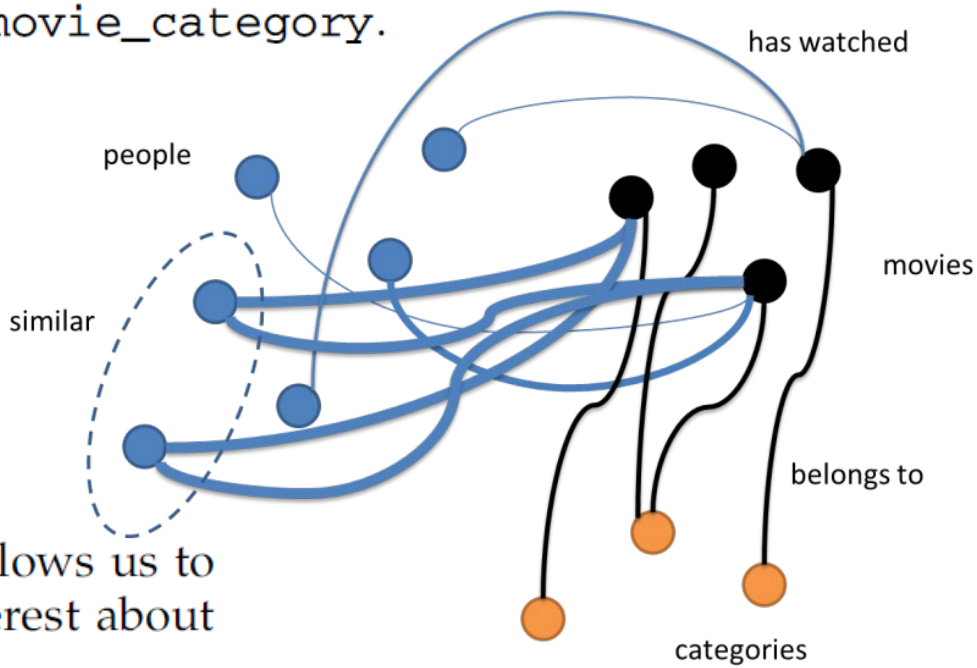
Similarities

For example, imagine a simple movie database with three sets of elements (or tables), `people`, `movie`, and `movie_category`, and two relationships `has_watched`, between `people` and `movie`, and `belongs_to`, between `movie` and `movie_category`.



Computing similarities between people allows us to cluster them into groups with similar interest about watched movies.

Computing similarities between people and movies allows us to suggest movies to watch or not to watch.

Computing similarities between people and movie categories allows us to attach a most relevant category to each person.

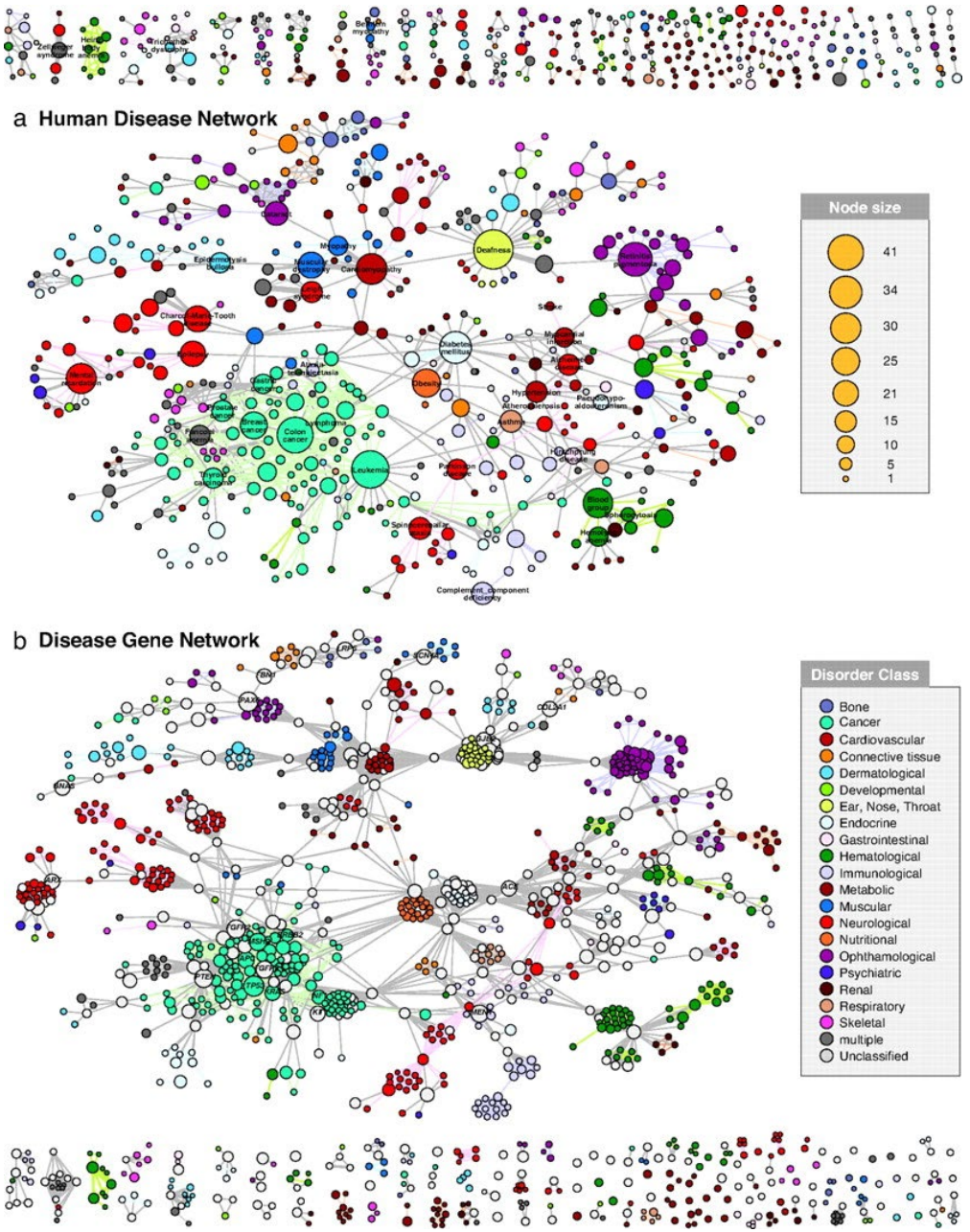
[FPRS] Random-walk based similarities

Similarities

HDN: Each node corresponds to a distinct disorder, colored based on the disorder class. The size of each node is proportional to the number of genes participating in the corresponding disorder, and the link thickness is proportional to the number of genes shared by the disorders it connects.

DGN: each node is a gene, with two genes being connected if they are implicated in the same disorder. The size of each node is proportional to the number of disorders in which the gene is implicated. Nodes are light gray if the corresponding genes are associated with more than one disorder class. Only nodes with at least one link are shown.

Goh et al. “Human Disease Network”, PNAS, 2007



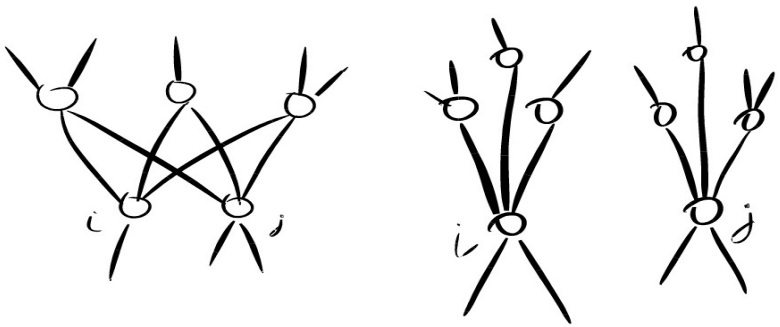
Classes of Similarities

Q: In what ways can vertices in a network be similar and how can we quantify this similarity?

- Similarity between vertices
  - Structural equivalence
  - Regular equivalence

i and *j* share many of the same network neighbors

i and *j* do not necessarily share neighbors but have neighbors who are themselves similar



Structural Regular

Structural Equivalence

- Number of common neighbors, i.e., $n_{ij} = \sum_k A_{ik} A_{kj} = ij\text{th element of } A^2$

Problem: Simple count of common neighbors for two vertices is not on its own a very good measure of similarity. If two vertices have 3 common neighbors is that a lot or a little? It's hard to tell unless we know, for instance, what the degrees of the vertices are, or how many common neighbors other pairs of vertices share.

Solution: adding some sort of normalization

Structural Equivalence

- Number of common neighbors, i.e., $n_{ij} = \sum_k A_{ik}A_{kj} = ij\text{th element of } A^2$
- Cosine similarity

$$\sigma_{ij} = \cos\theta = \frac{\sum_k A_{ik}A_{kj}}{\sqrt{\sum_k A_{ik}^2}\sqrt{\sum_k A_{kj}^2}} = \frac{n_{ij}}{\sqrt{d(i)d(j)}} \in [0,1]$$

degrees of *i* and *j*

Structural Equivalence

- Number of common neighbors, i.e., $n_{ij} = \sum_k A_{ik}A_{kj} = ij\text{th element of } A^2$
- Cosine similarity

$$\sigma_{ij} = \cos\theta = \frac{\sum_k A_{ik}A_{kj}}{\sqrt{\sum_k A_{ik}^2}\sqrt{\sum_k A_{kj}^2}} = \frac{n_{ij}}{\sqrt{d(i)d(j)}} \in [0,1]$$

- Pearson coefficients

$$\sum_k A_{ik}A_{jk} - \frac{d(i)d(j)}{n}$$

≈expected number of common neighbors

Given ***i*** and ***j,*** how many common neighbors should we expect them to have in random model? Imagine that vertex ***i*** chooses ***d(i)*** neighbors uniformly at random, and vertex ***j*** **similarly chooses** ***d(j)*** neighbors at random. For the first neighbor that *j* chooses there is a probability of ***d(i)/n*** that it will choose one of the ***d(i)***, and similarly for each succeeding choice. (We neglect the possibility of choosing the same neighbor twice, since it is small for a large network.) Then in total the expected number of common neighbors between the two vertices will be ***d(i)d(j)/n***.

Structural Equivalence

- Number of common neighbors, i.e., $n_{ij} = \sum_k A_{ik}A_{kj} = ij\text{th element of } A^2$
- Cosine similarity

$$\sigma_{ij} = \cos\theta = \frac{\sum_k A_{ik}A_{kj}}{\sqrt{\sum_k A_{ik}^2}\sqrt{\sum_k A_{kj}^2}} = \frac{n_{ij}}{\sqrt{d(i)d(j)}} \in [0,1]$$

$\langle A_i \rangle = \frac{1}{n}\sum_k A_{ik}$

- Pearson coefficients

$$\sum_k A_{ik}A_{jk} - \frac{d(i)d(j)}{n} = \sum_k A_{ik}A_{jk} - \frac{1}{n}\sum_k A_{ik}\sum_l A_{jl}$$

$$= \sum_k A_{ik}A_{jk} - n\langle A_i\rangle\langle A_j\rangle = \sum_k [A_{ik}A_{jk} - \langle A_i\rangle\langle A_j\rangle]$$

≈expected number of common neighbors

$$= \sum_k (A_{ik} - \langle A_i\rangle)(A_{jk} - \langle A_j\rangle) = n \cdot \mathrm{cov}(A_i, A_j)$$

$$r_{ij} = \frac{\mathrm{cov}(A_i, A_j)}{\sigma_i\sigma_j} = \frac{\sum_k (A_{ik} - \langle A_i\rangle)(A_{jk} - \langle A_j\rangle)}{\sqrt{\sum_k (A_{ik} - \langle A_i\rangle)^2}\sqrt{\sum_k (A_{jk} - \langle A_j\rangle)^2}}, \quad -1 \le r_{ij} \le 1$$

- Euclidean distance (number of neighbors that differ) $d_{ij} = \sum_k (A_{ik} - A_{jk})^2$

Regular Equivalence

The vertices have neighbors that are themselves similar

$$\sigma = \alpha A \sigma A \text{ or } \sigma_{ij} = \alpha \sum_{kl} A_{ik} A_{jl} \sigma_{kl}$$

k-l similarity

similarity matrix

Problem: σ_{ii} is not necessarily high
Solution: extra diagonal term

$$\sigma = \alpha A \sigma A + I \text{ or } \sigma_{ij} = \alpha \sum_{kl} A_{ik} A_{jl} \sigma_{kl} + \zeta_{ij}$$

Still problem: in iterative calculation (init 0) we count only even paths

New formulation: i and j are similar if i has a neighbor k that is similar to j

$$\sigma = \alpha A \sigma + I \text{ or } \sigma_{ij} = \alpha \sum_{k} A_{ik} \sigma_{kj} + \zeta_{ij}$$

Convergence: $\sigma = \sum_{m=0}^{\infty} (\alpha A)^m = (I - \alpha A)^{-1}$

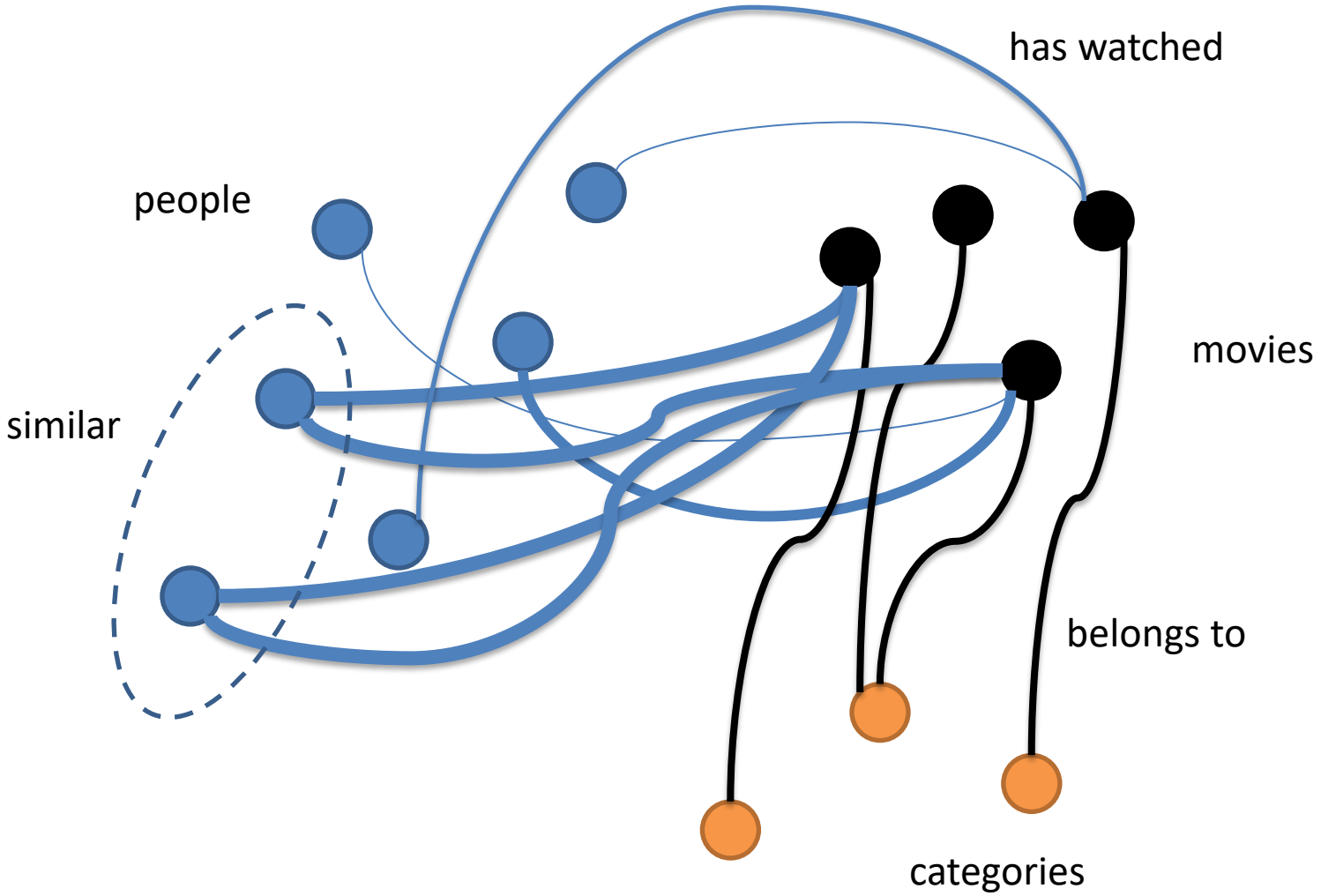
Another problem: too high similarity for high-degree nodes which is not necessarily true

Solution: divide by $d(i)$

$$\sigma = \alpha D^{-1} A\sigma + I \text{ or } \sigma_{ij} = \frac{\alpha}{d(i)} \sum_k A_{ik}\sigma_{kj} + \zeta_{ij}$$

PDF: Algebraic Distance

Random walk based similarities



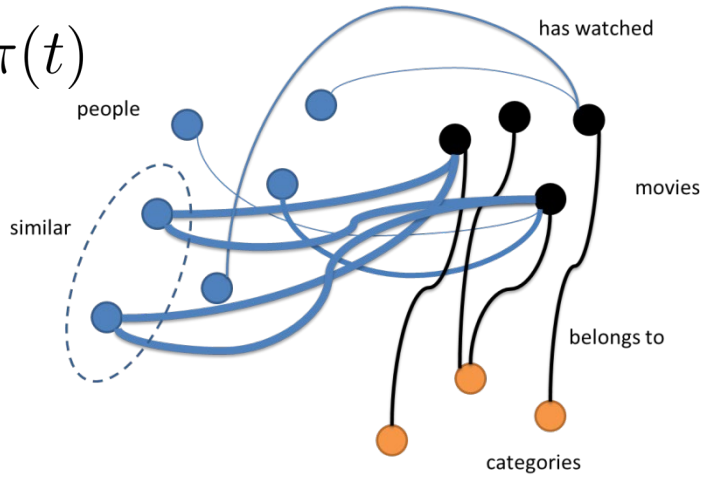
Random walk based similarities

The Markov chain (t - step, $s(t)$ - state at t) describing the sequence of nodes visited by a random walker is called a random walk. The random walk is defined with the following single-step transition probabilities of jumping from any state or node $i = s(t)$ to an adjacent node

$$j = s(t+1) : Pr(s(t+1) = j | s(t) = i) = a_{ij}/a_{ii} = p_{ij},$$

where $a_{ii} = \sum_{j=1}^{n} a_{ij}$. The probability of being in state i at time t is $\pi_i(t) = Pr(s(t) = i)$ and P is the transition matrix with entries p_{ij}. The evolution of Markov chain is given by

$$\pi(t+1) = P^T \pi(t)$$

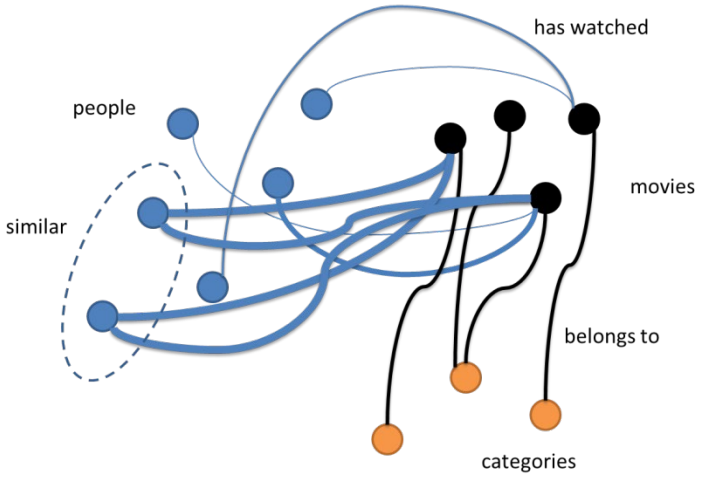


Random walk based similarities

The average first-passage time $m(k|i)$ is the average number of steps that a random walker, starting in (random) state $i \neq k$, will take to enter state k for the first time, i.e.,

$$m(k|i) = E[T_{ik}|s(0) = i], \text{ where } T_{ik} = \min(t \geq 0|s(t) = k,\ s(0) = i).$$

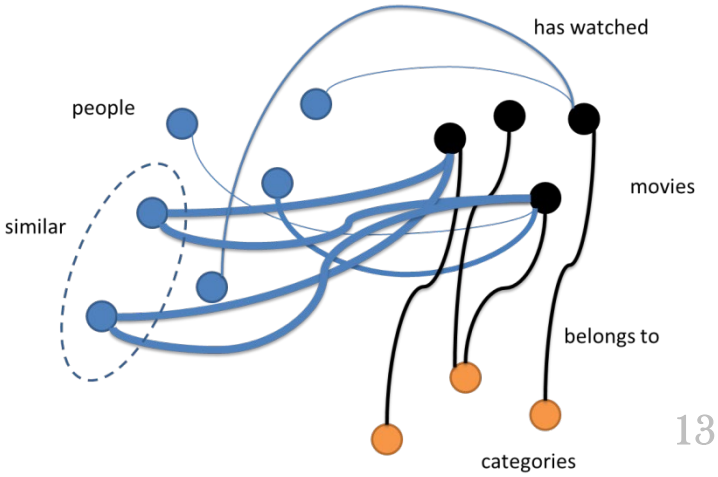
$$\begin{cases} m(k|k) = 0 \\ m(k|i) = 1 + \sum_{j=1}^{n} p_{ij}\, m(k|j), \quad \text{for } i \neq k, \end{cases}$$



Random walk based similarities

The average first-passage cost $o(k|i)$ is the average cost incurred by the random walker starting from state i to reach state k for the first time. The cost of each transition is given by $c(j|i)$.

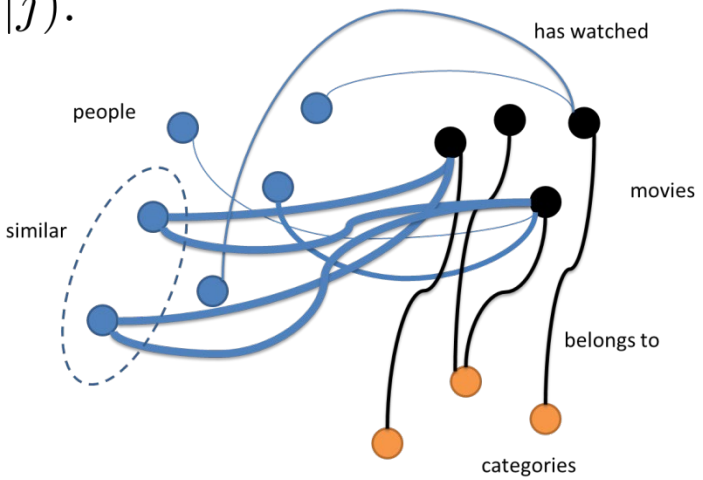
$$\begin{cases} o(k|k) = 0 \\ o(k|i) = \sum_{j=1}^{n} p_{ij}\, c(j|i) + \sum_{j=1}^{n} p_{ij}\, o(k|j), \quad \text{for } i \neq k. \end{cases}$$



Random walk based similarities

The average commute time $n(i, j)$ is the average number of steps that a random walker, starting in state $i \neq j$, will take to enter state j for the first time and go back to i, i.e.,

$$n(i, j) = m(j|i) + m(i|j).$$



Paper review 4: "Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation"
Submit by 9/26/2019

Homophily and Assortative Mixing

- the tendency of individuals to associate and bond with similar others.

Examples: social networks, citation networks, web pages languages, animals

Disassortative Mixing – opposite to assortative

Example: sexual contact networks

c_i - type of vertex i, $\delta(i,j)$ - Kronecker delta

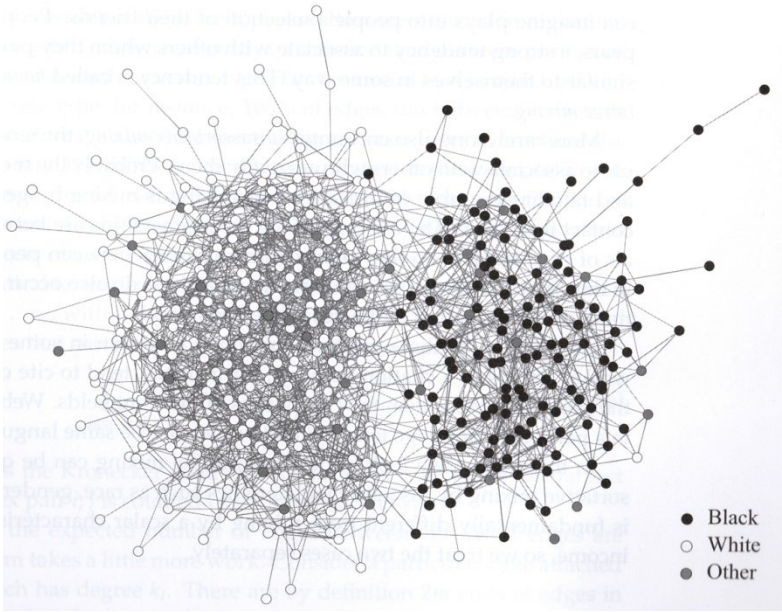
$$q = \frac{1}{2}\sum_{ij} A_{ij}\delta(c_i, c_j) - \frac{1}{2}\sum_{ij} \frac{d(i)d(j)}{2m}\delta(c_i, c_j)$$

total number of edges between similar vertices

expected number of edges between similar vertices in random model

$Q = q/m$ is called *modularity*.

Newman "Networks: An Introduction"



Friendship network at a US high school. 470 students, 14-18 yo
Q = 0.305

Modularity is a measure of the extent to which like is connected to like in a network.

Homophily and Assortative Mixing

c_i - type of vertex i, $\delta(i,j)$ - Kronecker delta

$$q = \frac{1}{2}\sum_{ij} A_{ij}\delta(c_i,c_j) - \frac{1}{2}\sum_{ij}\frac{d(i)d(j)}{2m}\delta(c_i,c_j)$$

total number of edges between similar vertices

expected number of edges between similar vertices in random model

or assortativity coefficient

$Q = q/m$ is called *modularity*.

$Q_{\max} = \frac{1}{2m}(2m - \sum_{ij}\frac{d(i)d(j)}{2m}\delta(c_i,c_j))$, normalized modularity is $Q/Q_{\max}$

LofEdges representation: $e_{rs} = \frac{1}{2m}\sum_{ij}\delta(c_i,r)\delta(c_j,s)$, $a_r = \frac{1}{2m}\sum_i d(i)\delta(c_i,r)$

$\Longrightarrow Q = \sum_r (e_{rr} - a_r^2)$.

LofEdges: nodes may have types but no info about degrees

fraction of edges between classes *r* and *s*

fraction of ends of edges attached to vertices of class *r*

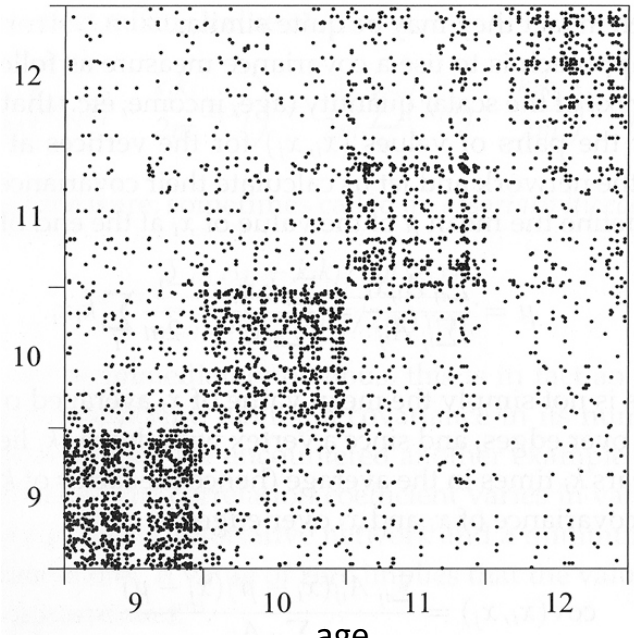
Maximization of the modularity is a well-known clustering approach

Assortative Mixing and Scalar Characteristics

values come in a particular order

In practice, the number of classes will be limited.
Reasons: complexity, bins, etc.
Example: school friends, age × age

Newman "Networks: An Introduction"



r = 0.616

Problem: vertices falling in different bins are different when in fact they may be similar (10.9yo≈11yo)

If x_i and x_j are scalars (instead of c_i and c_j then define

$$cov(x_i, x_j) = \frac{\sum_{ij} A_{ij}(x_i - \mu)(x_j - \mu)}{\sum_{ij} A_{ij}},$$

where mean $\mu = \frac{\sum_{ij} A_{ij} x_i}{\sum_{ij} A_{ij}} = 1/2m \cdot \sum_i d(i) x_i$

$$\Longrightarrow \ \ldots \ \Longrightarrow cov(x_i, x_j) = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{d(i)d(j)}{2m}) x_i x_j.$$

Assortative Coefficient

covariance

$$r = \frac{\sum_{ij}(A_{ij} - d(i)d(j)/2m)x_i x_j}{\sum_{ij}(d(i)\delta_{ij} - d(i)d(j)/2m)x_i x_j}$$

variance

1 – perfectly assortative network; **-1** - perfectly disassortative network

Example: Assortative Mixing by Degree

A special case of assortative mixing according to a scalar quantity, is that of mixing by degree. In a network that shows assortative mixing by degree the high-degree vertices will be preferentially connected to other high-degree vertices, and the low to low.

$$x_i = d(i)$$

$$cov(d(i), d(j)) = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{d(i)d(j)}{2m} \right) d(i)d(j)$$

$$r = \frac{\sum_{ij}(A_{ij} - d(i)d(j)/2m)d(i)d(j)}{\sum_{ij}(d(i)\delta_{ij} - d(i)d(j)/2m)d(i)d(j)}$$

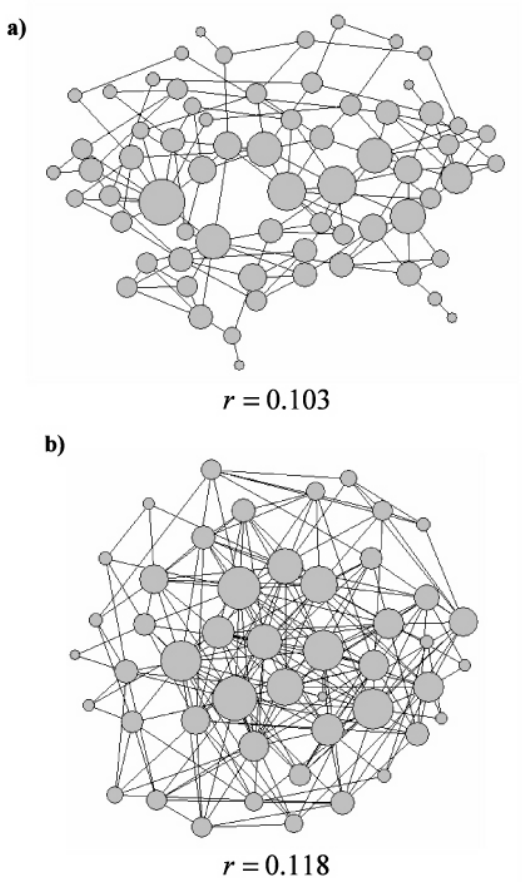
	network	n	r
real-world networks	physics coauthorship[a]	52 909	0.363
	biology coauthorship[a]	1 520 251	0.127
	mathematics coauthorship[b]	253 339	0.120
	film actor collaborations[c]	449 913	0.208
	company directors[d]	7 673	0.276
	Internet[e]	10 697	−0.189
	World-Wide Web[f]	269 504	−0.065
	protein interactions[g]	2 115	−0.156
	neural network[h]	307	−0.163
	food web[i]	92	−0.276
models	random graph[u]		0
	Callaway *et al.*[v]		$\delta/(1+2\delta)$
	Barabási and Albert[w]		0

TABLE I: Size n and assortativity coefficient r for a number of different networks: collaboration networks of (a) scientists in physics and biology [16], (b) mathematicians [17], (c) film actors [4], and (d) businesspeople [18]; (e) connections between autonomous systems on the Internet [19]; (f) undirected hyperlinks between Web pages in a single domain [6]; (g) protein-protein interaction network in yeast [20]; (h) undirected (and unweighted) synaptic connections in the neural network of the nematode *C. Elegans* [4]; (i) undirected trophic relations in the food web of Little Rock Lake, Wisconsin [21]. The last three lines give analytic results for model networks in the limit of large network size: (u) the random graph of Erdős and Rényi [22]; (v) the grown graph model of Callaway *et al.* [15]; (w) the preferential attachment model of Barabási and Albert [6].

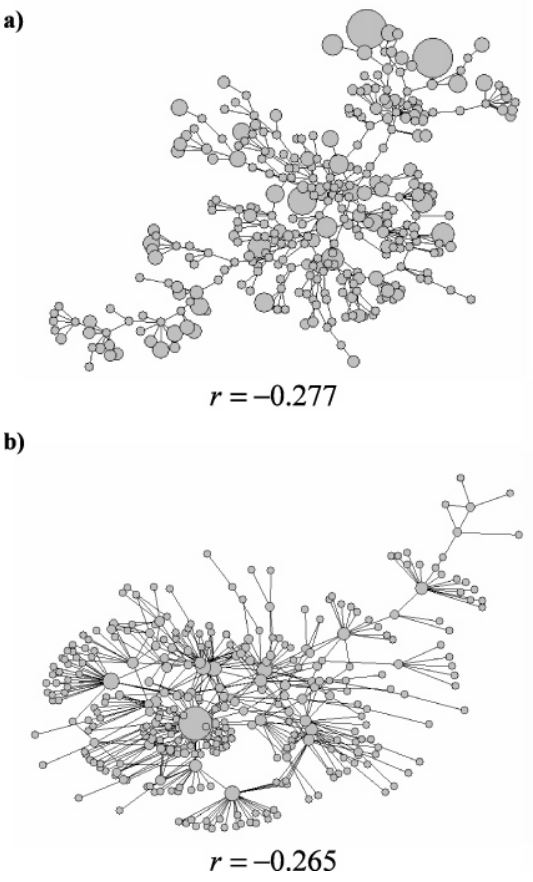
Newman "Assortative mixing in networks"

Example: Assortative Mixing by Degree

Estrada et al. "Clumpiness" mixing in complex networks



The network illustrated in Figure (a) corresponds to the inmates in a prison and that in Figure (b) to the food web. Both networks are almost of the same size, and both display uniform degree distributions and have almost identical assortativity coefficient, r = 0.103 and 0.118, respectively. However, while in the prison network the high-degree nodes are spread across the network, they are clumped together in the food web. This difference can have dramatic implications for the structure and functioning of these two systems.

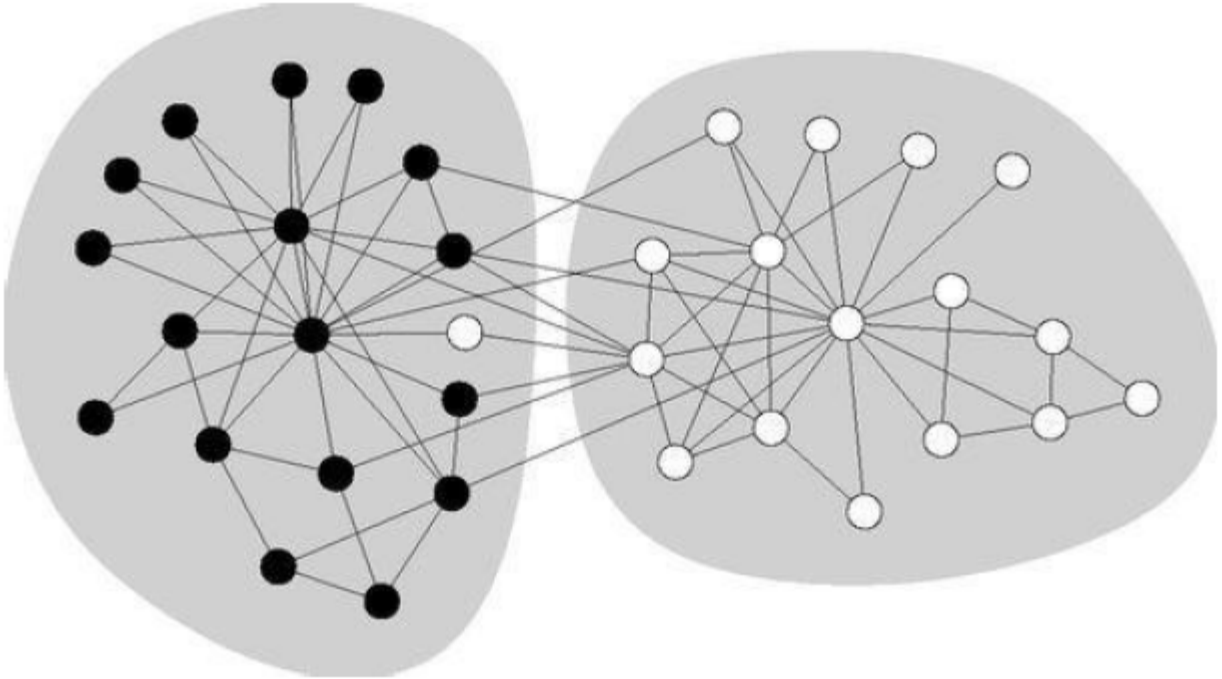


Disassortative networks. We can also find that the high-degree nodes can be separated by only two links with a low-degree node acting as a bridge or by very long paths. This situation is illustrated in sexual network in Colorado Springs (a) and the transcription interaction network of *E. coli* (b), which have almost equal negative assortative coefficients. In the former case the high-degree nodes are separated by very long chains while in the latter case most of the high-degree nodes are clumped together separated by only two or three links.

Simple Modularity Maximization

Iterative Algorithm (inspired by Kernighan-Lin algorithm for partitioning problem)
1. Choose initial division of a network into (equally sized) groups
2. Main sweep: repeatedly move the vertices that most increase or least decrease Q
3. Return to step 2 until Q no longer improves

Complexity of Step 2: *O*(mn)



Newman “Networks: An Introduction”

Spectral Modularity Maximization

$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{d(i)d(j)}{2m}\right)\delta(c_i, c_j) = \frac{1}{2m}\sum_{ij} B_{ij}\delta(c_i, c_j)$
Note that B_{ij} has the property

$$\sum_j B_{ij} = \sum_j A_{ij} - \frac{d(i)}{2m}\sum_j d(j) = 0$$

Denote by s_i the indicator variable +1/-1 for cluster number, i.e., $\delta(c_i, c_j) = \frac{s_i s_j + 1}{2}$

modularity matrix

$$Q = \frac{1}{4m}\sum_{ij} B_{ij}s_i s_j = \frac{1}{4m}s^T B s$$

Method: relax integer constraint for s with reals and $s^T s = n$
Solve maximization problem by using Lagrange multiplier

eigenproblem

$$\frac{\partial}{\partial s_i}\left(\sum_{jk} B_{jk}s_j s_k + \beta(n - \sum_j s_j^2)\right) = 0 \Longrightarrow \sum_j B_{ij}s_j = \beta s_i \text{ or } Bs = \beta s$$

Note: In practice we cannot assign *s* with eigenvector corresponding to the largest eval (*s* is +1/-1 vector)
We choose s to be close to u_1 by maximizing $\sum_i s_i(u_1)_i$, i.e.,
$s_i = +1\ (-1)$ if $(u_1)_i\ >\ (<)\ 0$

Homework

Paper review 5 + computational problem (due 10/3/2019)

1. (50%) Paper review: Newman “Assortative mixing in networks”
2. (50%) Compute modularity

7.8 In a survey of couples in the US city of San Francisco, Catania *et al.* [65] recorded, among other things, the ethnicity of their interviewees and calculated the fraction of couples whose members were from each possible pairing of ethnic groups. The fractions were as follows:

		Women				
		Black	Hispanic	White	Other	Total
Men	Black	0.258	0.016	0.035	0.013	0.323
	Hispanic	0.012	0.157	0.058	0.019	0.247
	White	0.013	0.023	0.306	0.035	0.377
	Other	0.005	0.007	0.024	0.016	0.053
	Total	0.289	0.204	0.423	0.084	

Assuming the couples interviewed to be a representative sample of the edges in the undirected network of relationships for the community studied, and treating the vertices as being of four types—black, Hispanic, white, and other—calculate the numbers e_{rr} and a_r that appear in Eq. (7.76) for each type. Hence calculate the modularity of the network with respect to ethnicity.