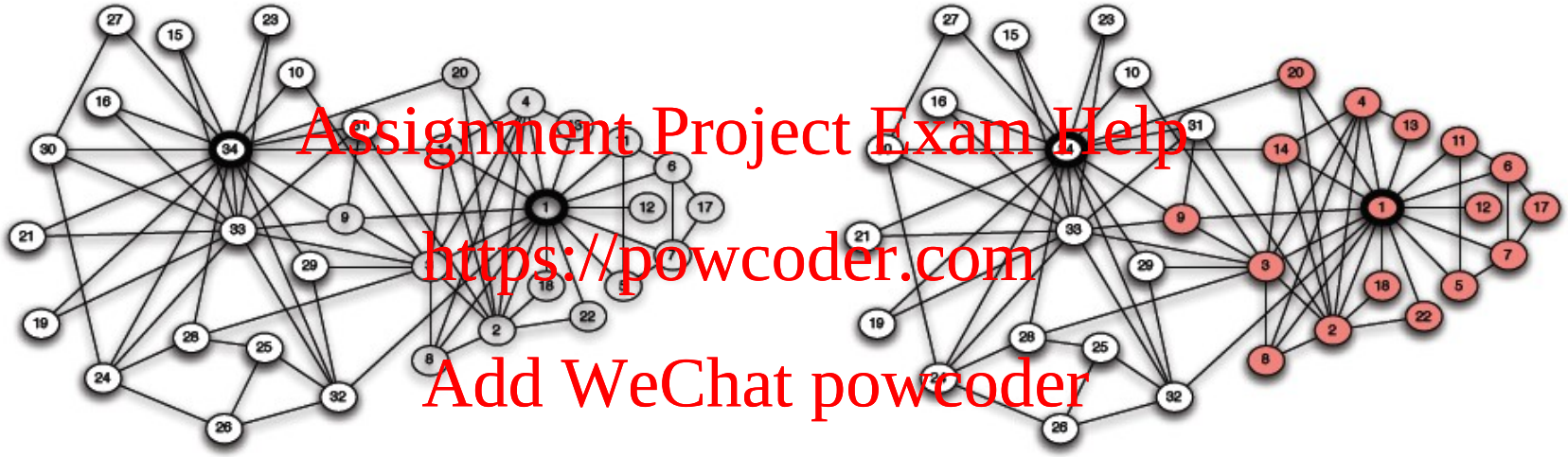
Social Network Analysis
Community Detection

Robin Burke
DePaul University
Chicago, IL

Finding groups in networks

- Discover communities of practice
- Measure isolation of groups
- Understand opinion dynamics / adoption

Zachary Karate Club

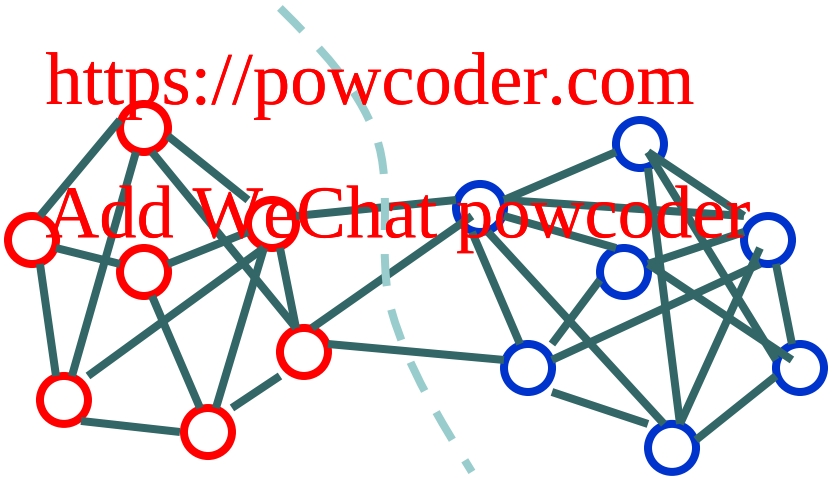


(a) *Karate club network* (b) *After a split into two clubs*

source:Easley/Kleinberg

Community finding

- Social and other networks have a natural community structure
- We want to discover this structure rather than impose a certain size of community or fix the number of communities



What makes a community

- similarity of members
  - shared connectivity
- mutuality of ties
  - everybody in the group knows everybody else
- frequency of ties among members
  - everybody in the group has links to at least k others in the group
- closeness or reachability of a subgroup
  - individual are separated by at most n hops
- relative frequency of ties among subgroup members compared to nonmembers

Communities summarize

- Grouping nodes can give us a “high level” picture of a network

Open research topic

- Even though we know (in some cases) good measures for cluster quality
  - we can't calculate those clusterings
  - for reasonably large networks
- Lots of research on-going on this question
  - Talk about BIGCLAM later

Modularity

- Seen this measure before
  - for assortativity
- Are there more edges than you would expect if the connections were random?
- Measure this probability over all clusters
- Random clustering has modularity = 0
  - because edges in and out of the clusters are equally likely
- Good clustering has high modularity
  - edges within clusters are many
  - edges between clusters relatively few

Modularity

- Use cluster ids in our previous equation

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{i,j} - \frac{k_i k_j}{2m}\right]\cdot\delta(c_i,c_j)$$

- Remember also B
  - modularity matrix

$$B_{i,j} = \left[A_{i,j} - \frac{k_i k_j}{2m}\right]$$

Modularity maximization

- We can attempt to find maximum modularity
  - look at all possible subsets of nodes
  - exponential task
  - built into R
    - `cluster_optimal()`
  - small graphs only!

Greedy approaches

- Merge nodes into a community as long as the modularity continues to increase
  - Then move to a different community
  - Continue until finished
- No guarantee of optimality
- In R
  - `cluster_louvain()`
- Also implemented in Gephi
  - Used this in lab
- A faster version
  - `cluster_fast_greedy()`

Edge weights

- igraph methods will use edge weights if present
  - and labeled “weight”
- Same modularity calculation applies
  - now the “value” of an edge is weighted
  - a weak edge crossing group boundaries
    - not as significant as a strong edge

Simulated annealing

- An optimization approach
- Basically
  - add and remove nodes from communities
  - occasionally allow moves that make the modularity worse
  - but tolerance for “bad” moves goes down over time
- Eventually the communities converge
- In R
  - `cluster_spinglass()`
  - this has the nice property that you can find a community for a single vertex without finding all of them

Spectral methods

- Remember equation with modularity matrix

$$Q = \frac{1}{2m} \sum_{i,j} B_{i,j} \delta(c_i, c_j)$$

- Create a vector s where s_i=1 if in group 1 and s_i=-1 if in group 2

- We can rewrite Q as

$$Q = \frac{1}{2m} \sum_{i,j} s^T B_{i,j} s$$

Solving

- We want to find assignments for s that maximize Q
- Looks like "all subsets" again
  - but...
- If s is the eigenvector of B with the greatest eigenvalue
  - then we'll maximize Q

Discretization

- Sounds good, but
  - entries in the eigenvector won't be +1 and -1
- Creating clusters
  - if $s_i > 0$, put node i in class 1
  - if $s_i < 0$, put node i in class 2
- General strategy
  - turn an exponential discrete problem
  - into a continuous one that can be optimized
  - move the solution back into the discrete domain
  - we will see this idea again

Multiple communities

- Keep subdividing until modularity cannot be improved
- In R
  - `cluster_leading_eigen()`

Modularity maximization

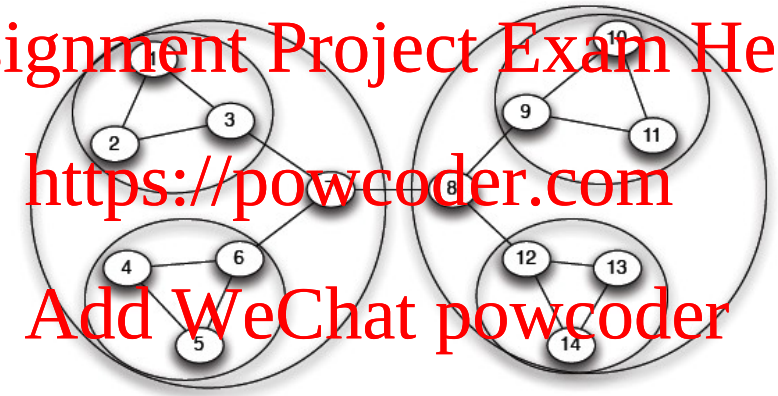
- Nice theoretical foundation
  - corresponds with intuition
- Problem
  - resolution limit
    - the larger the network, the larger the communities it finds
    - not possible to find small communities in a large network
    - this is a fundamental problem with the modularity measure
  - “resolution” parameter in Gephi
    - doesn’t really fix the problem
- Non-overlapping communities only

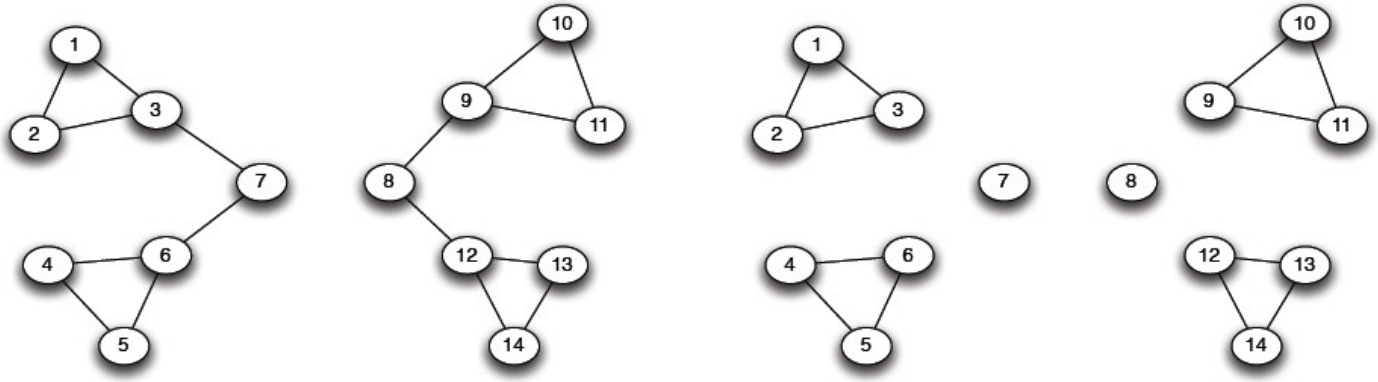
Betweenness clustering

- A hierarchical technique
- Find the edge of highest betweenness
  - this is the edge on the most shortest paths
- Remove it
- When you have disconnected the network
  - those are your communities
- Keep going until you are down to single vertices

Betweenness clustering:

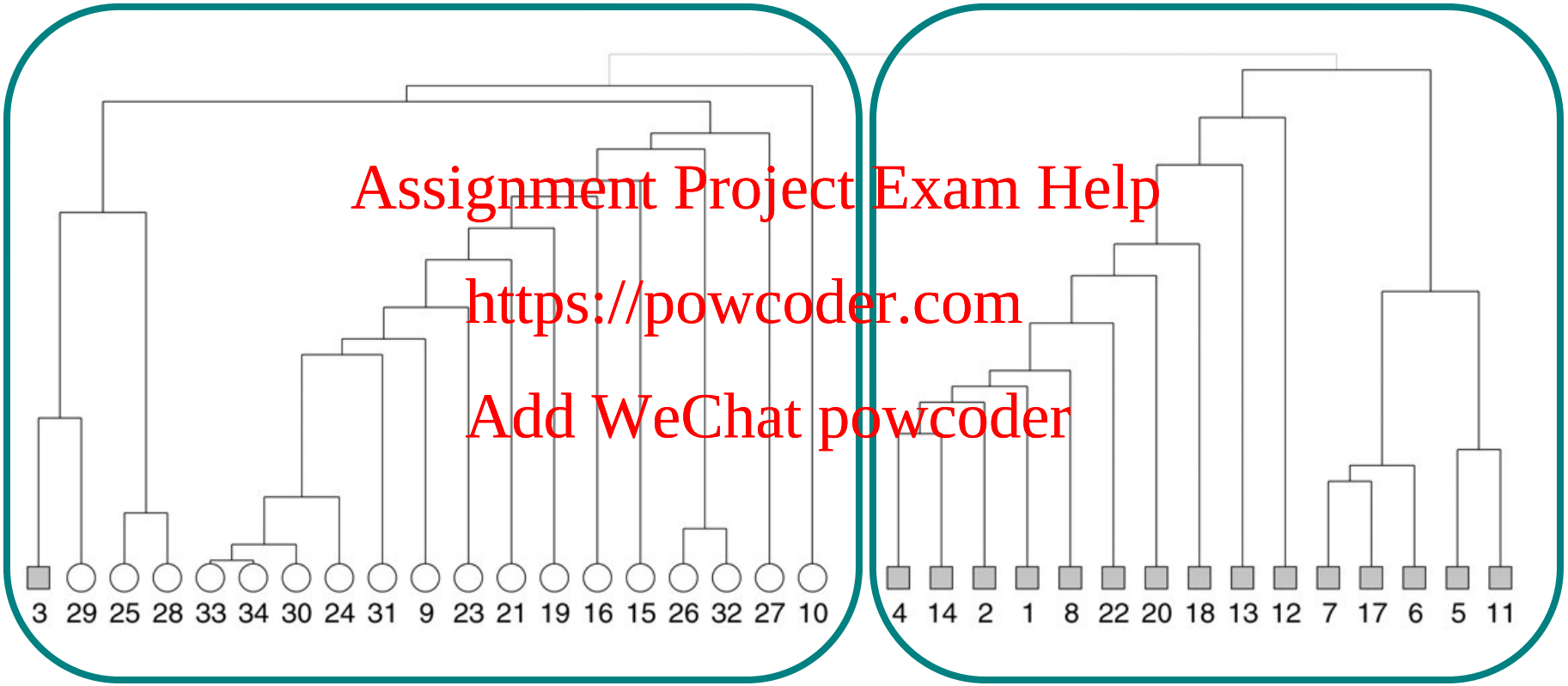
- successively remove edges of highest betweenness (the bridges, or local bridges), breaking up the network into separate components





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

betweenness clustering algorithm & the karate club data set



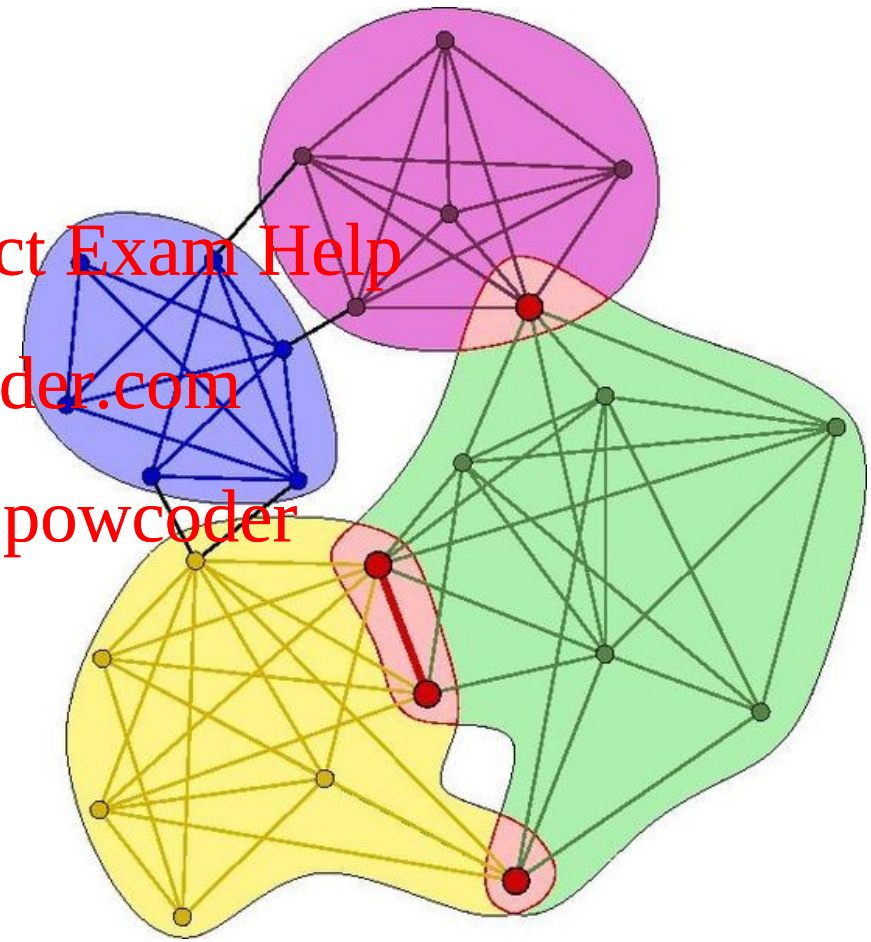
source: Girvan and Newman, PNAS June 11, 2002 99(12):7821-7826

Betweenness clustering

- In igraph
  - cluster_edge_betweenness()
- Completely different criteria from modularity
  - more global
  - may be good or bad
  - depends on the purpose of the network
- Note counter-intuitive interpretation of weights
  - Usually will want to set weights=NULL to avoid

What if communities overlap?

- Users are often in multiple communities
  - cannot use the same techniques reliably
- More open research questions
  - how to measure the quality of non-exclusive clustering
  - how to compute such clusters
- Particularly an issue in large social networks
  - modularity maximization doesn't work



Random walk

- Idea
  - short random walks will (on average) stay within the community
- Properties
  - very efficient compared to some others
  - often matches real data quite well
- Overlapping communities
  - possible to extend this idea to allow overlap
- In igraph
  - `cluster_infomap()`
  - `cluster_walktrap()`
  - `cluster_labelprop()`

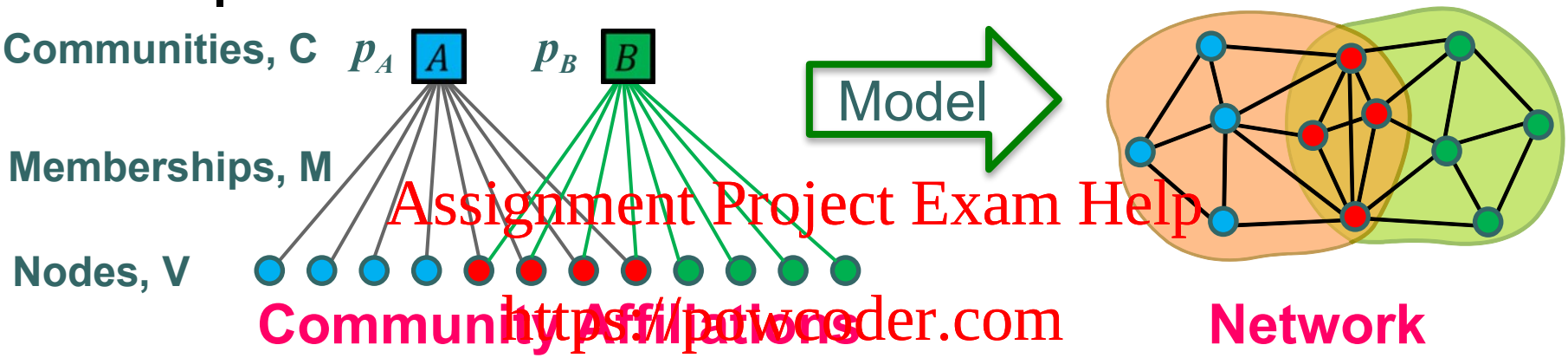
Return value

- Clustering methods return a community object
- Various methods
  - length = # of communities
  - sizes = vector of community sizes
    - (number of nodes)
  - membership = vector of community labels
  - modularity = the modularity of the clustering (even if it wasn't computed that way)

BIGCLAM

- For more see mmds.org
- A generative model of overlapping communities in networks
- Assume that the network is a projection of a bipartite network
  - Our job is to recover the latent (unknown) shared interests that generate the graph

Generative model

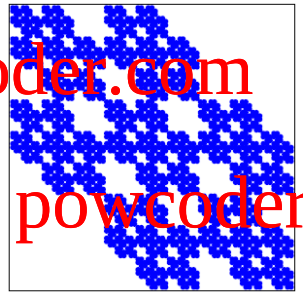


- **B(V, C, M, $\{p_c\}$) for graphs:**
  - Nodes **V**, Communities **C**, Memberships **M**
  - Each community **c** has a single probability $\mathbf{p_c}$

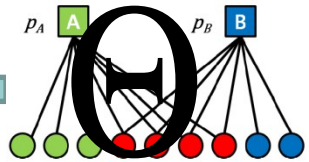
Maximum Likelihood Estimation

- **Our goal:** Find **such that:**

$$\arg\max_{\Theta} P(G \mid \Theta)$$



AGM



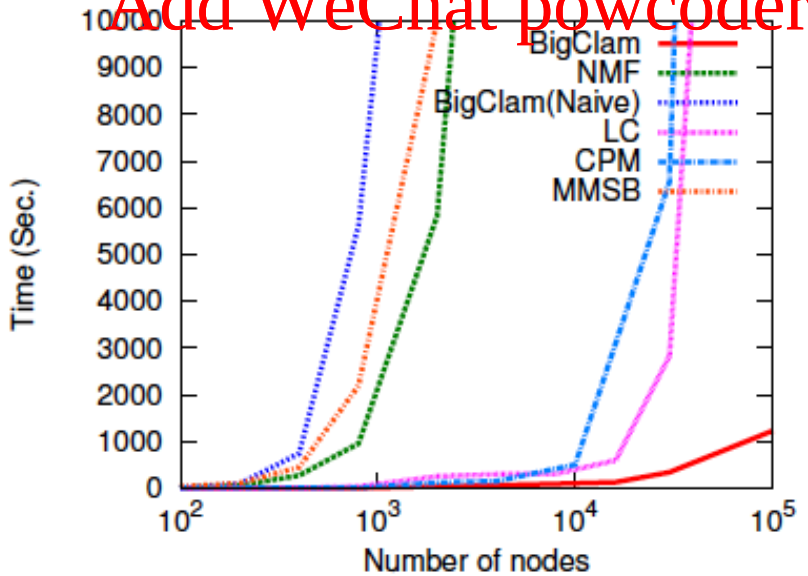
- **How do we find that maximizes the likelihood?**

Approximation

- Cannot be solved directly
- Approximation
  - Assume each user has an affiliation F_{uA} for each community A
  - Assume independence
  - Assume probability of shared community

With assumptions

- Turns into an optimization problem that can be solved with gradient descent
- Very efficient even on large networks



Comparing clusterings

- visually
- # of communities
- distribution of size
  - if there are lots of small communities, that's not good
- modularity
  - But this is not always a great measure
- Adjusted Rand index
  - computes the probability that two clusterings agree on a given pair of items
  - normalized by the expected number of agreements by chance
    - vi.dist is similar
  - Both in `mcclust` package

Conclusion

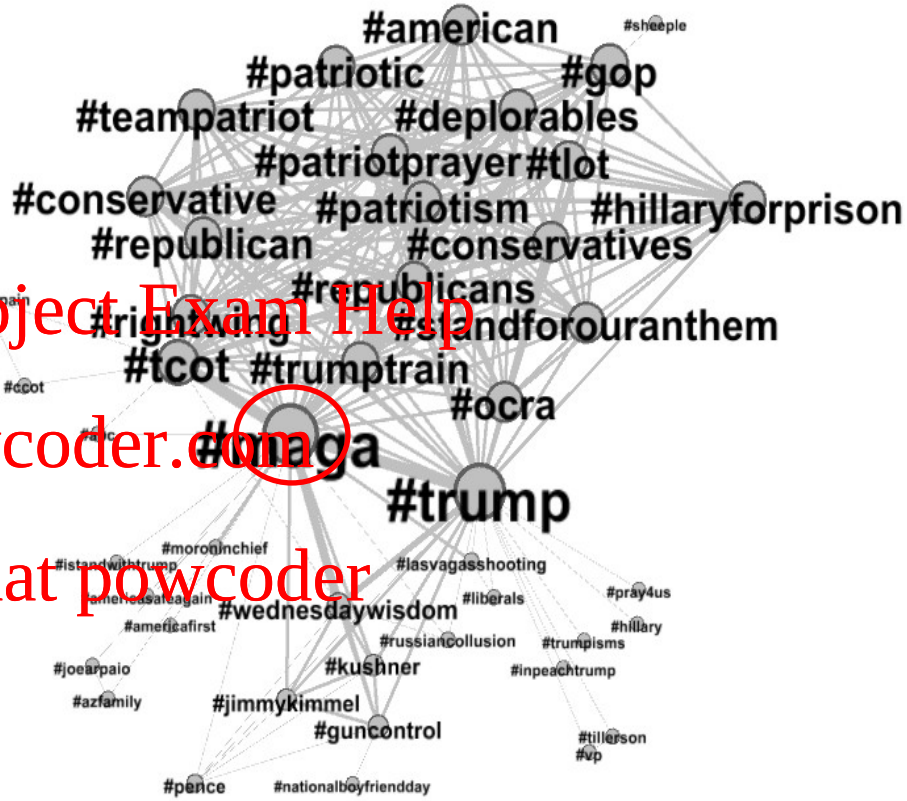
- Community structure is a way of ‘x-raying’ the network
  - seeing components with strong interactions
- Idea: discover the “natural” community boundaries
  - hard to define what this means
  - many techniques
  - an area of open research
- Analytic utility
  - identifying areas of the network with particular properties
  - subnetworks can be isolated for further analysis

Many techniques

- What to use?
- Modularity is intuitively appealing
  - but controversial
  - many studies show it does not work with real networks
    - esp. large ones
- Random walks
  - Pretty good
  - but analyst has to decide how to set walk size
- Overlapping communities
  - BIGCLAM and other techniques not implemented in igraph

Homework 5

- Hashtag-hashtag projection from twitter
- Tags connected if used by the same user
- Filter
- Community detection and analysis
- Extract communities surrounding a given node
  - Different for each community detection algorithm



Next week

- Mathematical models of networks
- Data due

Example