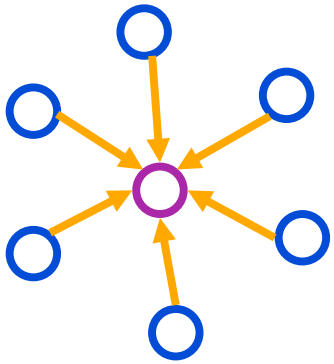
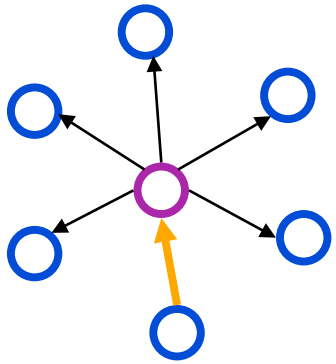
School of Information
University of Michigan

Lab: prestige

Prestige in Pajek

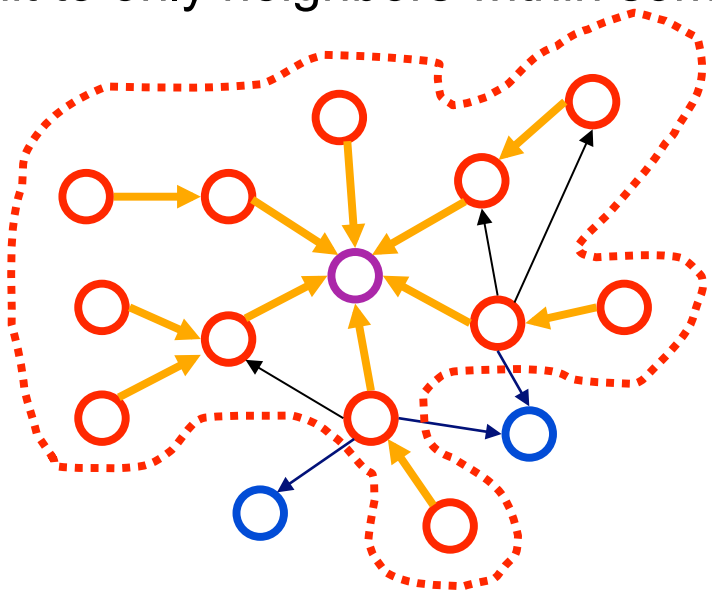
- Calculating the indegree prestige
  - Net>Partition>Degree>Input
  - to view, select File>Partition>Edit
  - if you need to reverse the direction of each tie first (e.g. lends money to -> borrows from): Net>Transform>Transpose





Prestige in Pajek

- Influence range (a.k.a. input domain)
  - Net>k-Neighbours>Input
    - enter the number of the vertex, and 0 to consider all vertices that eventually lead to your chosen vertex
    - to find out the size of the input domain, select Info>Partition
  - Calculate the size of the input domains for all vertices
    - Net>Partitions>Domain>Input
  - Can also limit to only neighbors within some distance



Proximity prestige in Pajek

- Direct nominations (choices) should count more than indirect ones
- Nominations from second degree neighbors should count more than third degree ones
- So consider proximity prestige

$$C_p(n_i) = \frac{\text{fraction of all vertices that are in i's input domain}}{\text{average distance from i to vertex in input domain}}$$

Opening a project file in Pajek

- File > Pajek Project File > Read
- Find SanJuanSur2.paj, which is associated with Chapter 9 of Exploratory Social Network Analysis by de Nooy et al.
- A project file can contain several network (.net), partition (.clu) and vector files (.vec) all in one bundle.

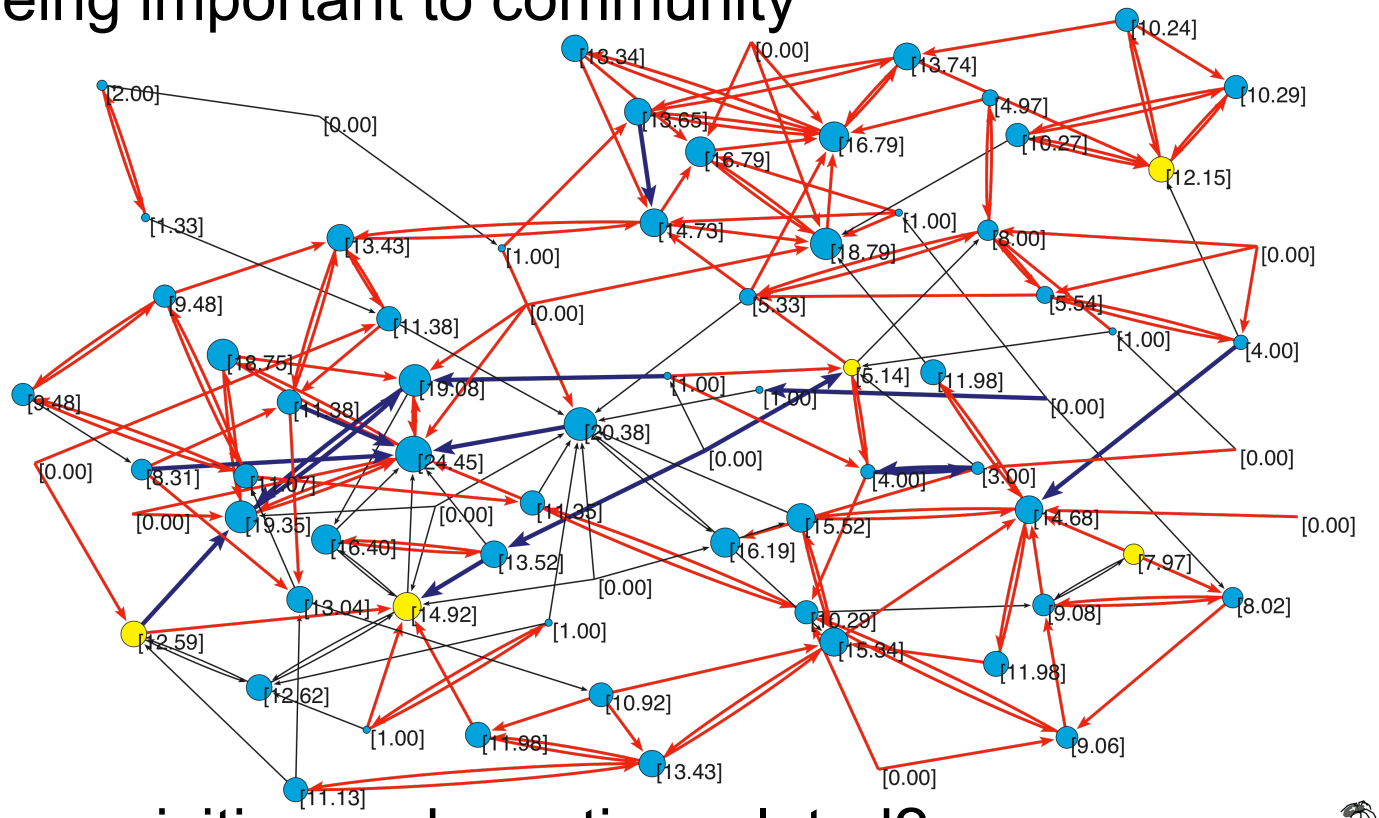
Proximity prestige in Pajek

$$C_p(n_i) = \frac{\text{fraction of all vertices that are in i's input domain}}{\text{average distance from i to vertex in input domain}}$$

- Net > Partitions > Domain > Input
  - Numerator appears under partitions as:
    - size of input domain in N…
  - Denominator appears under vectors as:
    - average distance from input domain in N9
  - Need to divide size of input domain by average distance
    - first create a vector from the partition
      - Partition > Make vector
    - use second vector drop-down menu to select av. distance
    - Vectors > divide first by second
    - this should give you a new vector with proximity prestige for each vertex

proximity prestige: example w/ haciendas in Costa Rican village

- Pajek project file to use: SanJuanSur2.paj
- edge: one family visits another
- yellow: leaders – nominated by most other families as being important to community



- are visiting and prestige related?

Pajek

using correlation to check correspondence

- Let's use Pearson's correlation coefficient
  - (Pajek can also give you Spearman's coefficient, see Ch. 9)
  - to compare visiting proximity prestige and direct prestige nominations

  - take # of leadership nominations (SanJuanSur_status.clu)
  - Partitions > Make Vector
  - Use second drop down menu to select your normalized proximity prestige
  - Vectors > Info
    - Pearson correlation coefficient for the two vectors is 0.26 (not incredibly high, would be nice to see a p-value to see if it is significant, but you would need to export into stats software to do that…)

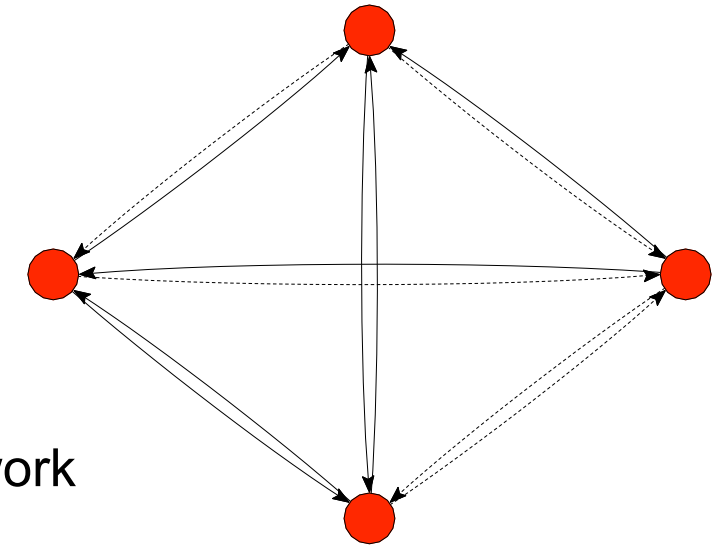
lab exercises:

- repeat this for input degree in the visitor's network
  - which is more predictive
    - the number of families that directly visit
    - the "visiting" proximity prestige?
- repeat for output degree. which is more important, visiting or being visited?
- try undirected measures: betweenness, degree
  - how do they stack up?

reminder: graph density

- Of the connections that may exist between n nodes
  - directed graph
    $e_{max} = n*(n-1)$
    each of the n nodes can connect to (n-1) other nodes
  - undirected graph
    $e_{max} = n*(n-1)/2$
    since edges are undirected, count each one only once

- What fraction are present?
  - density = e/ e_{max}
  - For example, out of 12 possible connections, this graph has 7, giving it a density of 7/12 = 0.583



- But it is more difficult for a larger network to achieve the same density

- measure not useful for comparing networks of different densities

density in Pajek

- Info> Network > General
  - returns two numbers
    - “loops allowed”: calculation if self-edges were allowed
    - “no loops allowed”: calculation if self-edges were not possible
  - usually we are dealing with a network with no self-edges, and you’ll want the second, “no loops allowed” number
  - assuming your graph has no self-loops, which of the two density calculations do you expect to give you the larger value? Why?

lab wrap up

- Prestige is centrality applied to directed edges
- We learned how to use Pajek to calculate
  - indegree
  - proximity prestige
- A way to validate centrality and prestige measures based on the social network is to correlate them with other prestige values for nodes
  - We learned how to correlate two vectors using Pajek
- The density of a network is the proportion of edges that are present out of all the edges that could be there
  - Density is easy to compute with Pajek, but difficult to interpret when comparing networks of different sizes