# Similarities

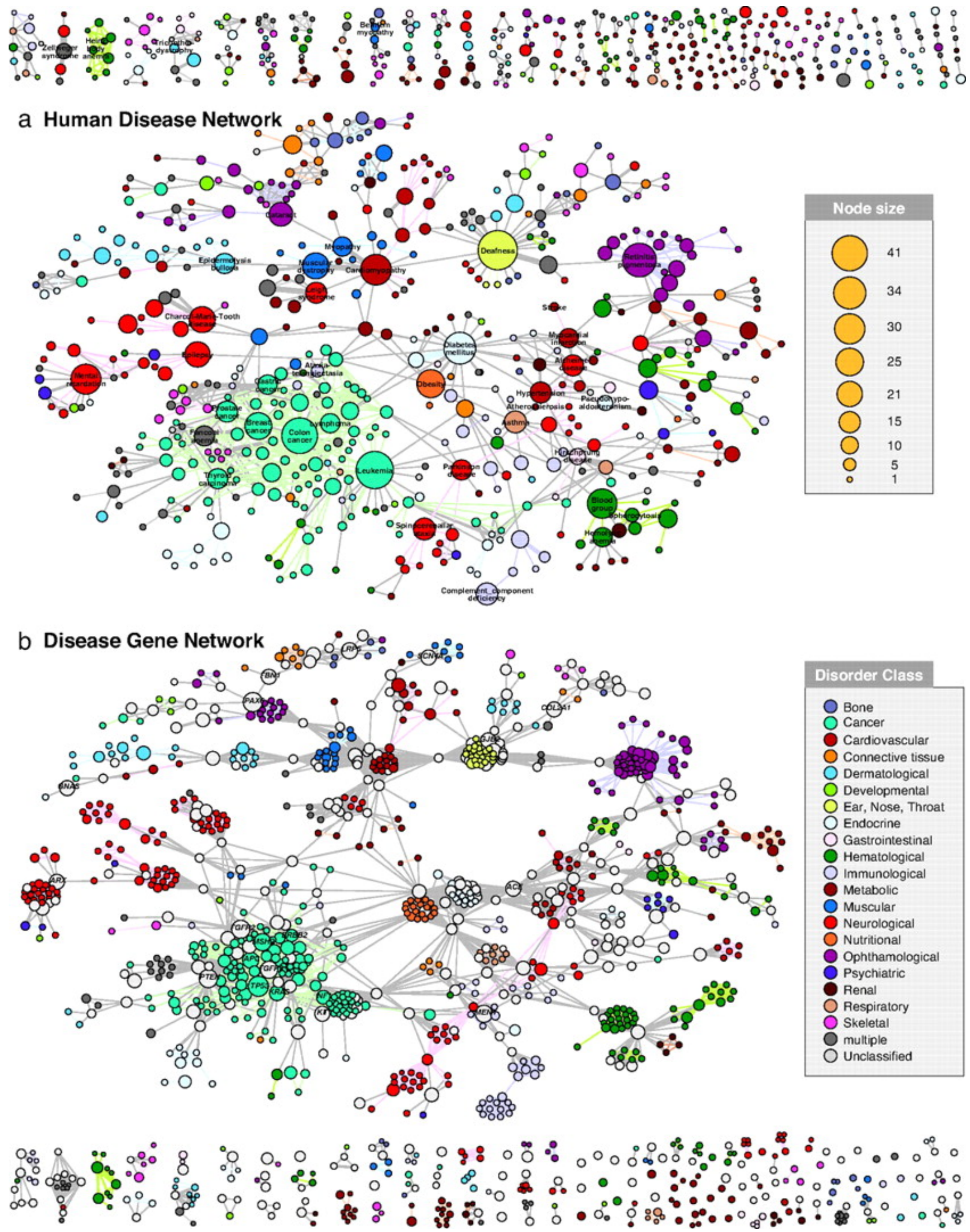
## Why do we need to compute them?

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

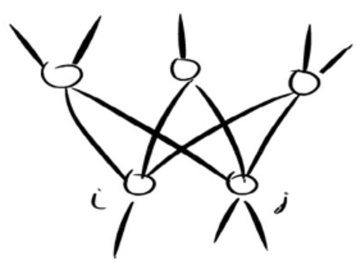



Goh et al. “Human Disease Network”, PNAS, 2007

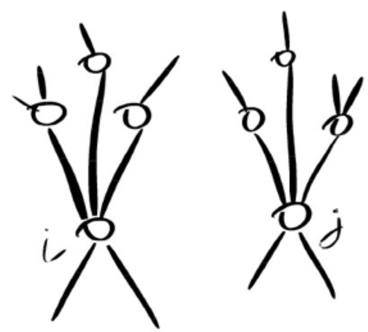
# Classes of Similarities

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

- Similarity between vertices
  - Structural equivalence — *I* and *j* share many of the same network neighbors
  - Regular equivalence — *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., $n_{ij} = \sum_k A_{ik}A_{kj} = ij$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

$$\begin{aligned}\sum_k A_{ik}A_{jk} - \frac{d(i)d(j)}{n} &= \sum_k A_{ik}A_{jk} - \frac{1}{n}\sum_k A_{ik}\sum_l A_{jl} \\ &= \sum_k A_{ik}A_{jk} - n\langle A_i\rangle\langle A_j\rangle = \sum_k [A_{ik}A_{jk} - \langle A_i\rangle\langle A_j\rangle] \\ &= \sum_k (A_{ik} - \langle A_i\rangle)(A_{jk} - \langle A_j\rangle) = n \cdot \mathrm{cov}(A_i, A_j)\end{aligned}$$

≈expected number of common neighbors

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

- 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

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

similarity

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

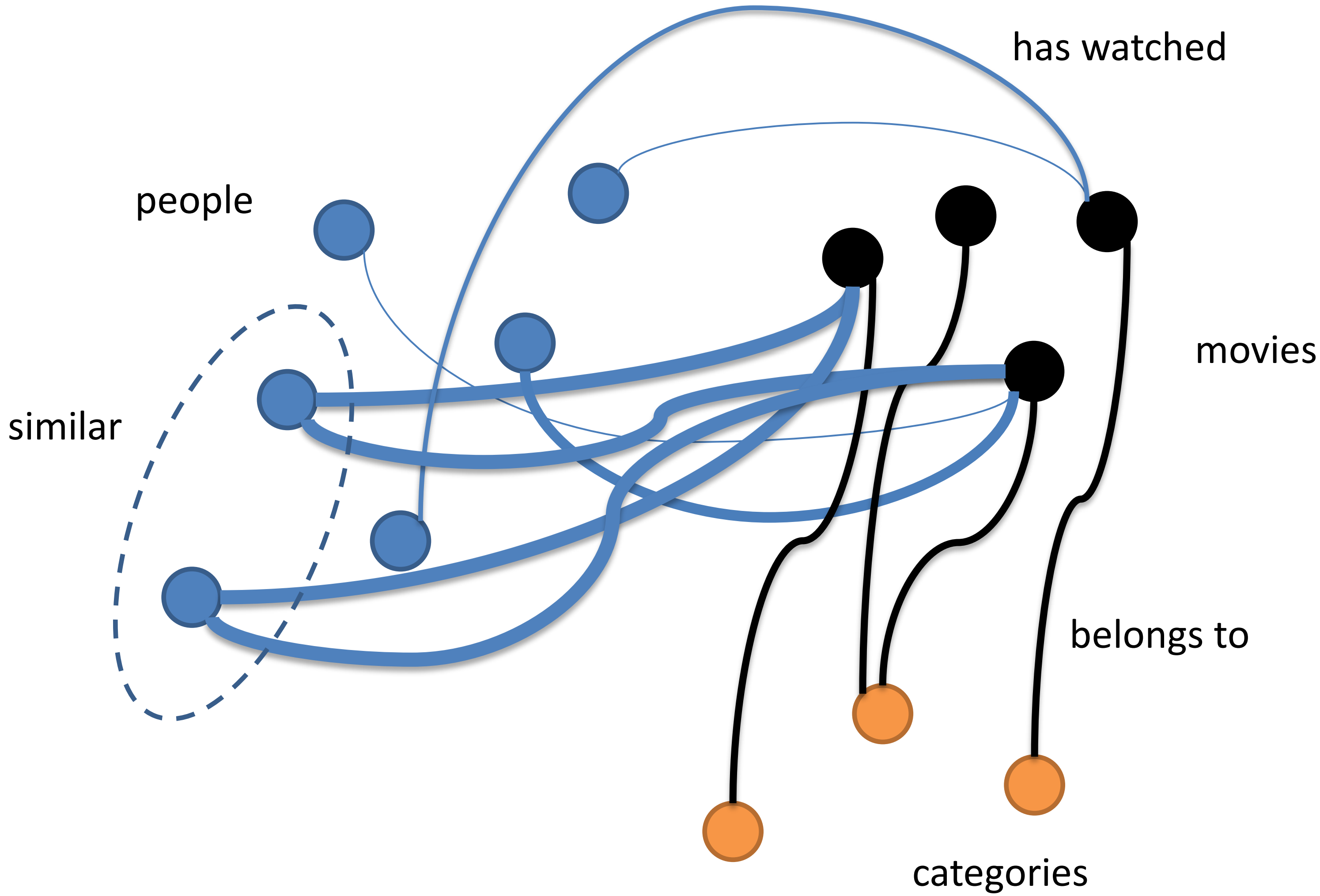
Another problem: too high similarity for high-degree nodes which is not necessarily true

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




[FPRS] Random-walk based similarities

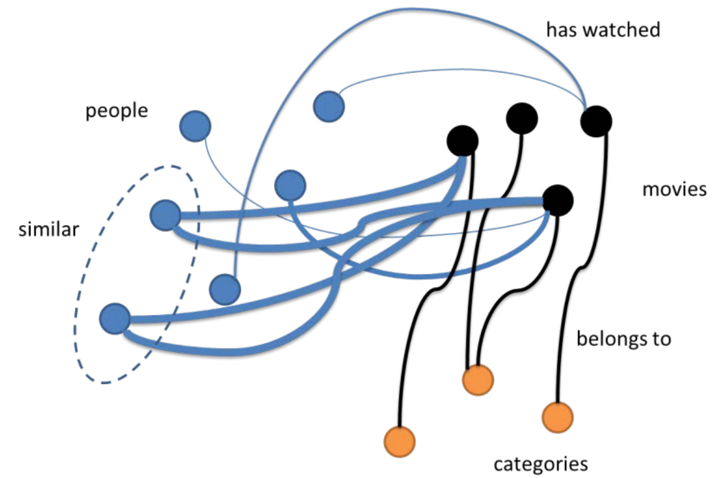
# Random walk based similarities

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

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




[FPRS] Random-walk based similarities

# Random walk based similarities

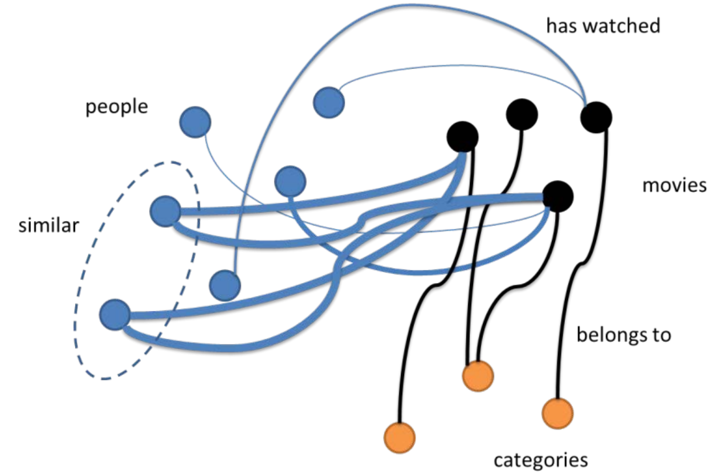
The average first-passage time $m(k|i)$ is the average number of steps that a random walker, starting in (random) state $i \neq k$, will take to enter state $k$ for the first time, i.e.,

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

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 $c(j|i)$.

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

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

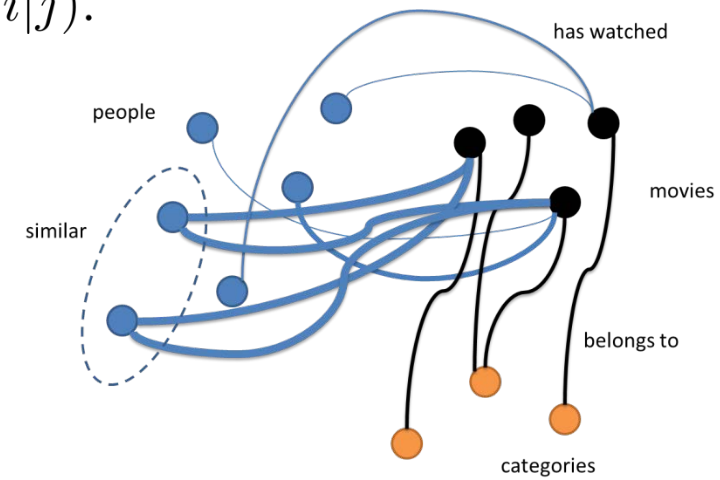



[FPRS] Random-walk based similarities

# 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.,

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




Homework: review of "Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation"
Submit by 2/11/2014

[FPRS] Random-walk based similarities