Community detection and cascades

Rik Sarkar

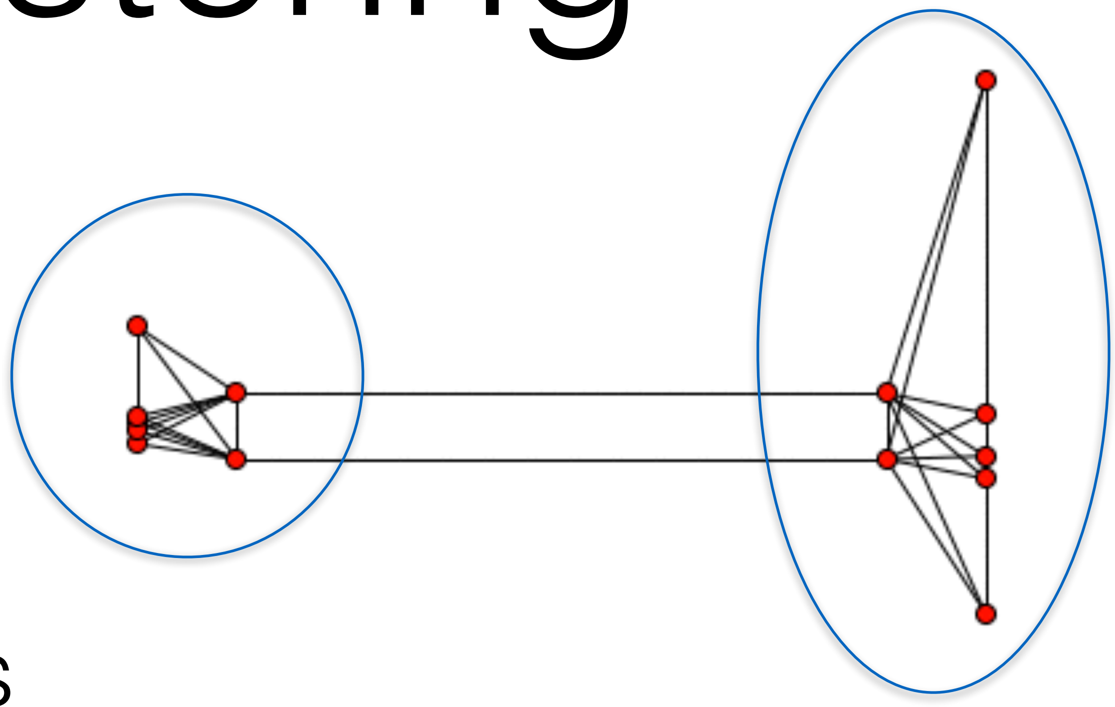
Today

- Community Detection
  - Spectral clustering
  - Overlapping community detection
  - Cascades

Spectral clustering

- Clustering or community detection using eigen vectors of the laplacian
- Standard clustering algorithms assume a Euclidean space
- Many types of data do not have Euclidean coordinates
  - Often, they come from other spaces,
  - Or we are given just a notion of “similarity” or “distance” of items

Spectral clustering



- Idea:
- Compute a graph from the similarity or distance measures
- Use the eigen vectors of the graph to embed in a euclidean space.
- Cluster using standard methods

Spectral clustering

- Essentially developed for graphs/networks
- Applies to many types of data
- Even where standard methods do not apply

- Ideas from networks are easy to apply to many other cases

Spectral clustering

- Basic algorithm: Finding k clusters
- Represent data as graph: connect edges between “similar” nodes
- Compute laplacian L
- Compute first k eigen vectors of L
  - Remember: Each vector contains a value for each node
- Embed the nodes in $\mathbf{R}^k$ using their values in the eigen vectors
- Apply k-means or other euclidean clustering

Why spectral clustering works

- Laplacian L = D - A
- For a real vector x: $$x^T L x = \sum_{(i,j)\in E} (x_i - x_j)^2$$
- And $$\lambda_1 = \min \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum x_i^2}$$

Rayleigh Theorem

$$\lambda_1 = \min \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum x_i^2}$$

- Min achieved when x is a unit eigen vector e_1 (Fiedler vector)
- $$\sum x_i^2 = 1$$
- Since x is orthogonal to e_0= [1,1,1,…],

$$\sum x_i = 0$$

$$\lambda_1 = \min_{\sum x_i = 0} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum x_i^2}$$

- In x, some components +ve, some -ve
- Min achieved when number of edges across zero are minimized
  - A good “cut”

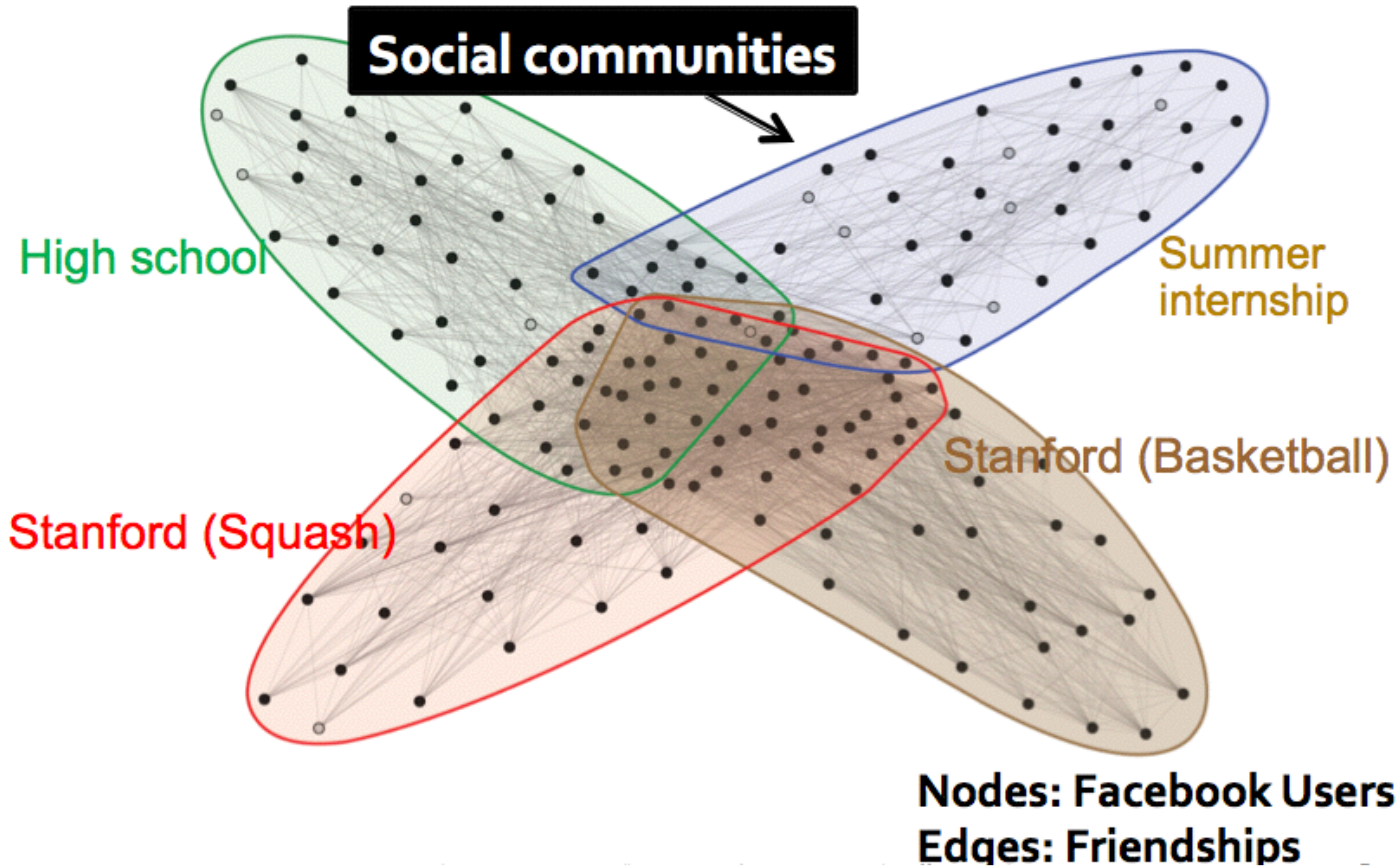
Variants of Spectral clustering

- It is possible to use other types of laplacians called normalized Laplacians
  - Give slightly different approximation properties in terms of optimizing cuts

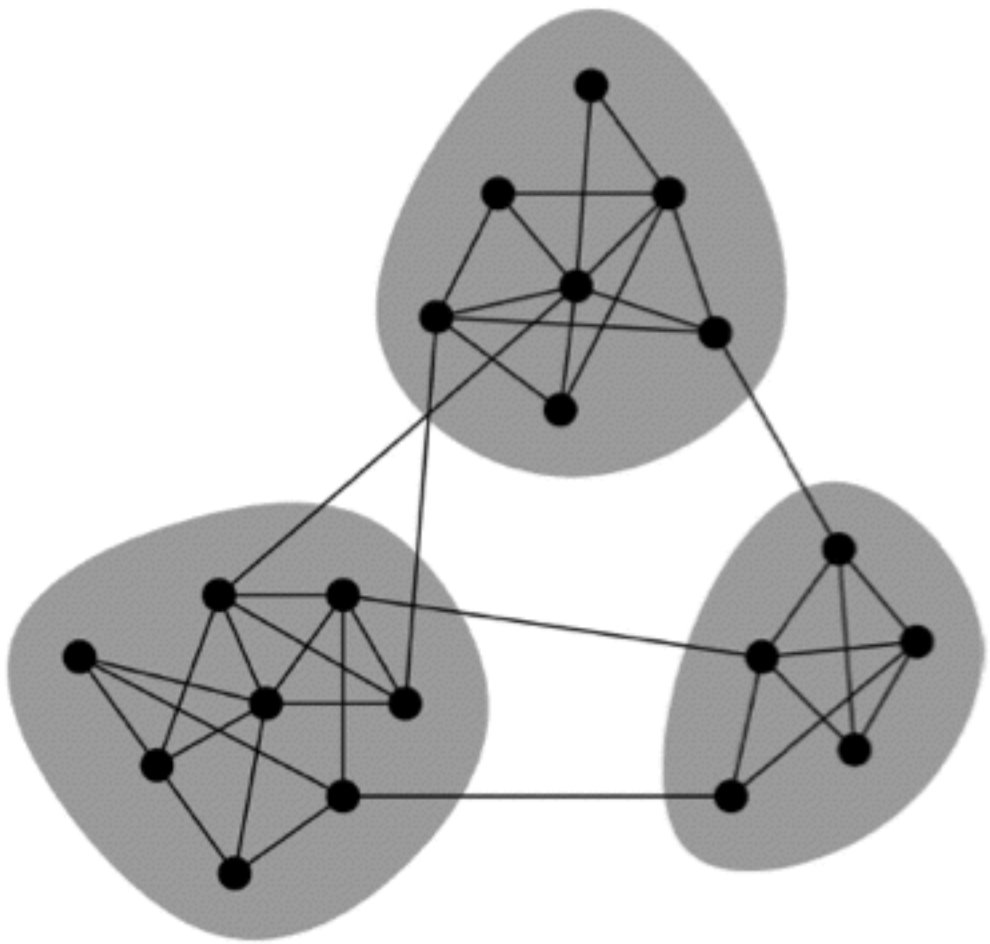
$$L_{\text{sym}} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$
$$L_{\text{rw}} := D^{-1} L = I - D^{-1} W.$$

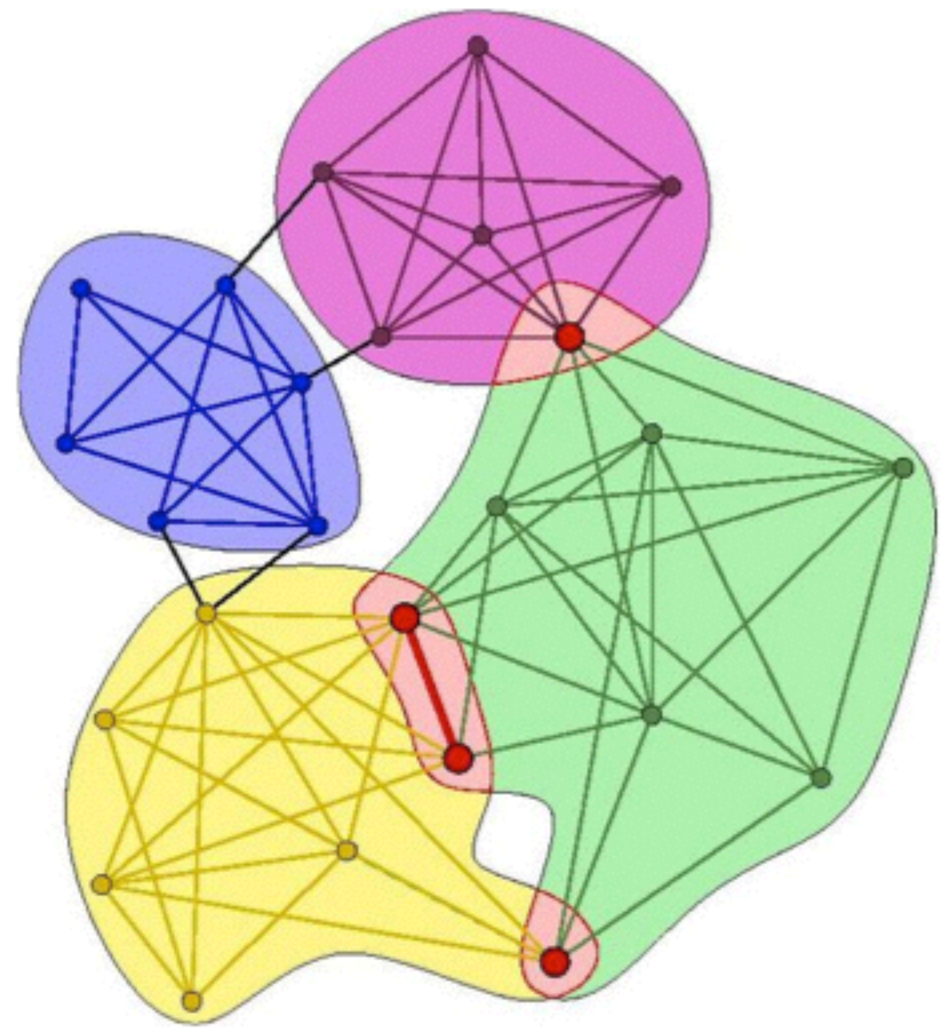
- For more details, see : Luxburg, Tutorial on Spectral Clustering
- **Note: Eigen vectors are sometimes written differently**
  - We started count at 0, some authors start at 1.
  - Then the Fiedler vector will be e_2 and the eigen value is λ_2

Overlapping communities

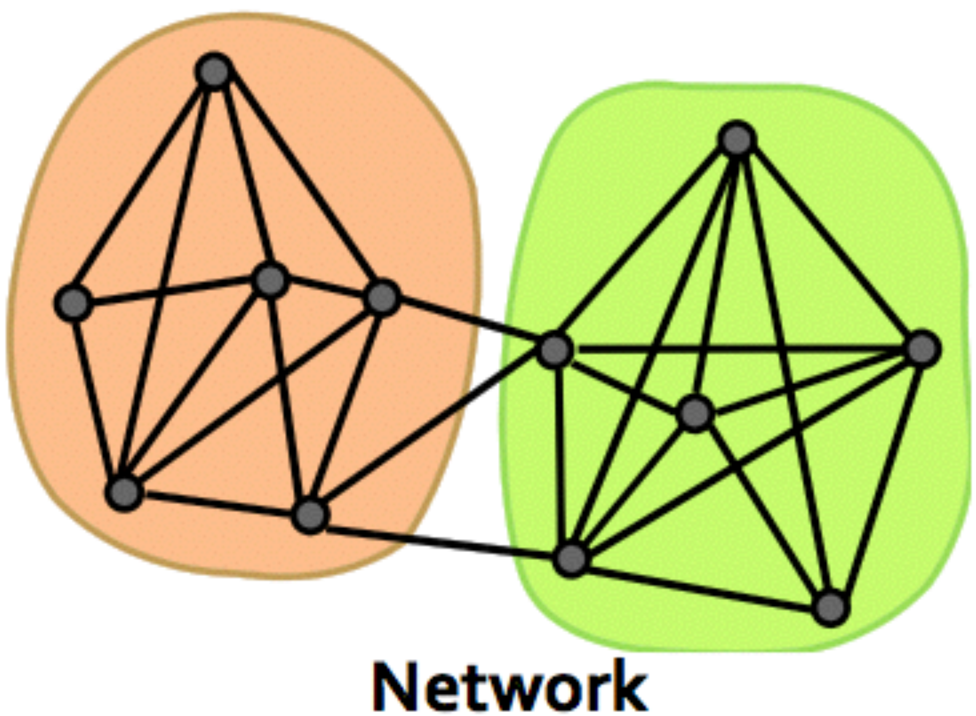


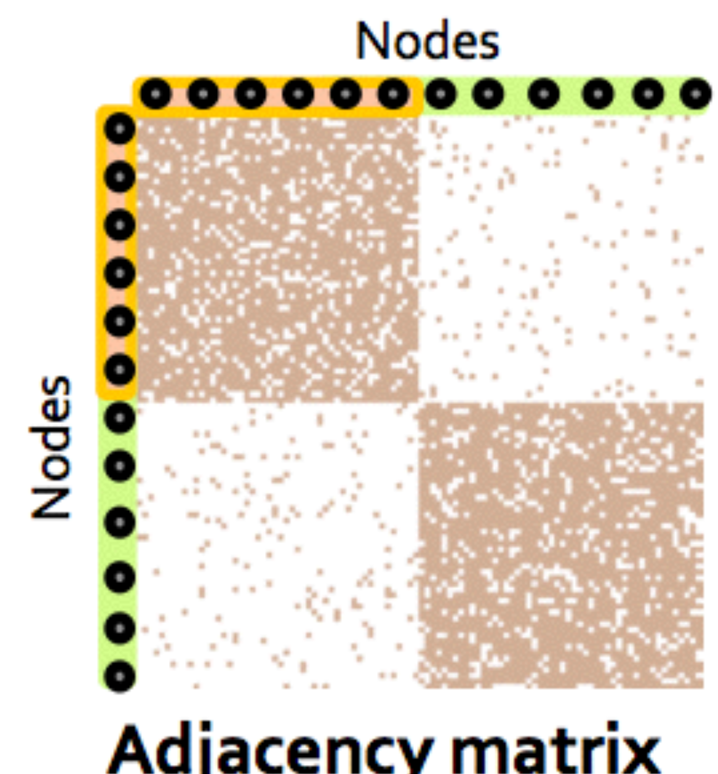
- Non-overlapping vs. overlapping communities



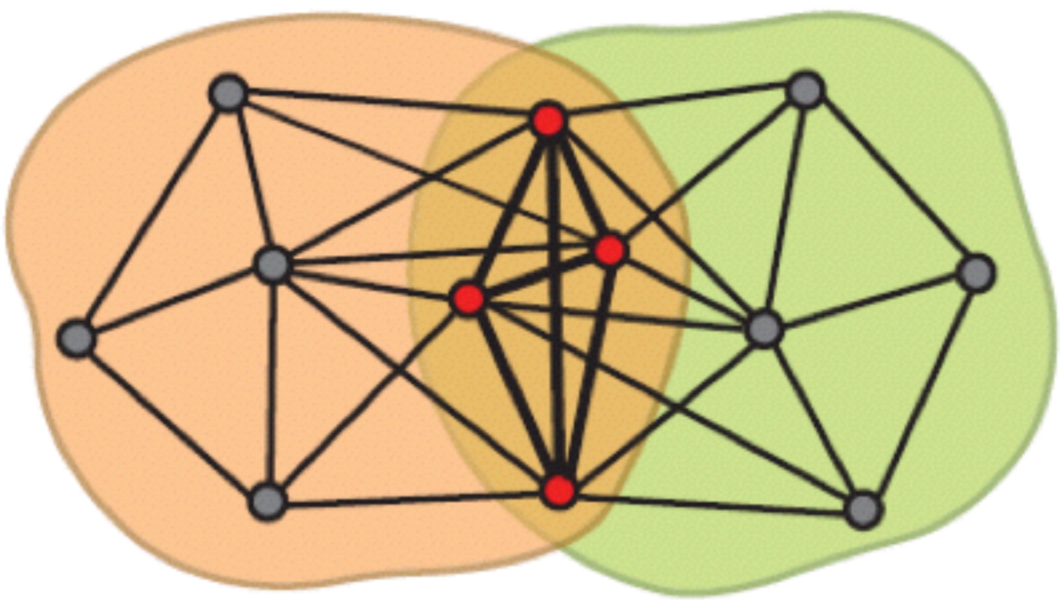


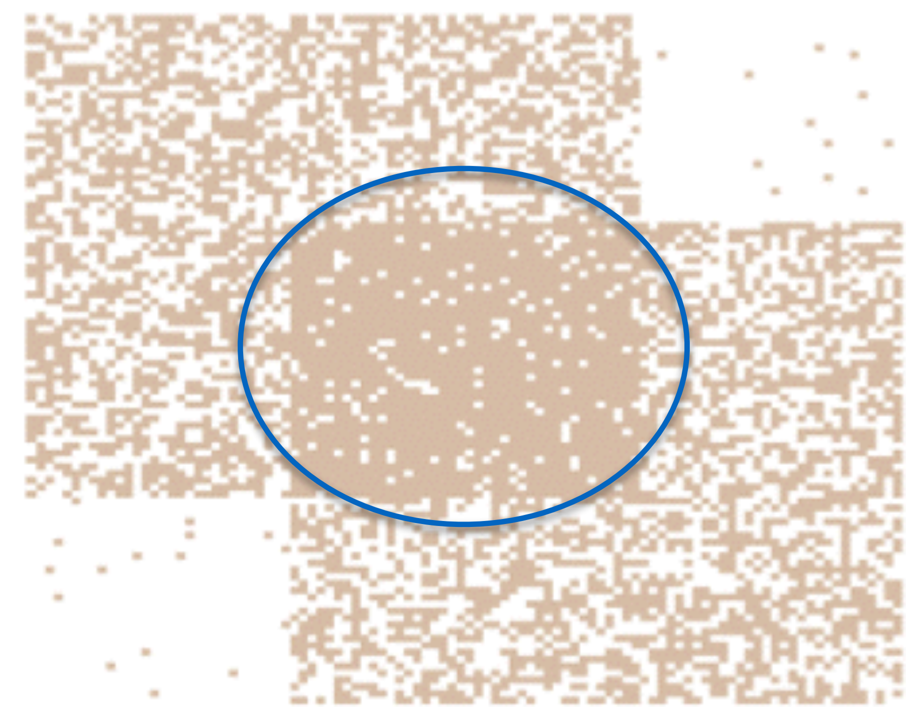
Non-Overlapping communities



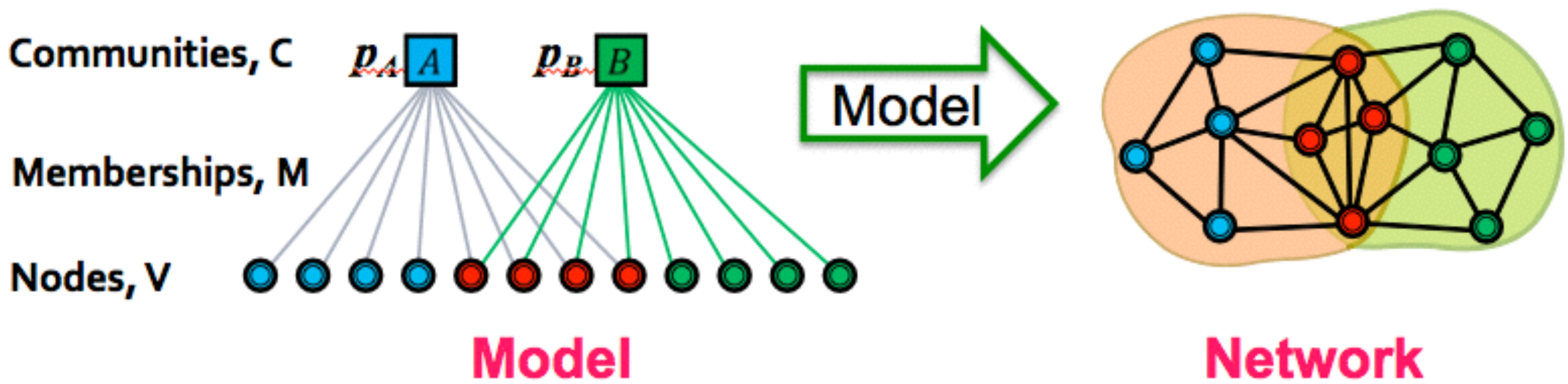


Overlapping communities



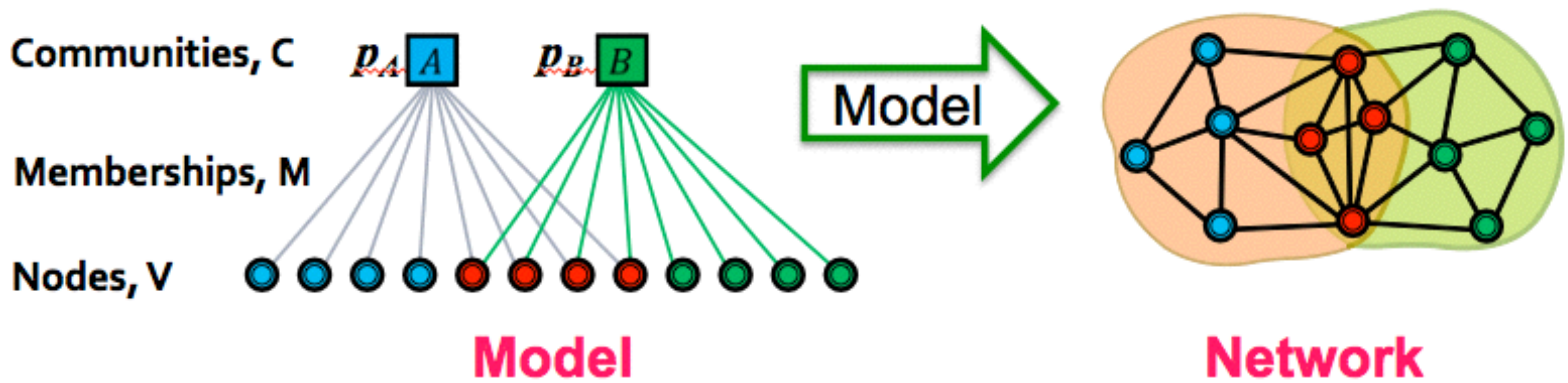


Affiliation graph model



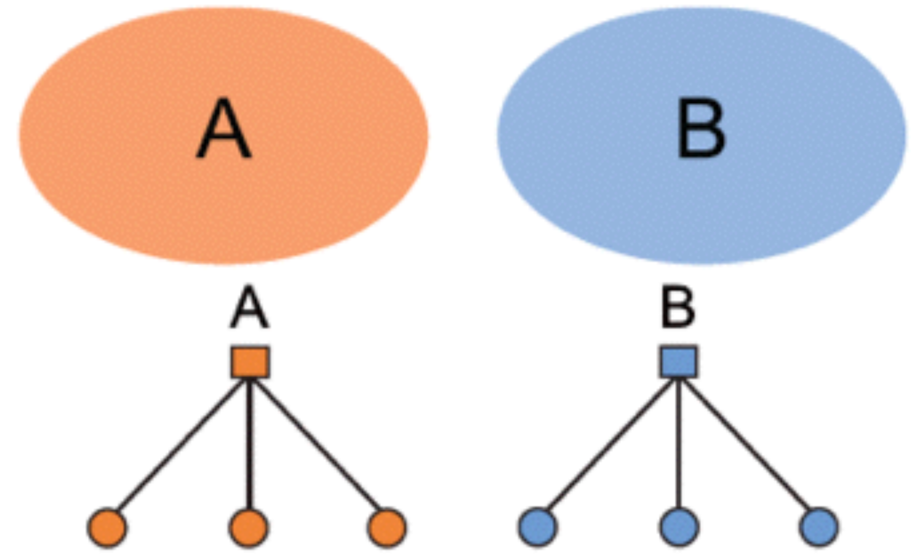
- Generative model:
- Each node belongs to some communities
- If both A and B are in community c
    - Edge (A, B) is created with probability p_c

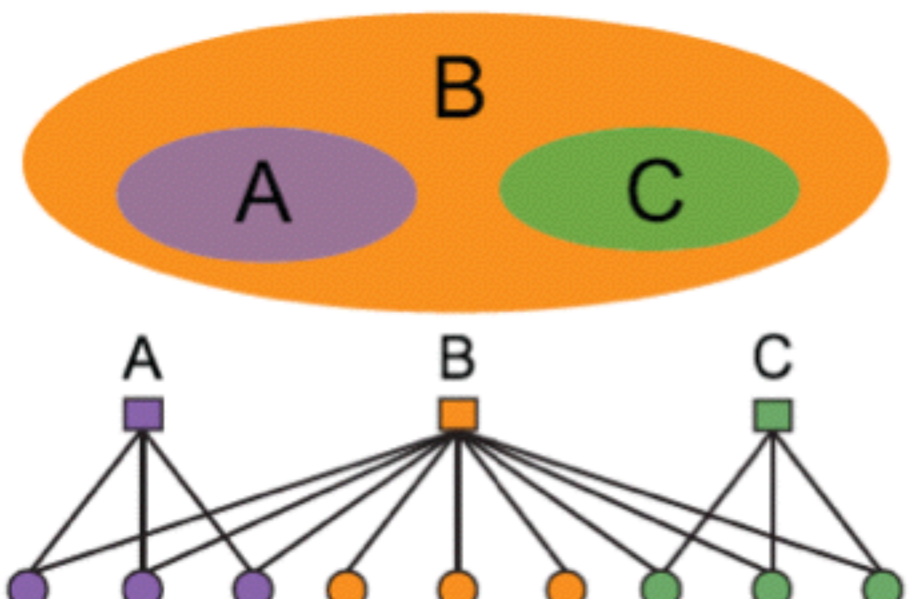
Affiliation graph model

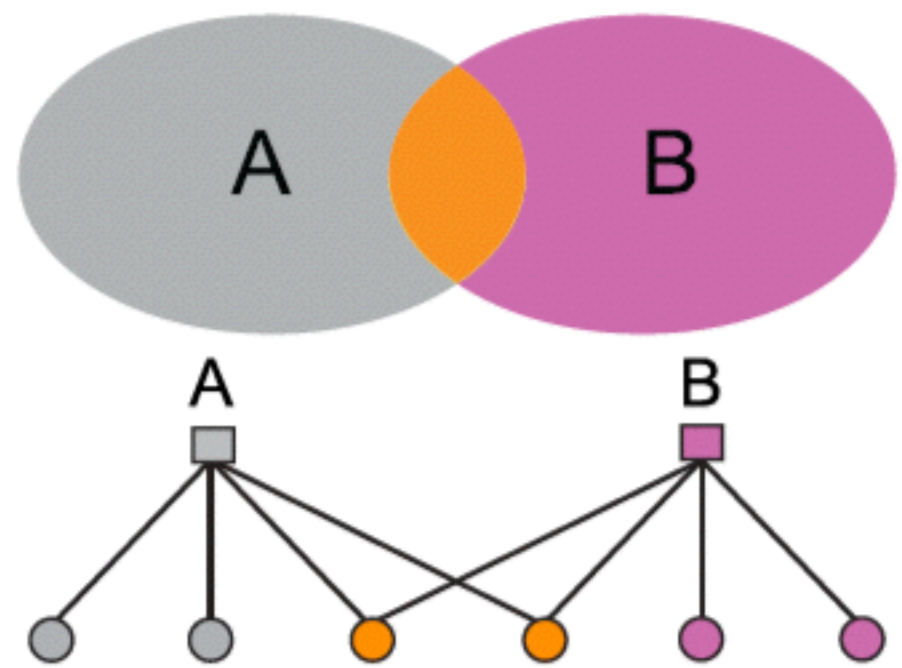


- Problem:
  - Given the network, recover:
    - Communities: C
    - Memberships or Affiliations: M
    - Probabilities: p_c

- **AGM can express a variety of community structures:** Non-overlapping, Overlapping, Nested







Maximum likelihood estimation

- Given data X
- Assume data is generated by some model f with parameters Θ
- Express probability $P[f(X|\ \Theta)]$: f generates X, given specific values of Θ.
- Compute $\mathrm{argmax}_{\Theta}\ (P[f(X|\ \Theta)])$

MLE for AGM: The BIGCLAM method

- Finding the best possible bipartite network is computationally hard (too many possibilities)
- Instead, take a model where memberships are real numbers: Membership strengths
  - F_{uA} Strength of membership of u in A
  - $P_A(u,v) = 1 - \exp(-F_{uA}.F_{vA})$: Each community links independently, by product of strengths
  - Total probability of an edge existing:
    - $P(u,v) = 1 - \Pi_C(1 - P_c(u,v))$

BIGCLAM

- Find the F that maximizes the likelihood that exactly the right set of edges exist.
- Details Omitted
- Optionally, See
- Overlapping Community Detection at Scale: A Nonnegative Matrix Factorization Approach by J. Yang, J. Leskovec. *ACM International Conference on Web Search and Data Mining (WSDM)*, 2013.

Network cascades

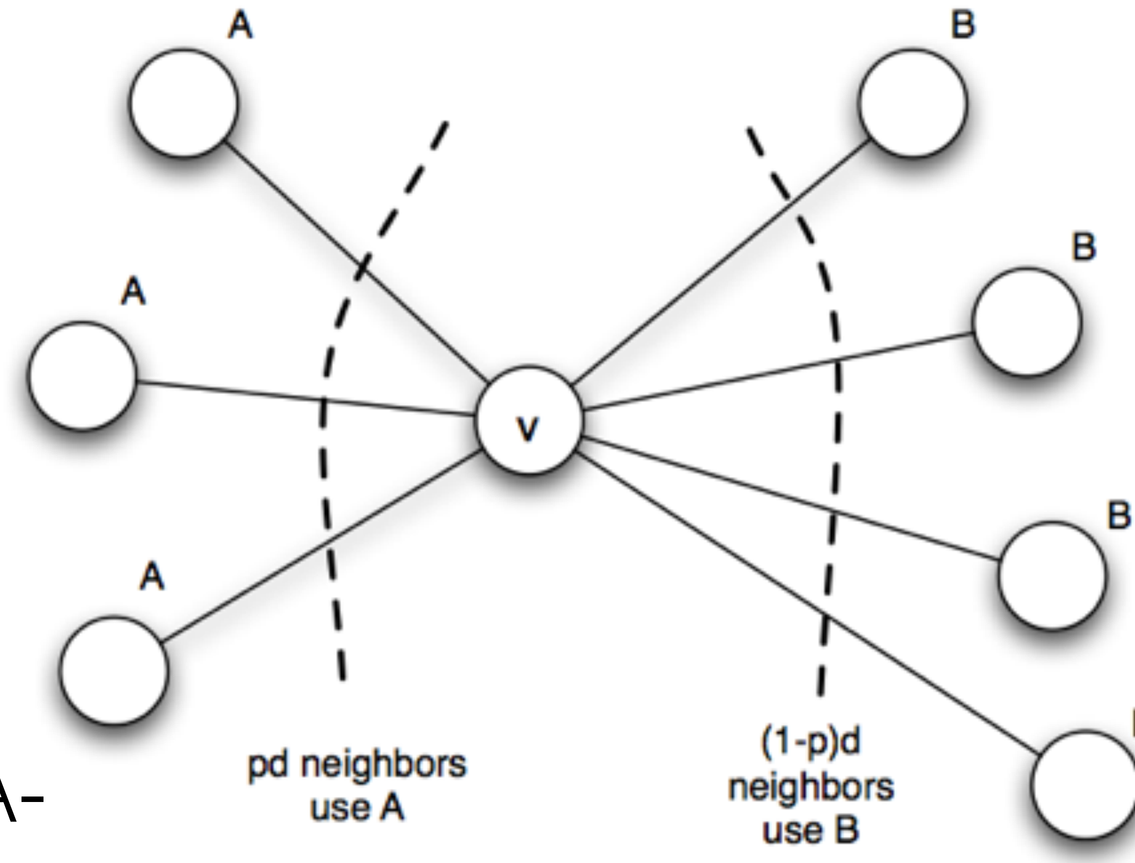
- Things that spread (diffuse) along network edges
- Innovation:
  - We use technology our friends/colleagues use
  - Compatibility
  - Information/Recommendation/endorsement

Models

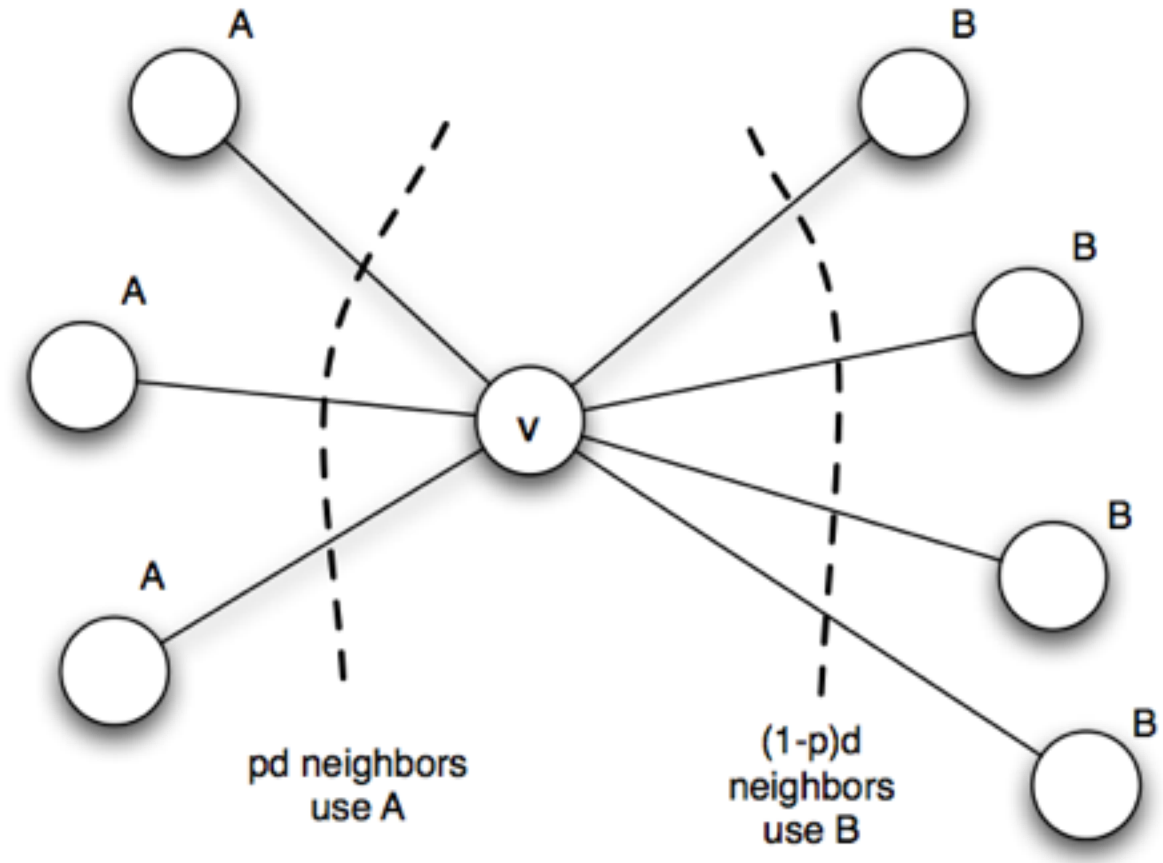
- Basic idea: Your benefits of adopting a new behavior increases as more of your friends adopt it
  - Technology, beliefs, ideas… a “contagion”

A Threshold

- v has d edges
- p fraction use A
- (1-p) use B
- v's benefit in using A is a per A-edge
- v's benefit in using B is b per B-edge



Threshold



- A is a better choice if:

$$pda \geq (1-p)db,$$

- or:

$$p \geq \frac{b}{a+b}.$$

The contagion threshold

- Let us write q = b/(a+b)
- If q is small, that means b is small relative to a
  - Therefore a is useful even if only a small fraction is using it
- If q is large, that means the opposite is true, and B is a better choice

Cascading behavior

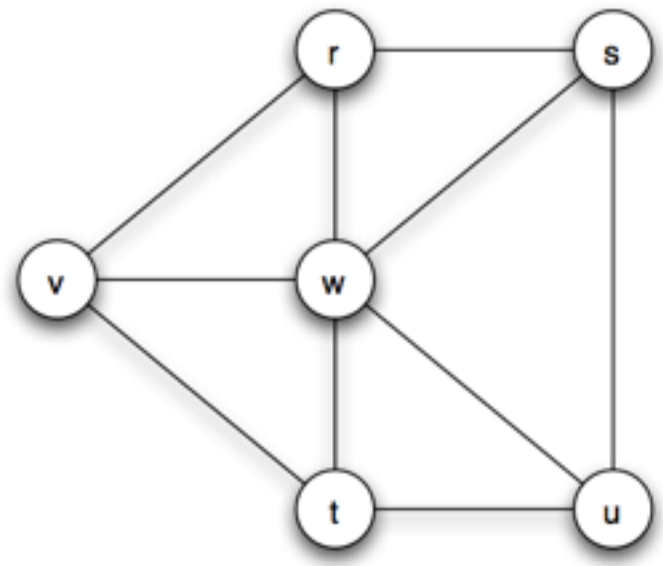
- If everyone is using A (or everyone is using B)
- There is no reason to change — equilibrium
- If both are used by some people, the network state may change towards one or the other.
  - Cascades: We want to understand how likely that is.
- Or there may be a deadlock
  - Equilibrium: We want to understand what that may look like

Cascades

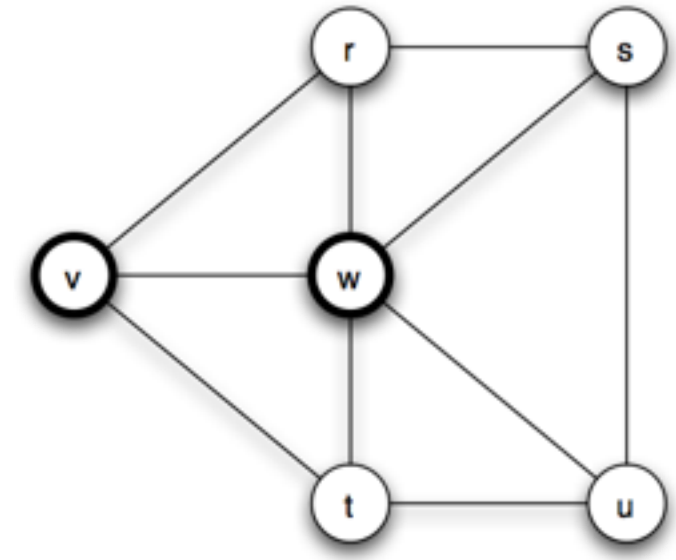
- Suppose initially everyone uses B
- Then some small number adopts A
  - For some reason outside our knowledge
- Will the entire network adopt A?
- What will cause A’s spread to stop?

Example

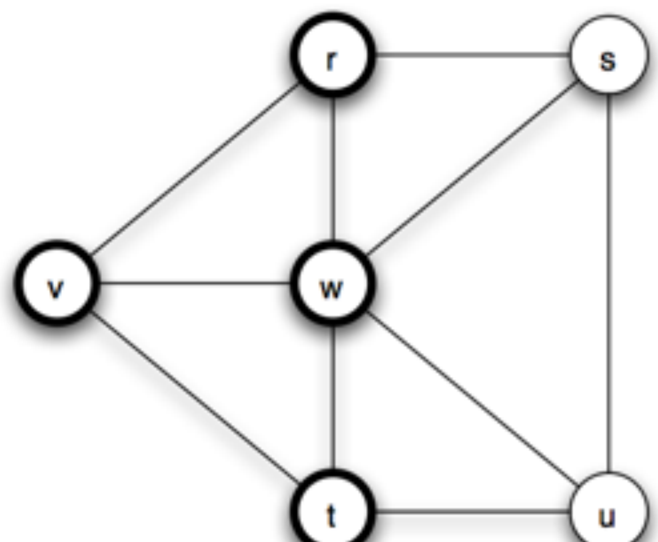
- a =3, b=2
- q = 2/5



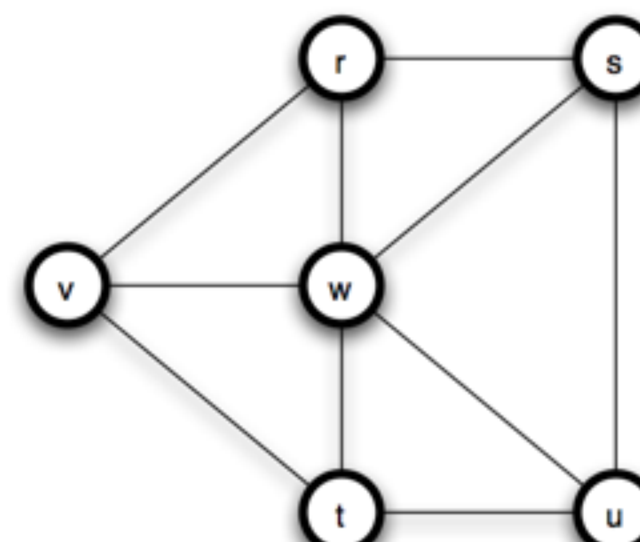
(a) *The underlying network*



(b) *Two nodes are the initial adopters*



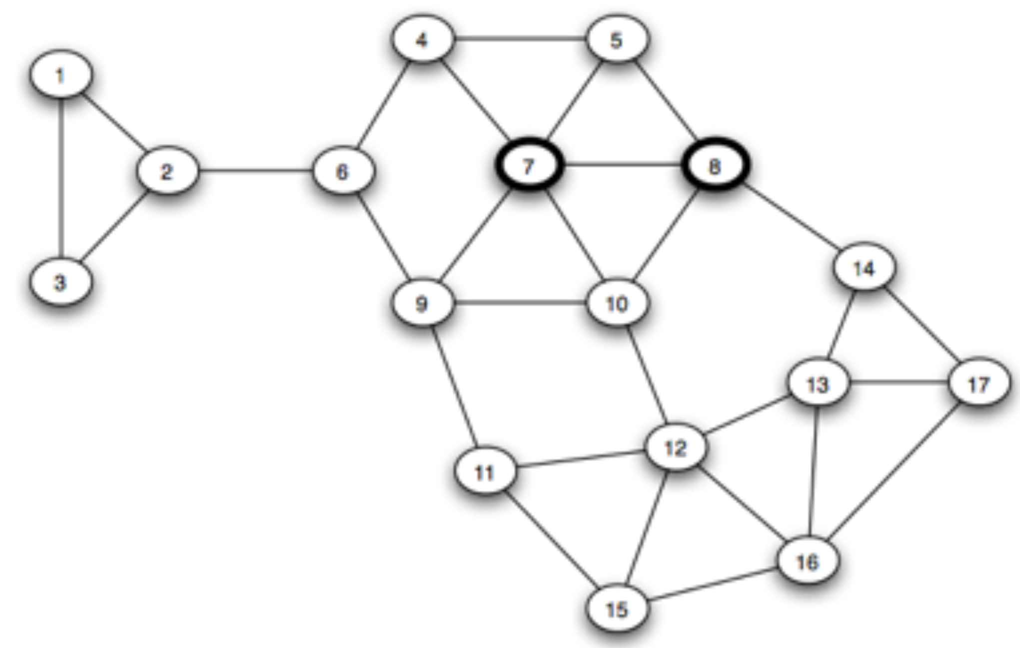
(c) *After one step, two more nodes have adopted*



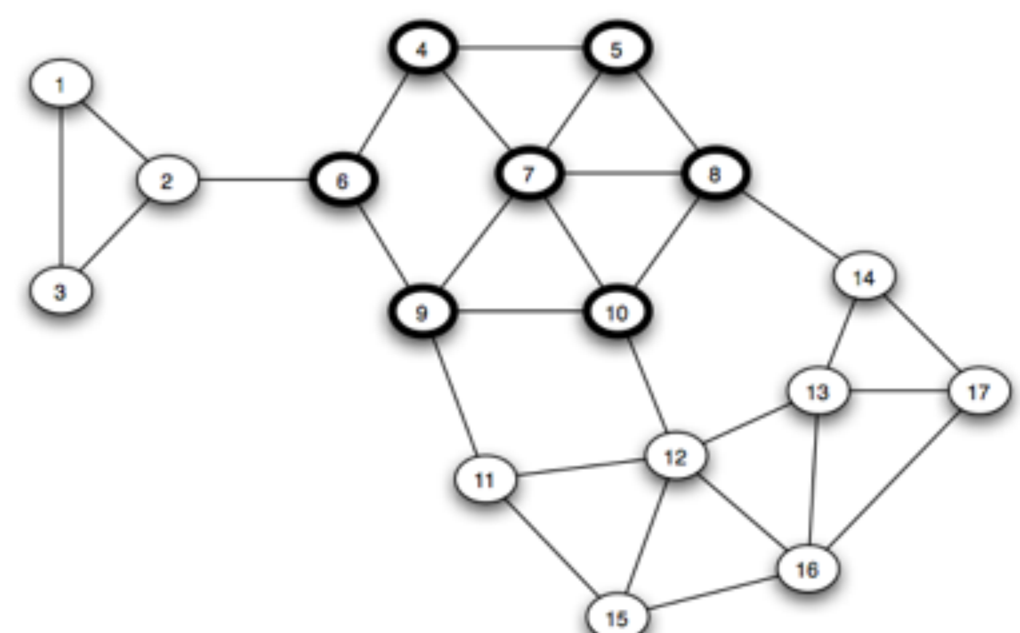
(d) *After a second step, everyone has adopted*

Example

- a =3, b=2
- q = 2/5



(a) *Two nodes are the initial adopters*



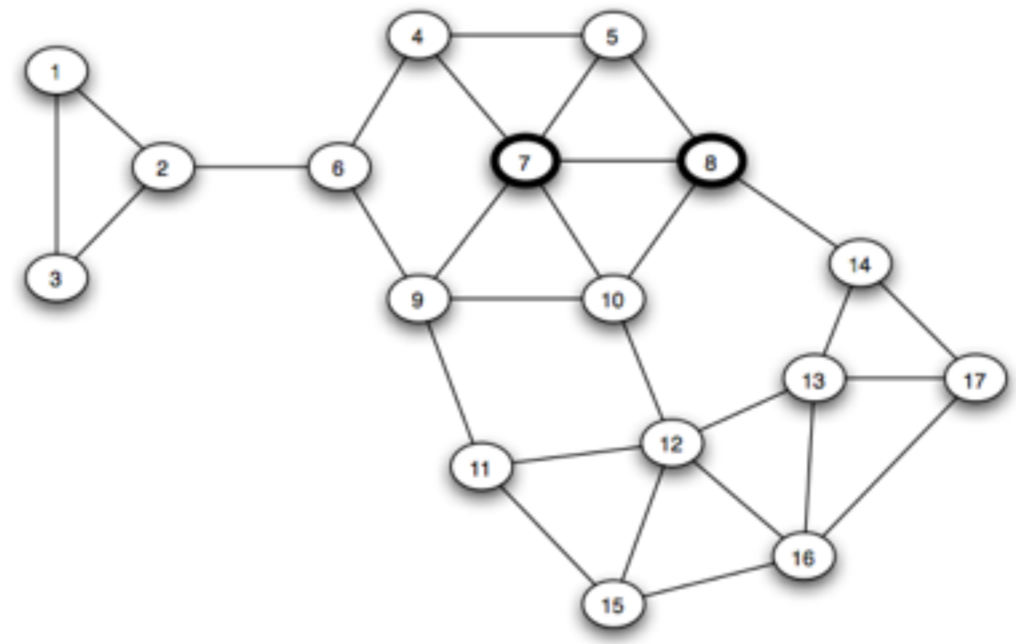
(b) *The process ends after three steps*

Stopping of spread

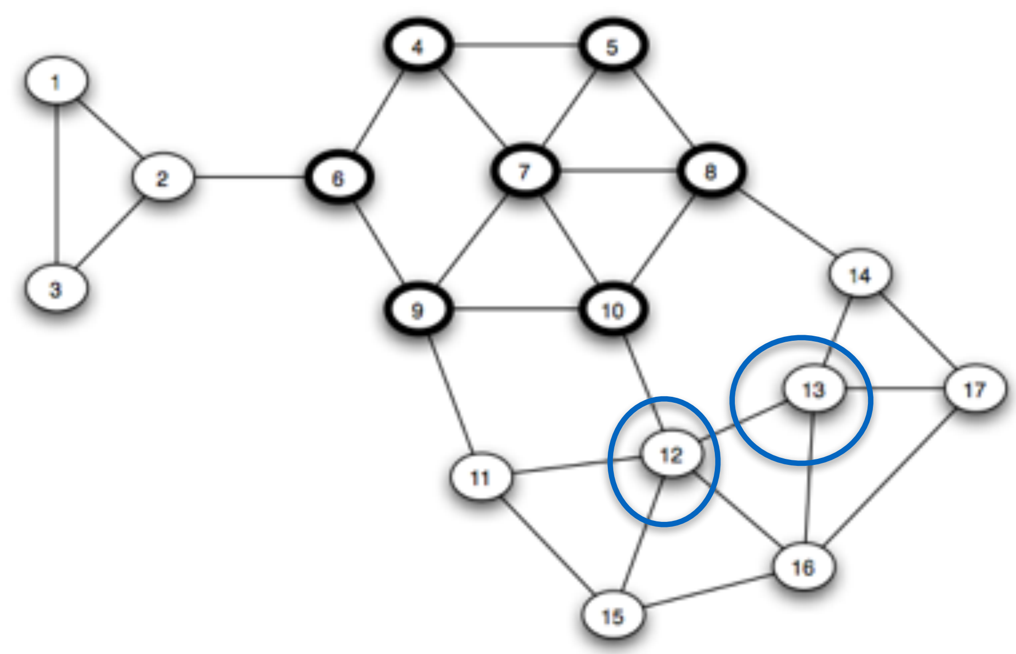
- Tightly knit communities stop the spread
- Weak links are good for information transmission, not for behavior transmission
- Political conversion is rare
- Certain social networks are popular in certain demographics
- You can defend your “product” by creating tight communities among users

Spreading innovation

- A can be made to spread more by making a better product,
  - say a = 4, then q = 1/3
  - and A spreads
- Or, convince some key people to adopt A
  - node 12 or 13



(a) *Two nodes are the initial adopters*



(b) *The process ends after three steps*