SOCIAL NETWORK MINING

Chris Ayala
CSE 8331 – Data Mining
March 16, 2015

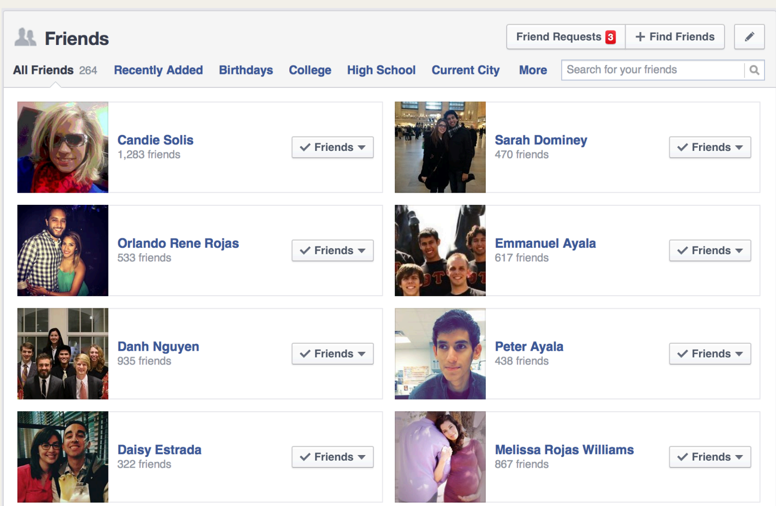
Social Networks

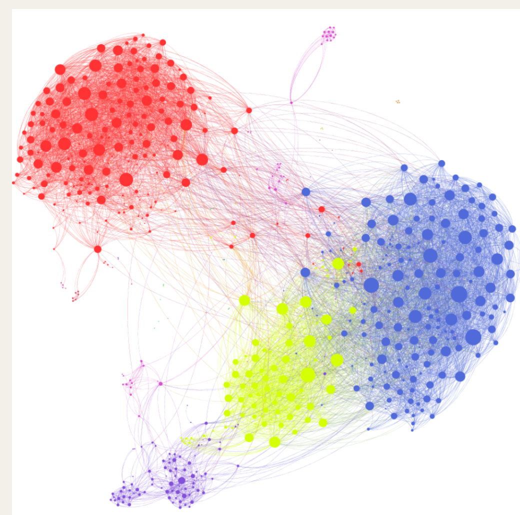
- Collection of users
- Users are somehow related to one another
- Friends, followers, likes, real-world groups

http://blog.revolutionanalytics.com/2010/12/facebooks-social-network-graph.html

Social Networks as a Graph

- Nodes represent users, edges represent relationships
- Edges can have weights (e.g. more interaction = more weight)
- Can be used to find clusters





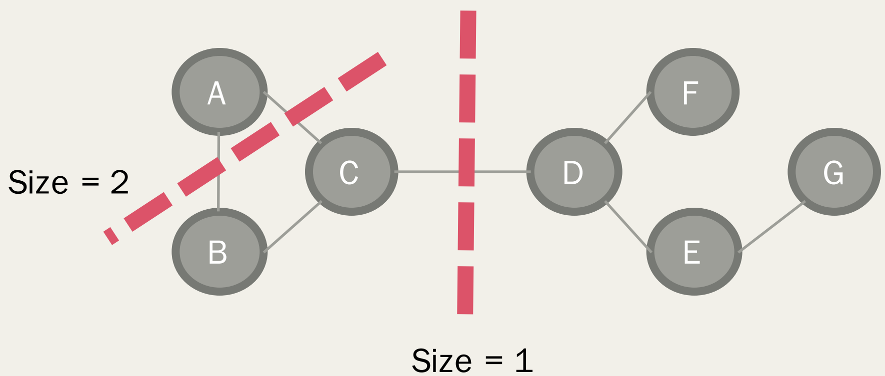
https://griffsgraphs.wordpress.com/2012/07/02/a-facebook-network/

Questions we can ask

- Based on relationships, what clusters can we detect?
- Similar people may not be friends. Can we provide recommendations?
- Would similar people be interested in similar advertising?
- Are there outliers? What do outliers represent? What constitutes an outlier?
- If lots of people have a relationship with a certain person, does this mean they would likely have a relationship with another?
- What is the average degree of separation between any two people?

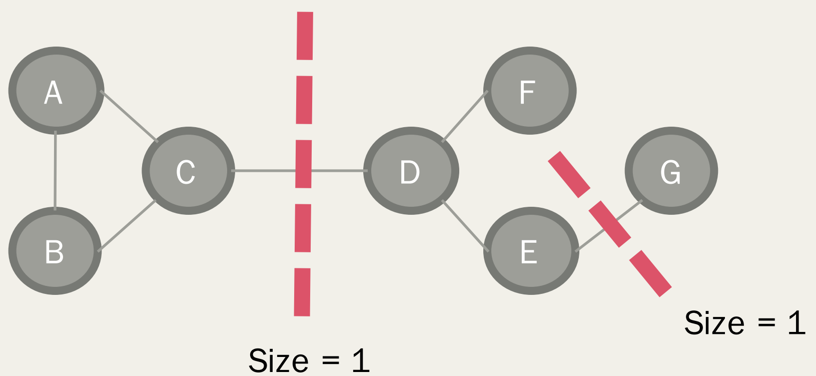
The Cut of a graph

- Defined as a partition of the graph into two sets, S and T
- A cut is a set of edges, where one node on an edge is in set S, and the other in set T
- The size of the cut is how many edges cross the cut



The Cut of a graph

- We want to minimize the size of the cut
- As in, create sets such that there are as few edges between sets as possible
- Only considers outbound edges from a set, not edges inside the sets
- Are these different? Which is better?



An improvement

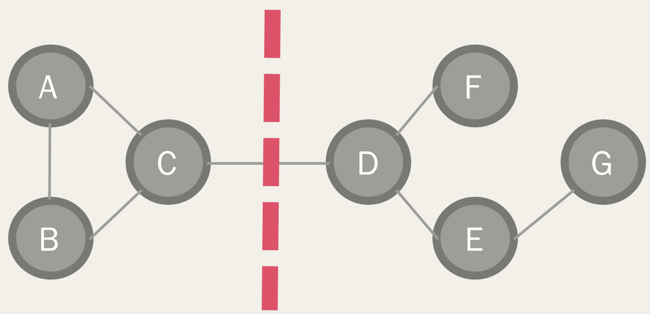
- We also want to consider the interconnectedness of a set
- Minimize the cut, maximize the “volume” of the resulting sets
- Known as the **normalized cut**
- vol(A) = sum of degrees of the nodes in A
- m = number of edges in graph

$$\phi(A) = \frac{|\{(i,j) \in E; i \in A, j \notin A\}|}{\min(vol(A),\ 2m - vol(A))} = \frac{\text{cut(A)}}{\text{vol(A)}}$$

Thus a lower Phi is better

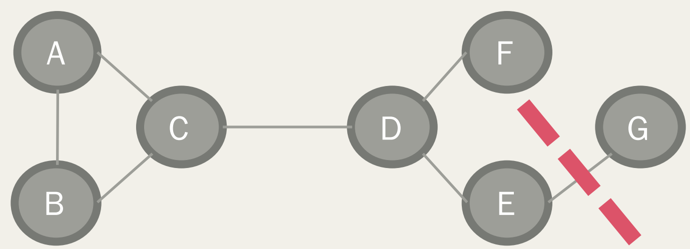
Example

$$\phi(A) = \frac{|\{(i,j) \in E; i \in A, j \notin A\}|}{\min(vol(A),\ 2m - vol(A))} = \frac{\text{cut(A)}}{\text{vol(A)}}$$



Cut(A) = 1
Vol(A) = 7

Cut(A) = 1
Vol(A) = 7



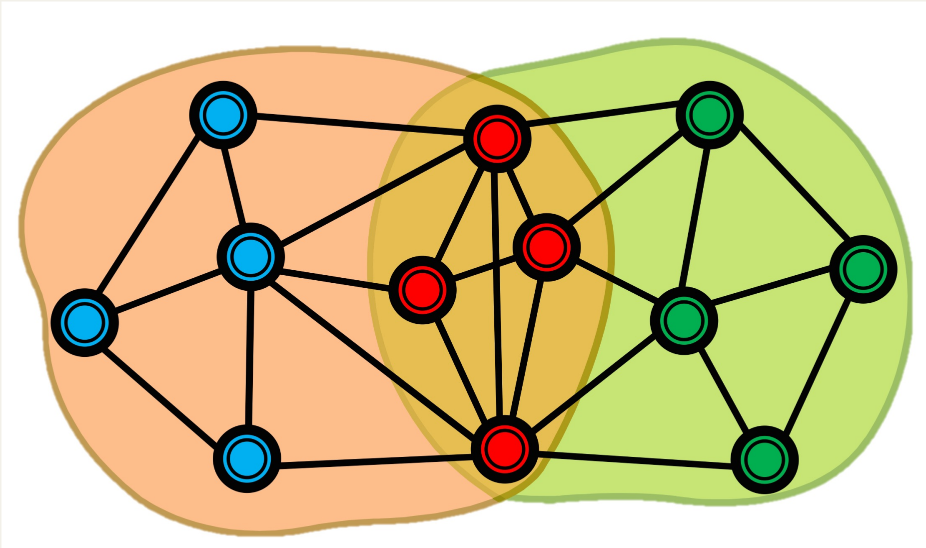
Cut(A) = 1
Vol(A) = 1

Cut(A) = 1
Vol(A) = 1

Thus the left cut should be preferred

Why this is important

- Optimizing that equation helps us find distinct groups of people
- Meant for disjoint groups. Not meant for overlaps
- How do we efficiently find groups in the first place?



Modularity

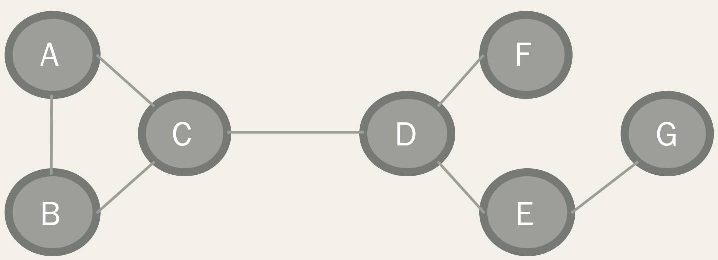
- Defined by M.E.J. Newman and M. Girvan in 2003
  – *Newman, M. E. J. & Girvan, M. (2004), 'Finding and evaluating community structure in networks', Phys. Rev. E* 69 (2) , 026113
- A means of finding communities in graphs
- “A good division of a network into communities is not merely one in which there are few edges between communities; it is one in which there are **fewer than expected** edges between communities”
  – *Newman MEJ. ‘Modularity and community structure in networks’. Proceedings of the National Academy of Sciences of the United States of America. 2006;103(23):8577-8582. doi:10.1073/pnas.0601602103.*

Modularity

- Want to find groups where number of edges in the group is higher than what we expect by random chance
  - *Another view: between-group edges is lower than random*
- Higher modularity = more likely to be a group

Adjacency Matrix

- Matrix that shows connections
- $A_{ij} = 1$ if nodes i and j are connected, 0 otherwise
- Symmetric Matrix



	A	B	C	D	E	F	G
A	0	1	1	0	0	0	0
B	1	0	1	0	0	0	0
C	1	1	0	1	0	0	0
D	0	0	1	0	1	1	0
E	0	0	0	1	0	0	1
F	0	0	0	1	0	0	0
G	0	0	0	0	1	0	0

Modularity cont.

- Suppose we permute the edges of the graph, while keeping the degree of each node unchanged
- The expected number of edges that connect i and j:
  - $e = (k_i k_j)/2m$
    - where 2m = sum of all degrees in graph
    - k_i = degree of node i
- Recall: actual number is either 0 or 1 from A
- We want to sum up (actual – expected) for each node in the set

Modularity cont.

- Suppose we divide the graph into two sets
- We define:
  - $s_i = 1$ *if node i is in set 1*
  - $s_i = -1$ *if node i is in set 2*
- Observe: $(s_i * s_j + 1)/2$
- If two nodes i and j are in the same set, then that equals 1
- Otherwise, it equals 0

Finally: Modularity Defined!

- "Modularity Q is given by the sum of $A_{ij} - k_i k_j / 2m$ over all pairs of vertices i, j that fall in the same group."
- Restated: Sum of actual (A_{ij}) minus expected ($k_i k_j / 2m$) over all pairs of vertices in the same group
- We want to maximize modularity

$$Q = \frac{1}{2m} \sum_{vw} \left[A_{vw} - \frac{k_v * k_w}{2m} \right] \frac{s_v s_w + 1}{2}$$

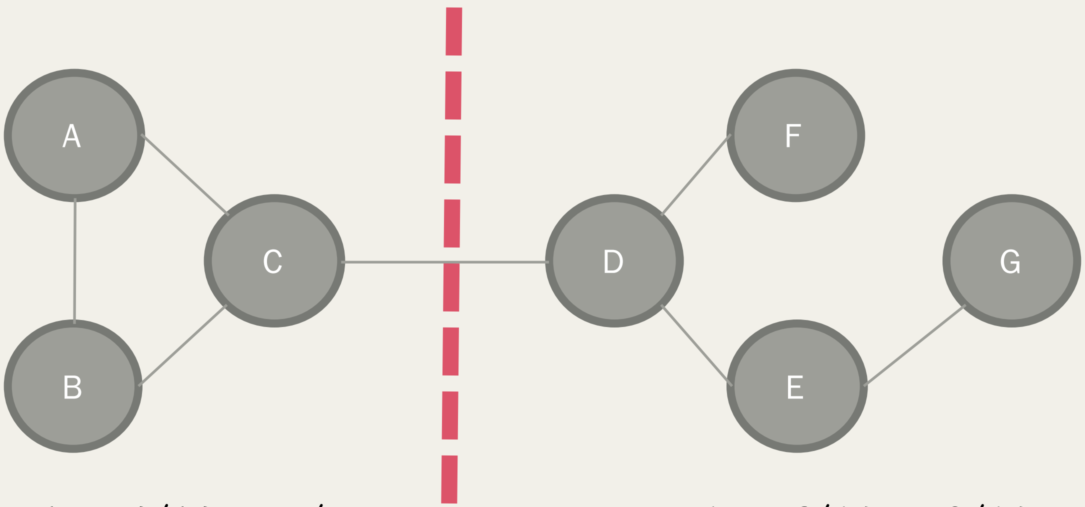
Sum over All pairs

Actual - expected

0 if different sets, 1 if in same set

Example

$$Q = \frac{1}{2m}\sum_{vw}\left[A_{vw} - \frac{k_v * k_w}{2m}\right]\frac{s_v s_w + 1}{2}$$



M_{ab} = 1 – 4/14 = 5/7
M_{ac} = 1 – 6/14 = 4/7
M_{bc} = 1 – 6/14 = 4/7

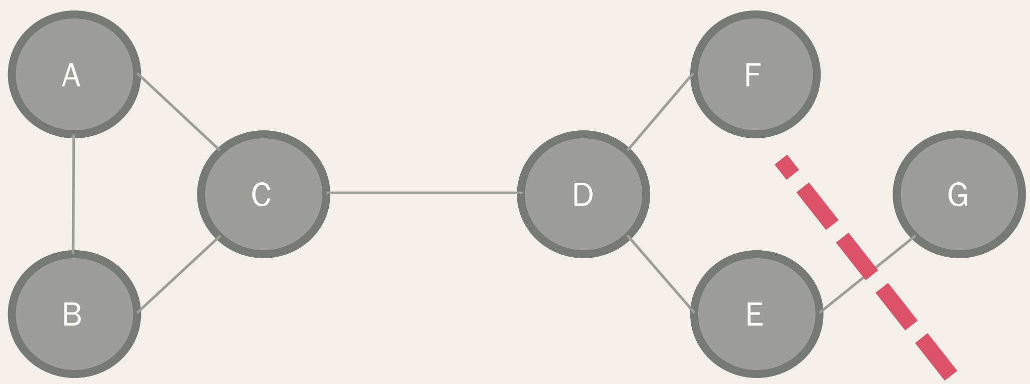
Thus Q_{s1} ~= 1.85

M_{de} = 1 – 6/14 = 8/14
M_{df} = 1 – 3/14 = 11/14
M_{dg} = 0 – 3/14 = -3/14
M_{ef} = 0 – 2/14 = -2/14
M_{eg} = 1 – 2/14 = 12/14
M_{fg} = 0 – 1/14 = -1/14

Thus Q_{s2} ~= 1.78

Example

$$Q = \frac{1}{2m}\sum_{vw}\left[A_{vw} - \frac{k_v * k_w}{2m}\right]\frac{s_v s_w + 1}{2}$$



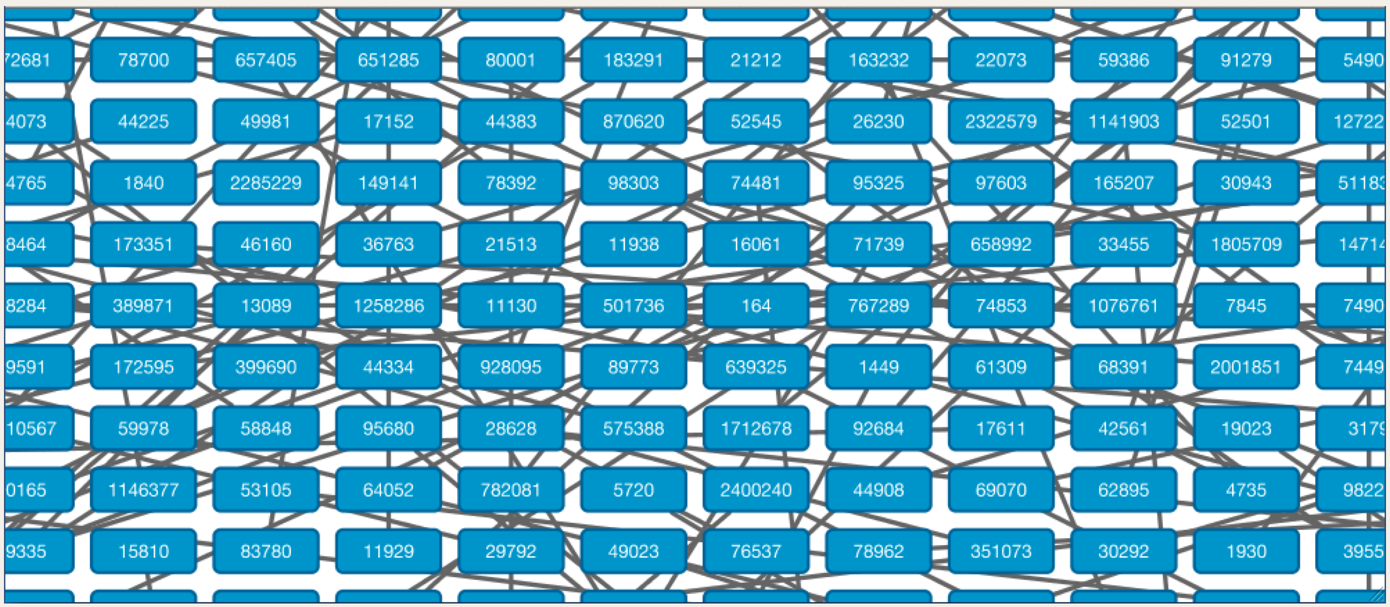
M_{ab} = 1 – 4/14 = 10/14
M_{ac} = 1 – 6/14 = 8/14
M_{ad} = 0 – 6/14 = -6/14
M_{ae} = 0 – 4/14 = -4/14
M_{af} = 0 – 2/14 = -2/14
M_{bc} = 1 – 6/14 = 8/14
M_{bd} = 0 – 6/14 = -6/14
M_{be} = 0 – 4/14 = -4/14
M_{bf} = 0 – 2/14 = -2/14

M_{cd} = 1 – 9/14 = 5/14
M_{ce} = 0 – 6/14 = -6/14
M_{cf} = 0 – 3/14 = -3/14
M_{de} = 1 – 6/14 = 8/14
M_{df} = 1 – 3/14 = 11/14
M_{ef} = 0 – 2/14 = -2/14

Thus, Q_{s1} ~= 1.07

Problems at scale

- Social networks often have millions of active users
- Finding the optimal cut is computationally difficult
  - *Modularity helps*
- Visualizing can be problematic



Visualization Tools

http://gephi.github.io

http://www.cytoscape.org

Sample Datasets

- Facebook Netvizz – Can be used to download a graph of your personal network
  - *Alternatively: GetNet (http://snacourse.com/getnet)*
  - *Will use this in a demo shortly*
- Arizona State University Social Computing: http://socialcomputing.asu.edu/pages/datasets
- Stanford Large Network Dataset Collection: https://snap.stanford.edu/data/
- Recommended Reading: “Mining of Massive Datasets”
  - *Jure Leskovec, Anand Rajaraman, Jeff Ullman*
  - *http://www.mmds.org*

Demo

Thanks!