Social Networks Characteristics & Link Prediction

Yannis Kotidis

Department of Informatics

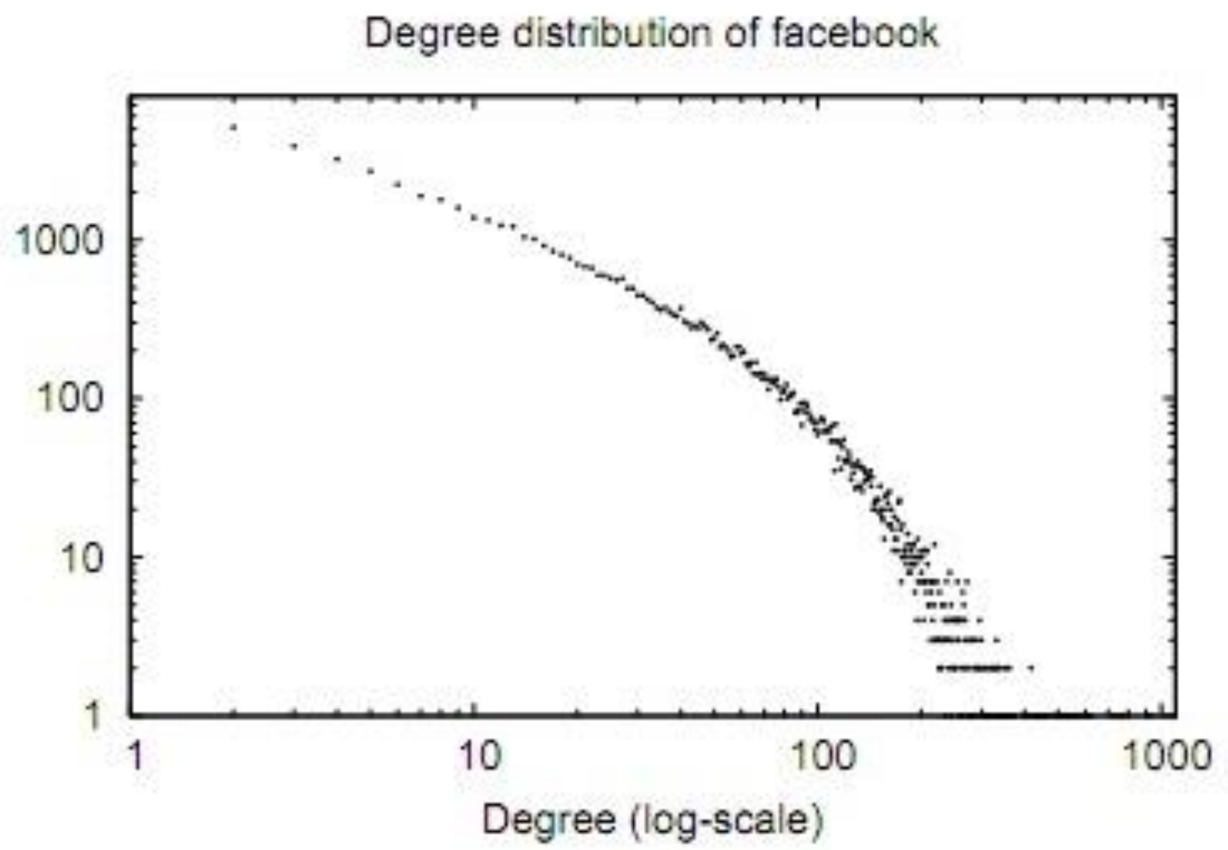
Athens University of Economics and Business

The small world effect

- The average distance in the topology is very small compared to the size of the network.
  - Facebook avg distance is ~5

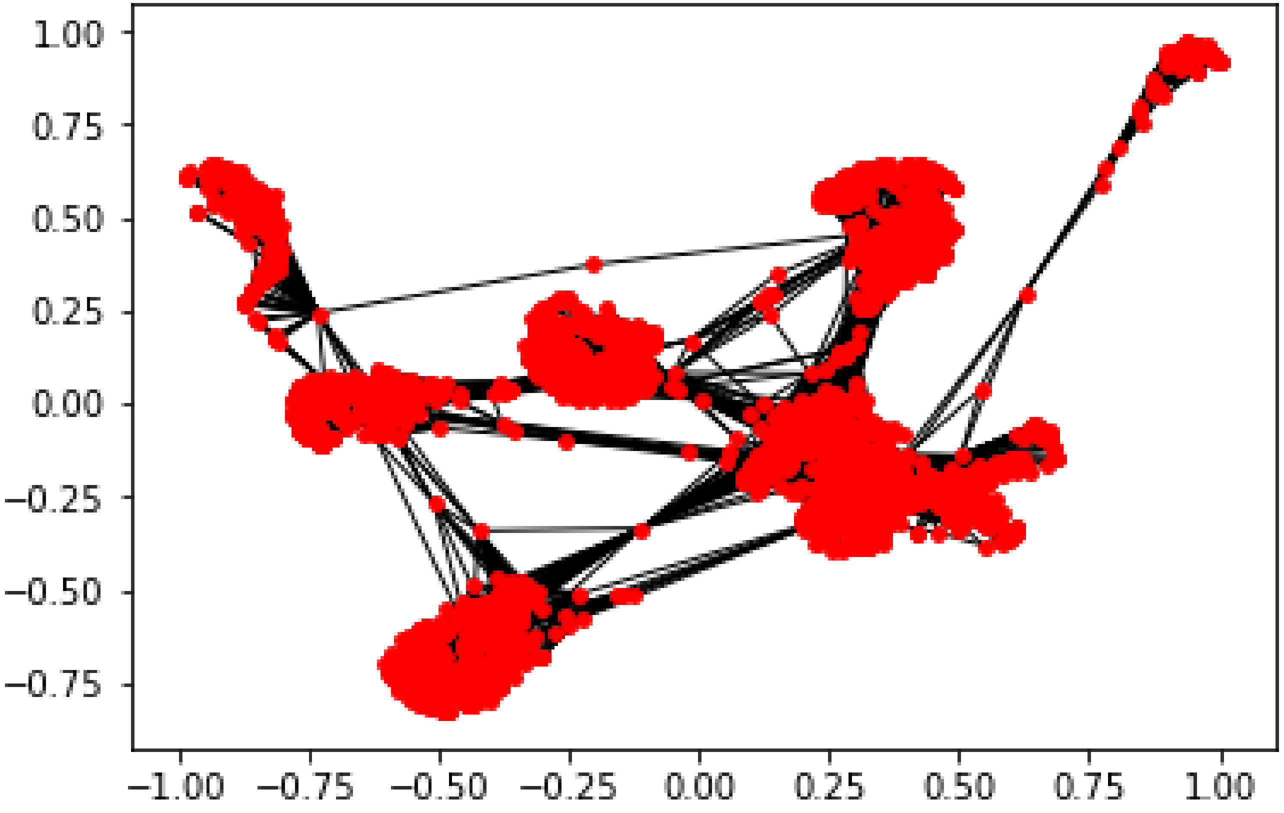
Power law degree distribution

- Only a few nodes have lots of links. Most nodes' links are very small in the network
- See: M. Faloutsos, P. Faloutsos, C. Faloutsos: *On power-law relationships of the Internet topology*



Clustering effect

- There are many small groups in a social network where each member of the group knows each other.
  - Results in many fully connected subgraphs.

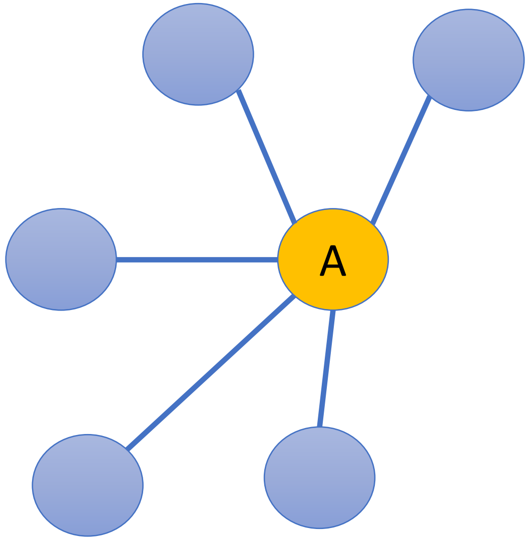


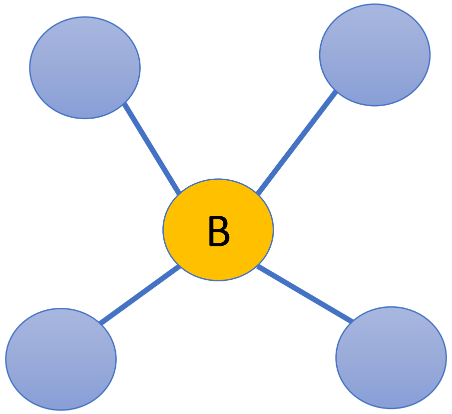
Link prediction

- Predict the **likelihood** of a future association between two nodes
- This is typically achieved by utilizing some notion of **similarity** between the two nodes

Preferential Attachment

- **Intuition:** users with many friends tend to create more connections in the future
  - Lets N(u) denote the set of friends of node u
  - Then, use |N(u)|*|N(v)| as a measure for scoring the likelihood of a link between node u and node v

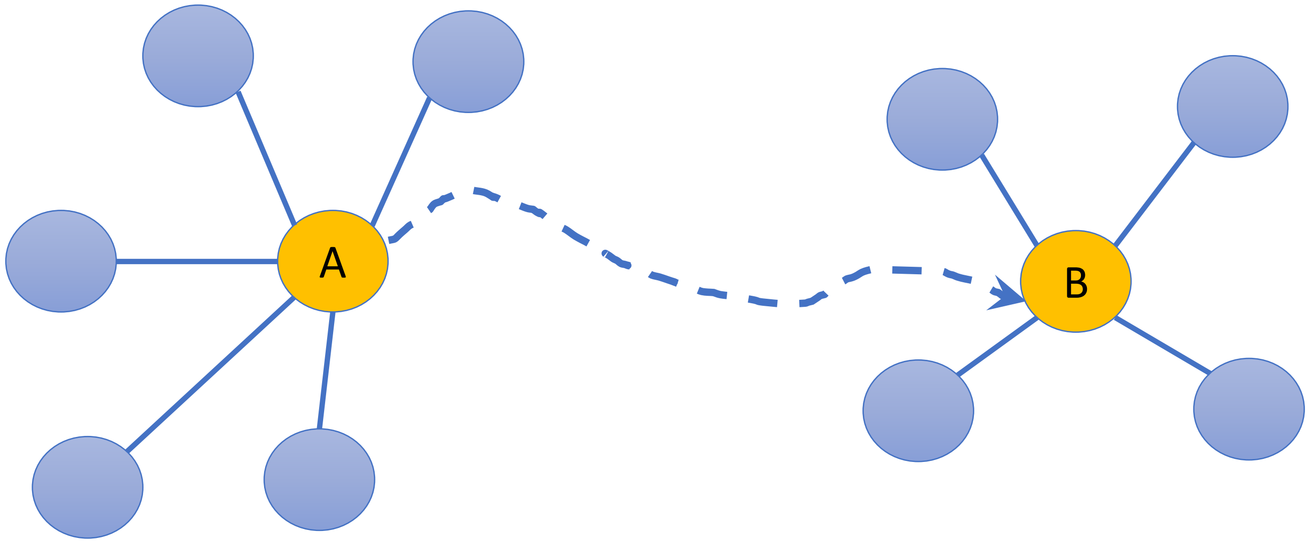




Caveat?

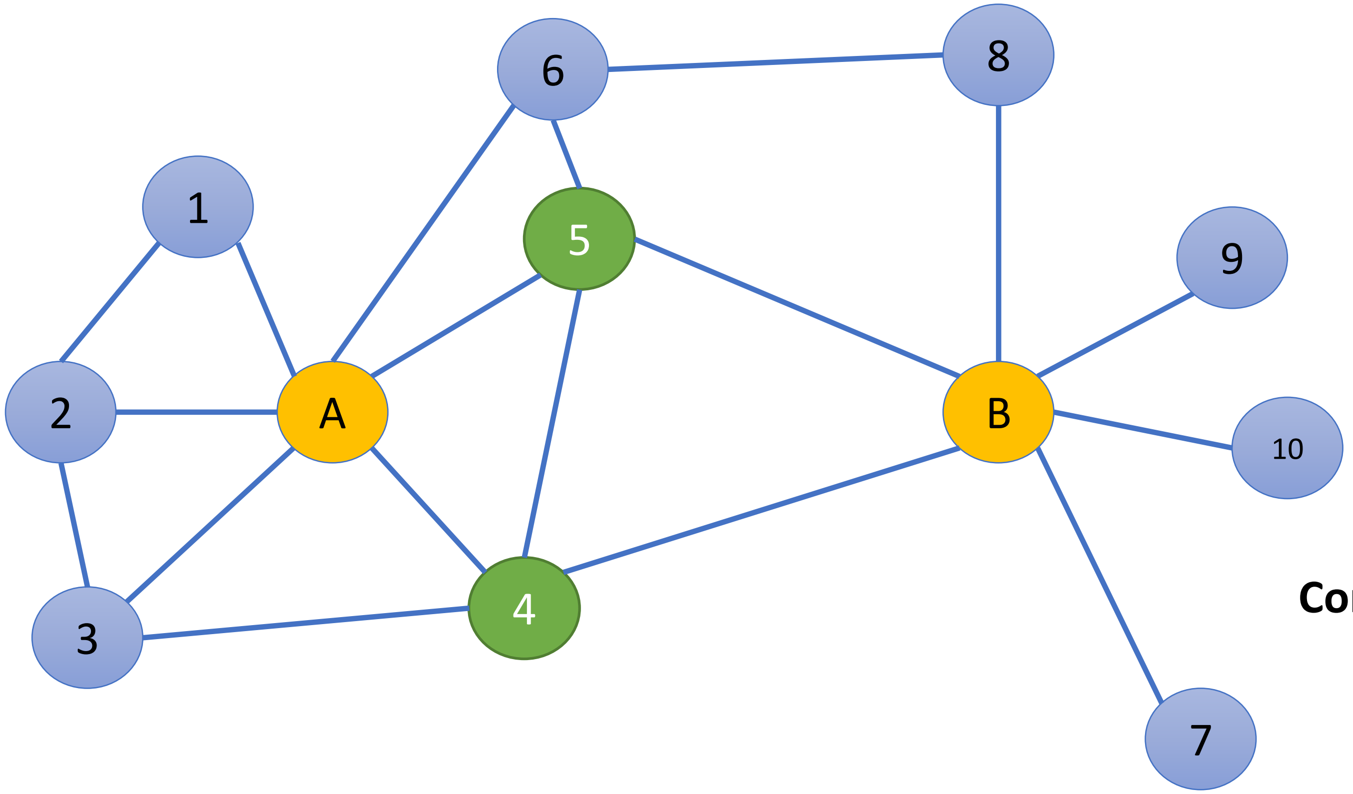
Graph Distance

- Define similarity as the negative length of the shortest path between nodes A and B
  - Recall that due to the small-world effect this distance is typically small
  - Thus, it may be hard to differentiate pairs using shortest-paths



Count Common Neighbours

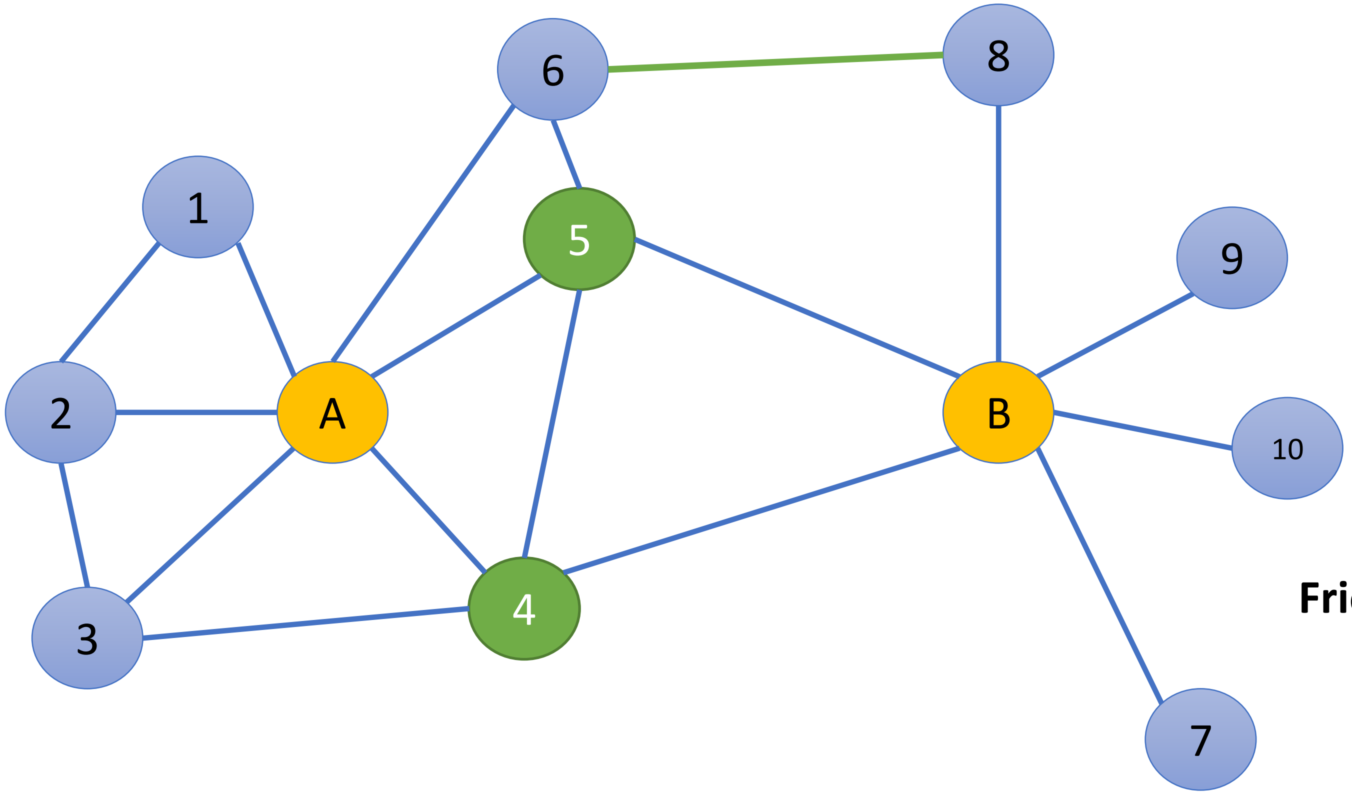
- **Intuition:** two strangers who have a common acquaintance may be introduced by that friend.



Common-Neighbours(A,B)=2

Friends-measure

- **Count #common-friends + #connections_between_neighbors**

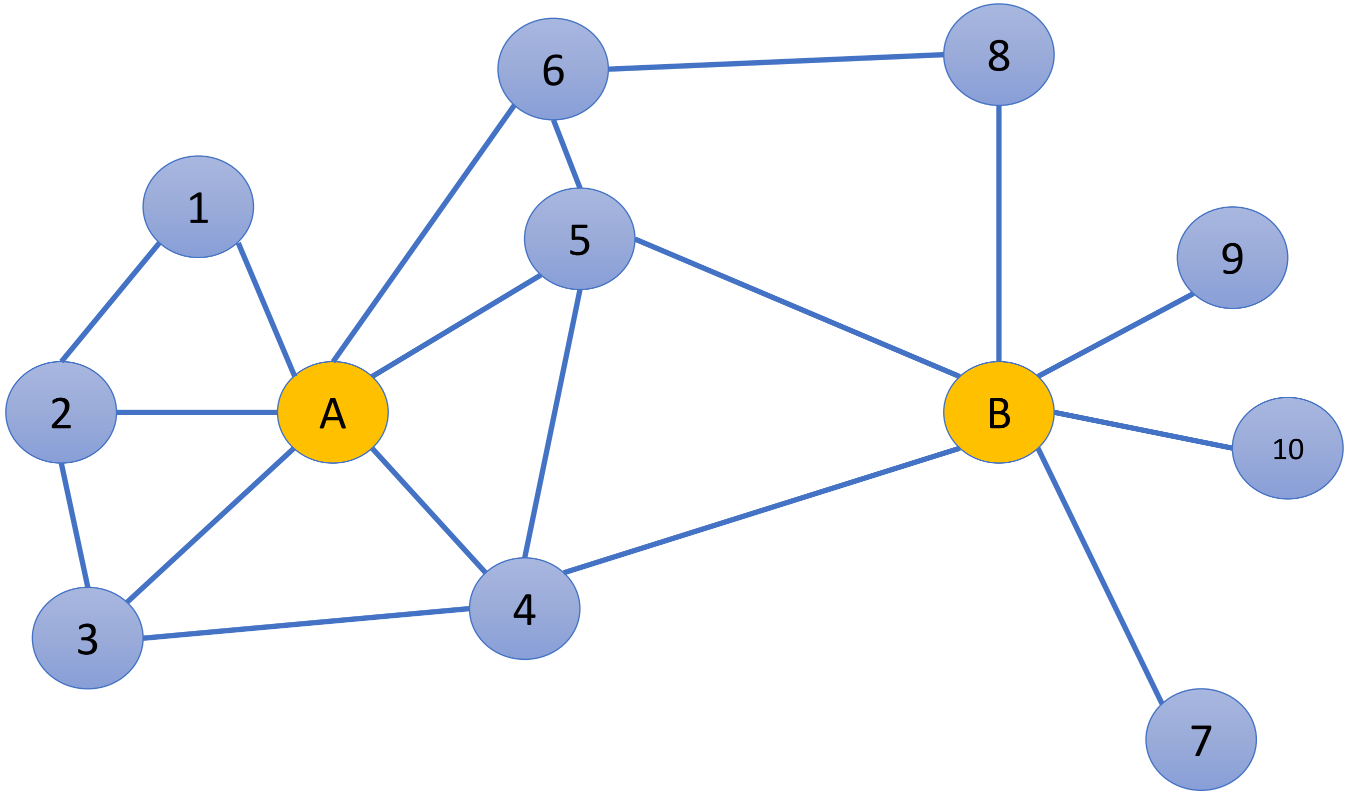


Friends-measure(A,B)=3

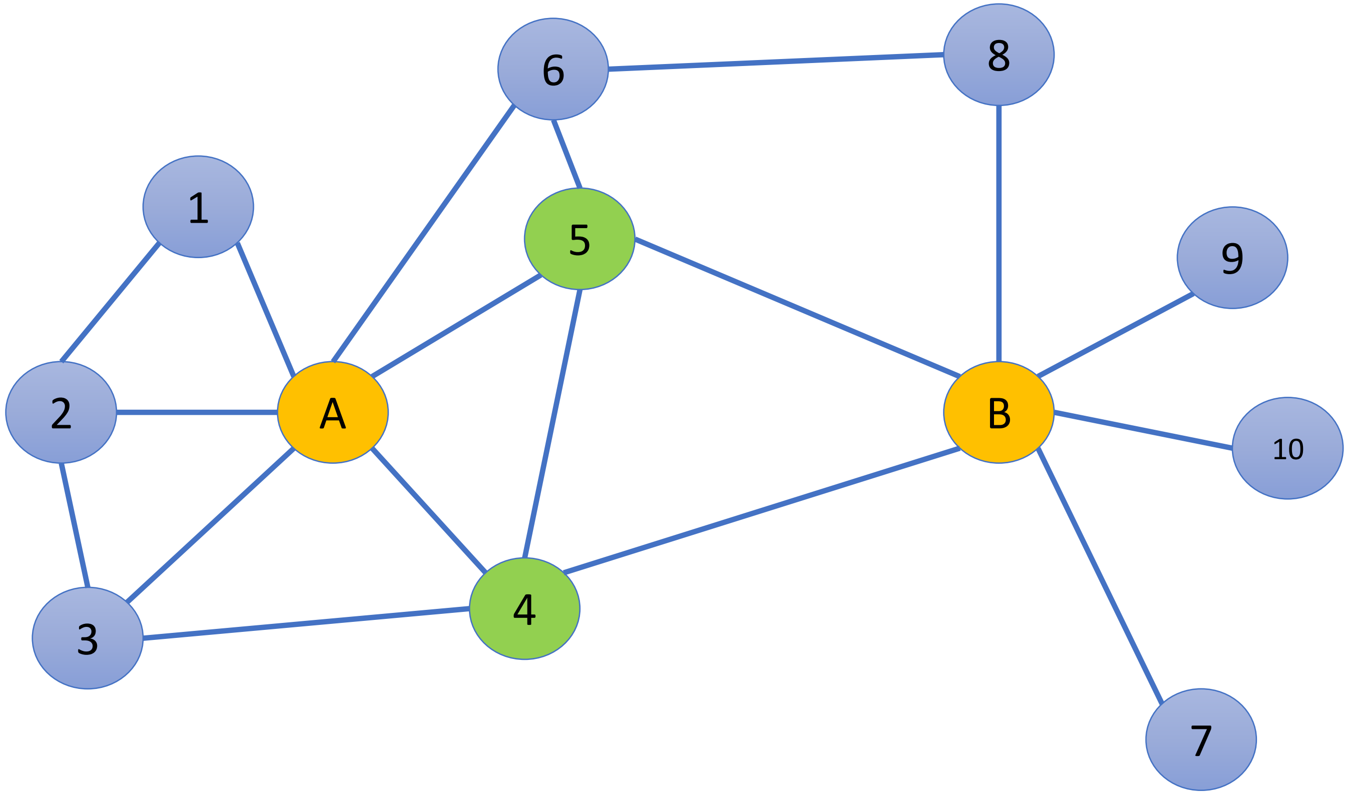
Jaccard Coefficient

- Use proportion of the common friends as a similarity metric
- $Jaccard(N(A),N(B)) = |N(A) \cap N(B)|/|N(A) \cup N(B)|$

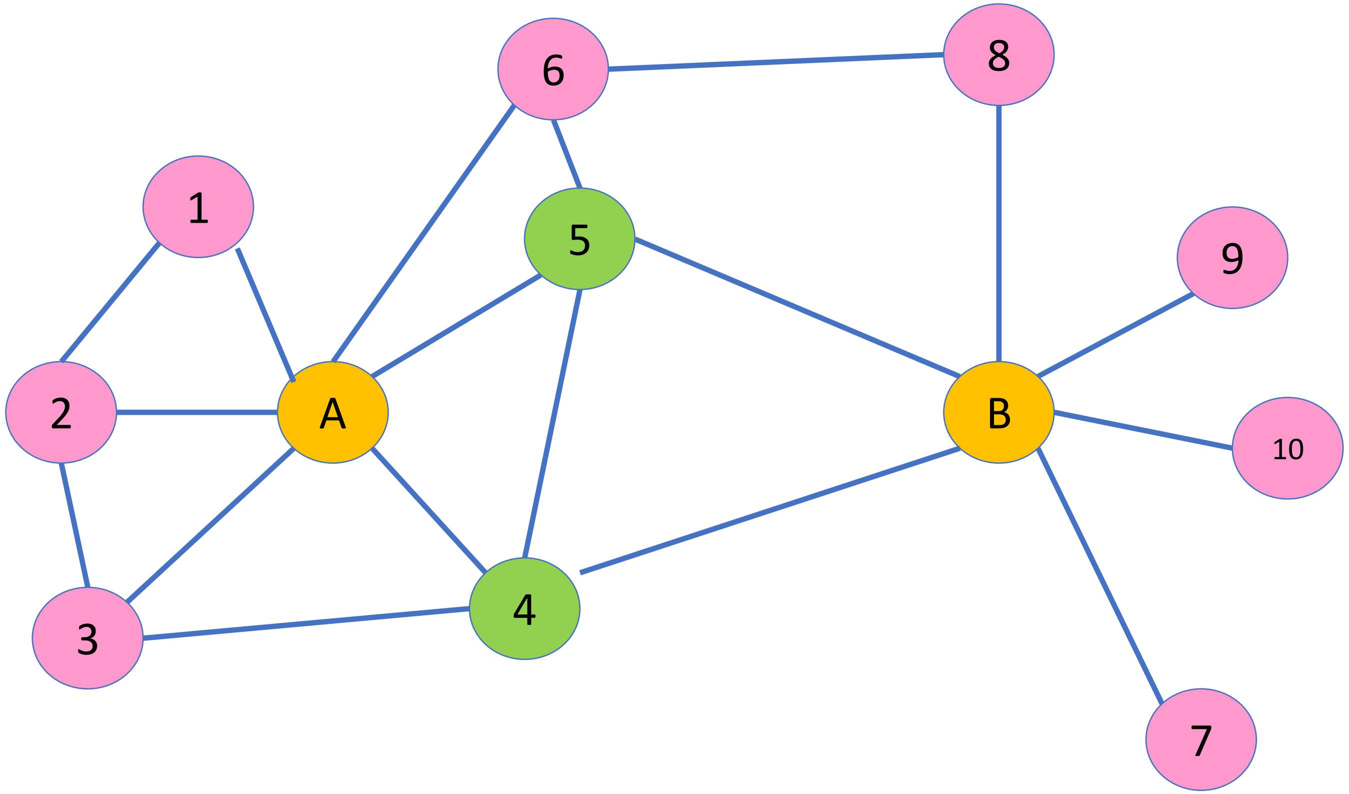
Example



Consider neighbors in-common

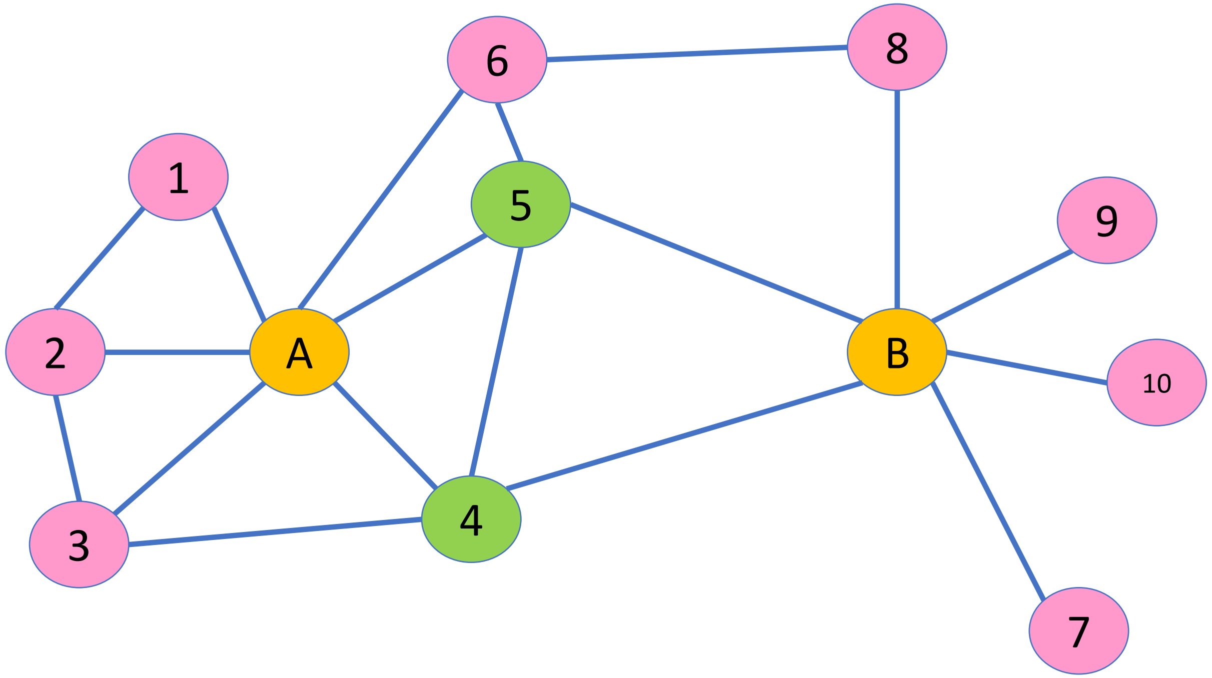


Consider neighbors not in-common

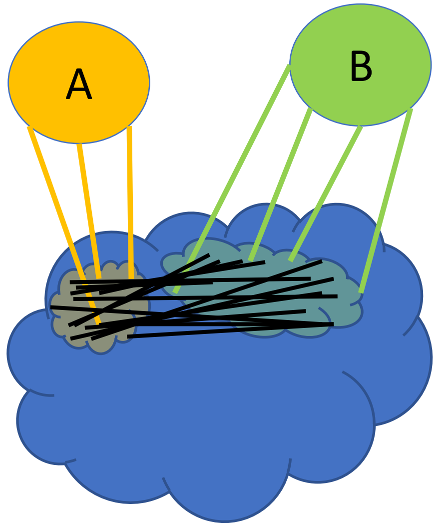


Jaccard Coefficient

- Jaccard(A,B) = 2/(2+8) = 20%

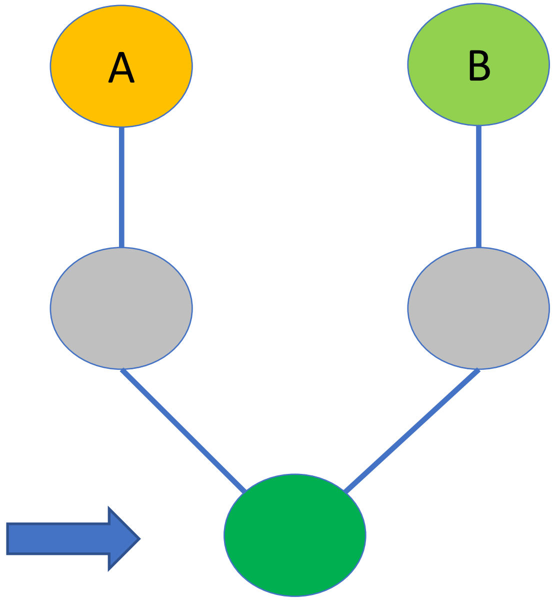


Caveat – 1: Common friends beyond 1-hop



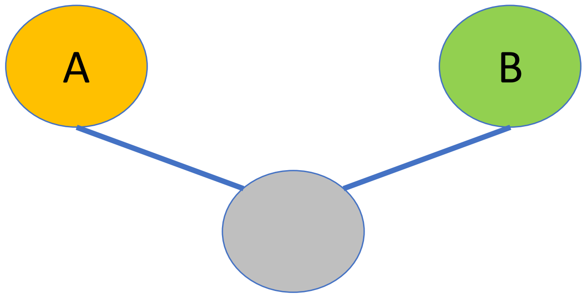
sim(A,B)=0

Simpler case:
common friend-of-friend

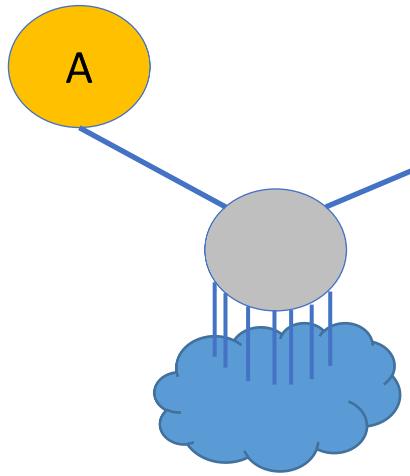


Should we increase sim(A,B)
because of this node?

Caveat – 2: Popularity of common neighbor



vs



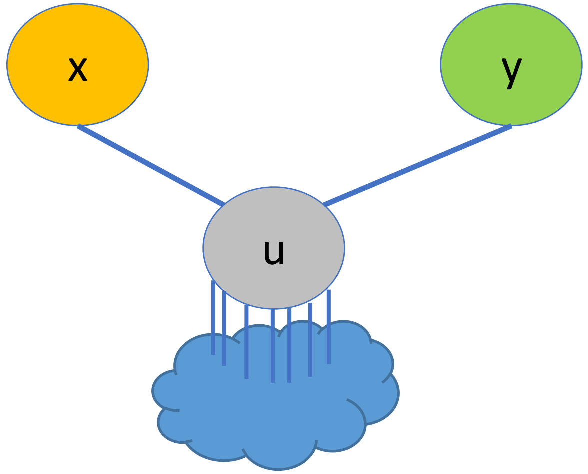
Adamic/Adar index

- The **Adamic/Adar index** computes the sum of the inverse logarithmic degree centrality of the neighbours shared by the two nodes

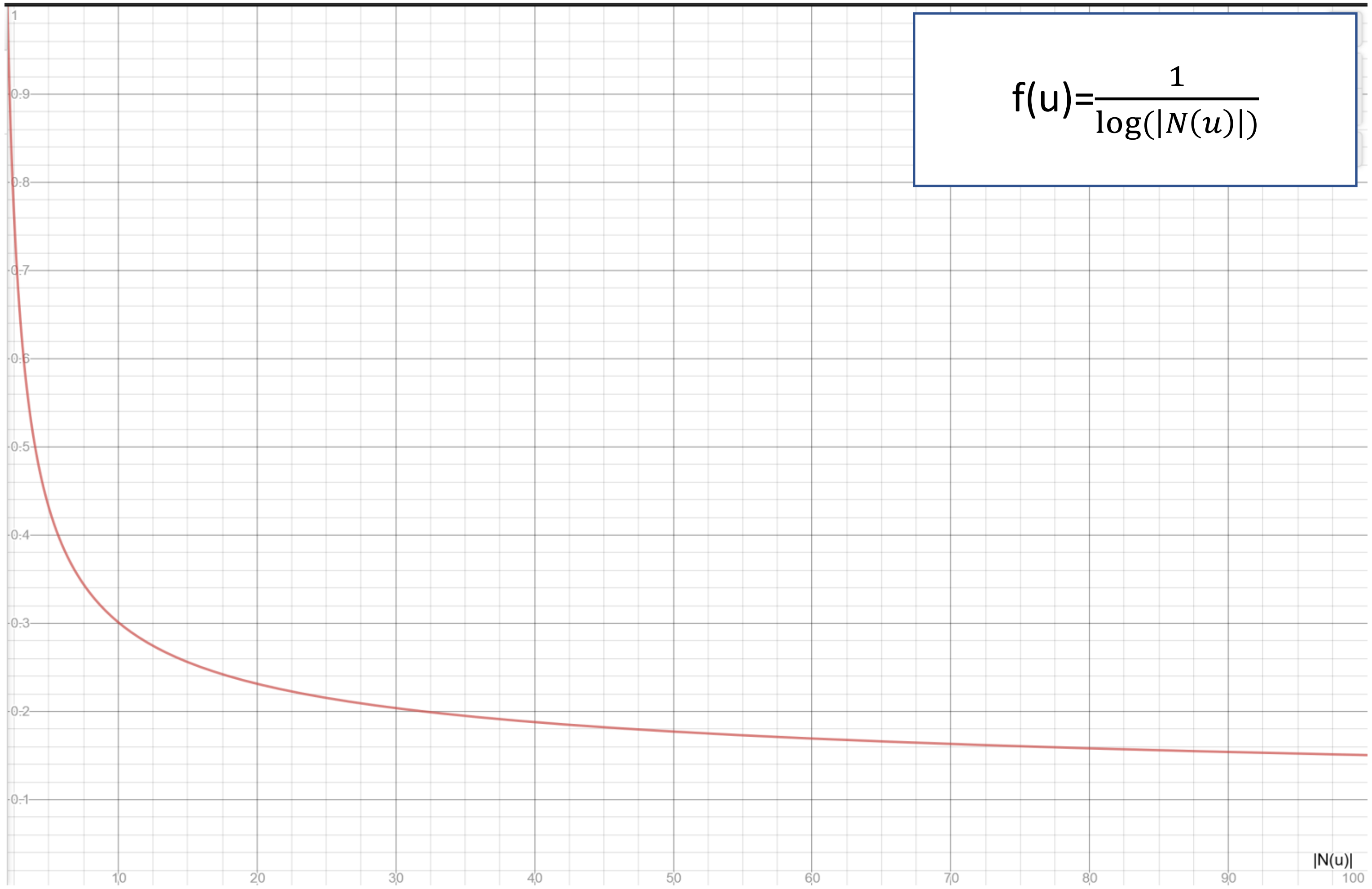
$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$$

N(x) : set of neighbours of node x

For $u \in N(x) \cap N(y)$: $|N(u)| \geq 2$ (why?)



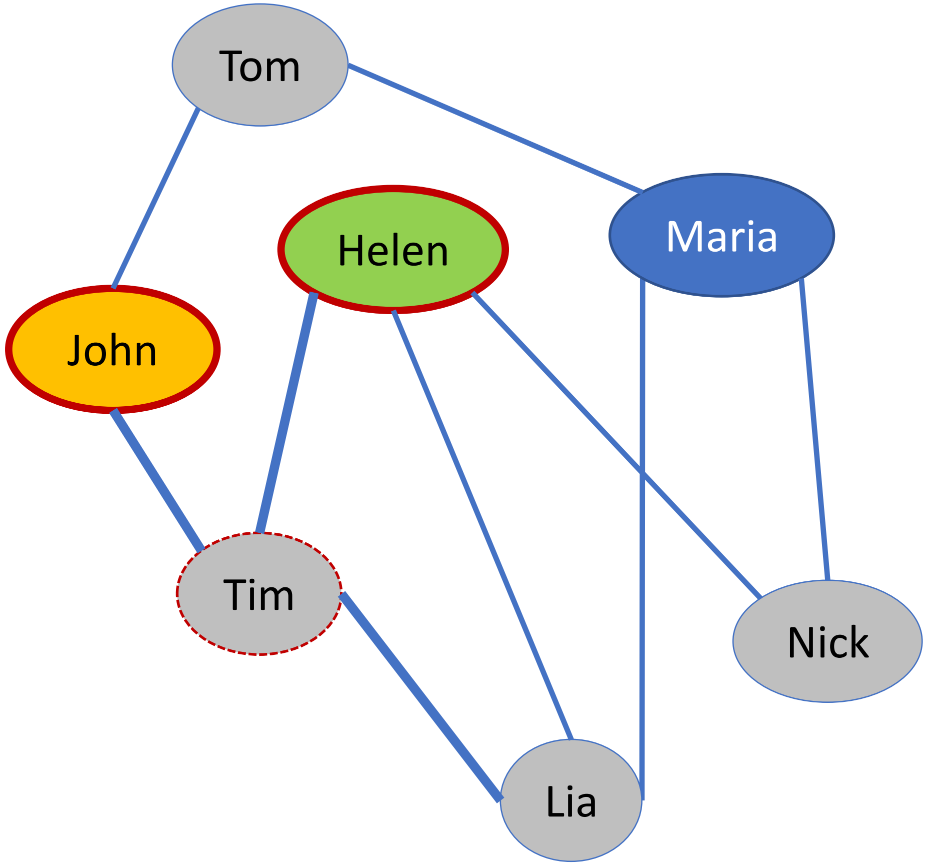
Reduce weigh of common neighbor (u) with many connections (|N(u)|)



Adamic/Adar Example

$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$$

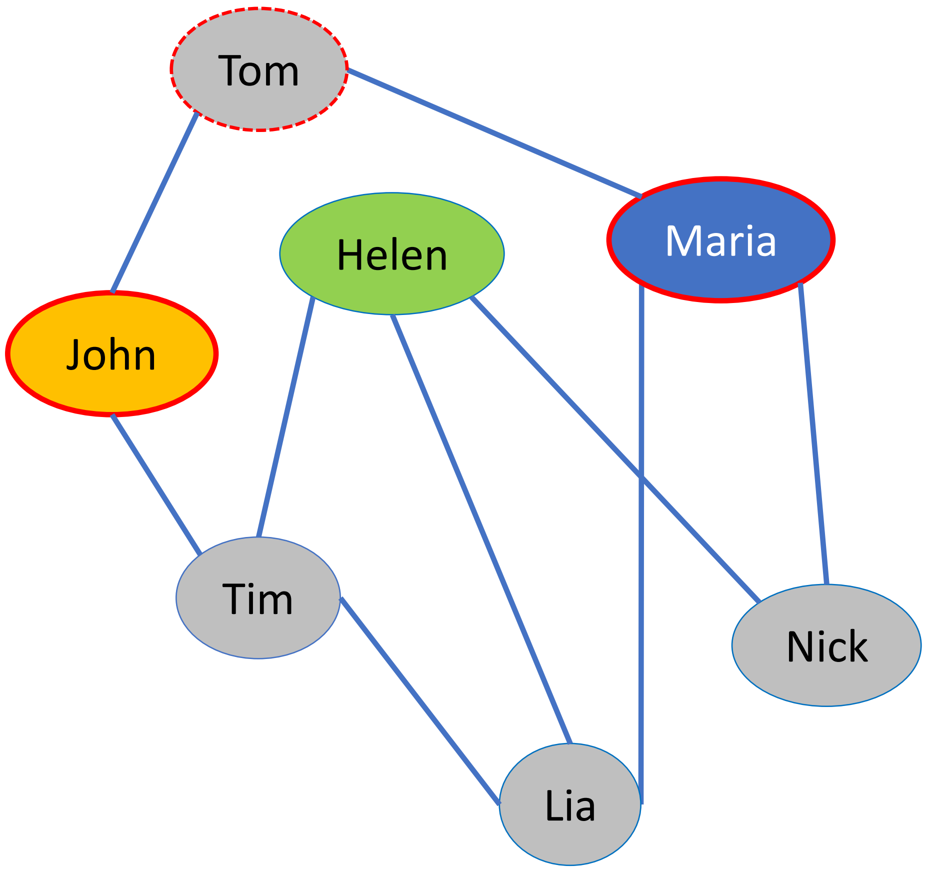
A(John,Helen) = 1/log(3) = 0.63



Adamic/Adar Example

$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$$

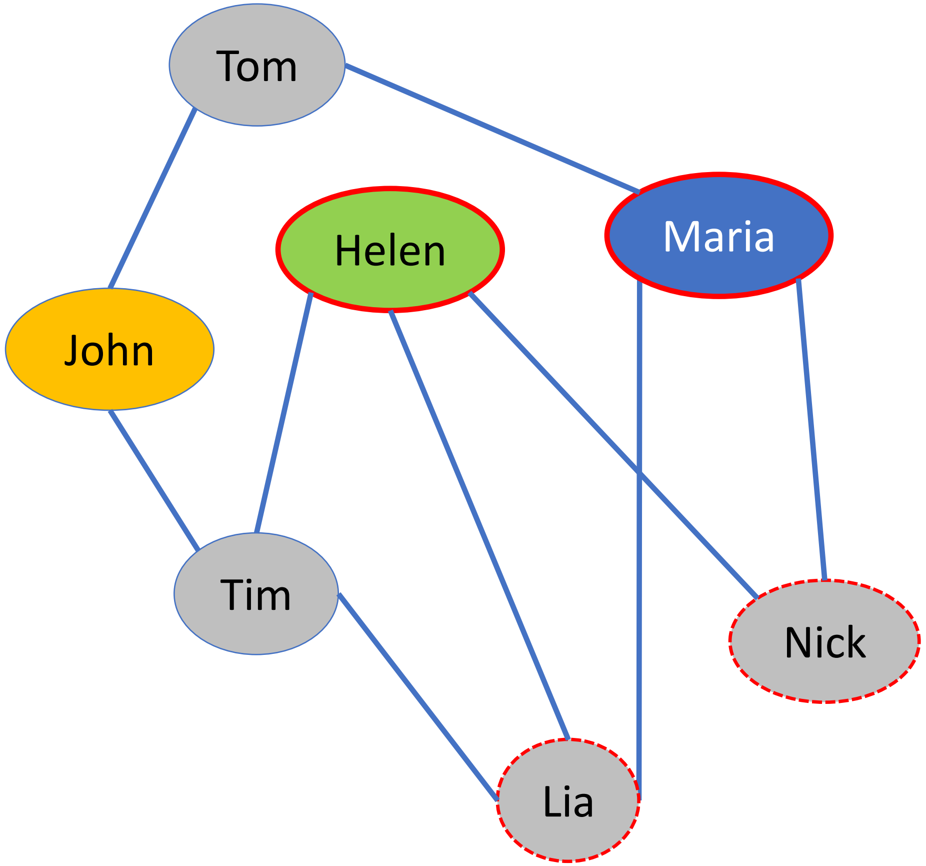
A(John,Maria) = 1/log(2) = 1



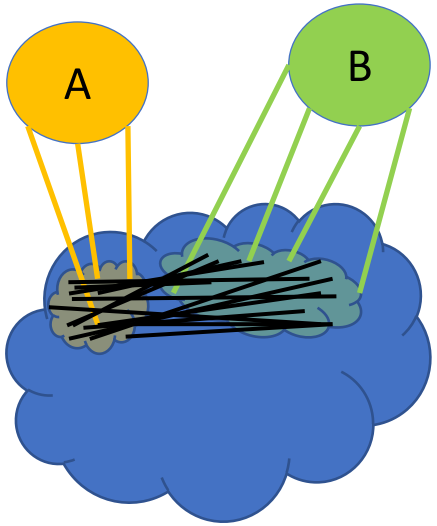
Adamic/Adar Example

$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$$

A(Helen,Maria) = 1/log(3) + 1/log(2) = 1.63

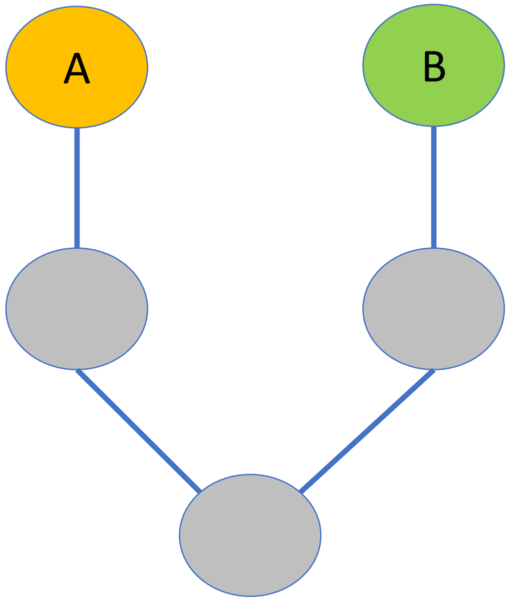


Caveat – 1: Need to look beyond 1-hop



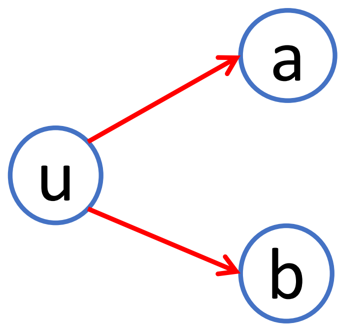
sim(A,B)=0

Simpler case:
common friend-of-friend



SimRank: two nodes are similar if they are referenced by similar nodes

- Recursive calculation in the same spirit as pageRank
- In the example bellow u references both a,b and this contributes to their similarity

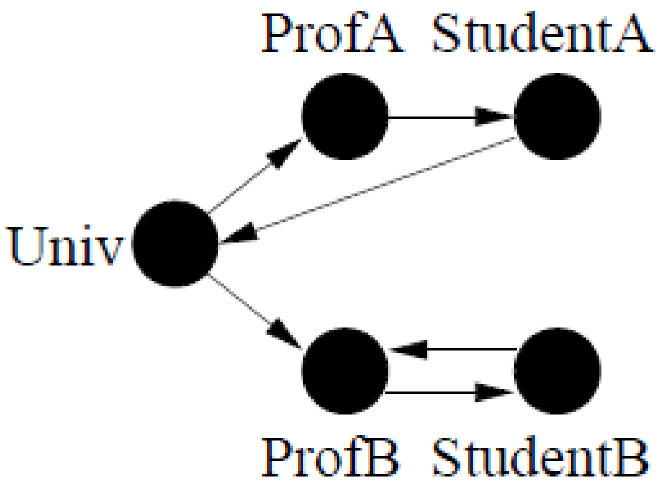


Motivation

- A similarity measure that exploits the object-to-object relationships found in many domains of interest
  - Web page X “points to” Web page Y
  - Customer “buys” product
  - Customer X transfers money to Customer Y
- May be used to cluster objects, such as for collaborative filtering in a recommender system

Intuition

- Concentrate on structural content
  - Can be combined with other similarity metrics that consider content similarity
- Two nodes are similar if they are referenced by similar nodes
  - Accounts X,Y are similar if they both receive money from some account Z



SimRank Recursive Computation

- Initialize:
  - $s(a,b) = \begin{cases} 1, \text{ if } a=b \\ 0, \text{ otherwise} \end{cases}$
- Iteratively compute ($a \neq b$):

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$$

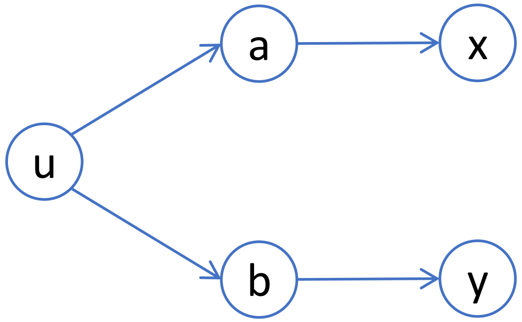
- Where
  - $I(x)$ = in-neighbors of x
  - $I_i(x)$ = i^{th} in-neighbor of x and $C<1$ (decay factor)

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$$

Explanation

- Nodes receive the average similarity of their in-neighbors multiplied by the decay factor C
- Special case: s(a,b) = 0 if |I(a)| = 0 or |I(b)|=0
  - i.e. nodes have no in-neighbors

Example

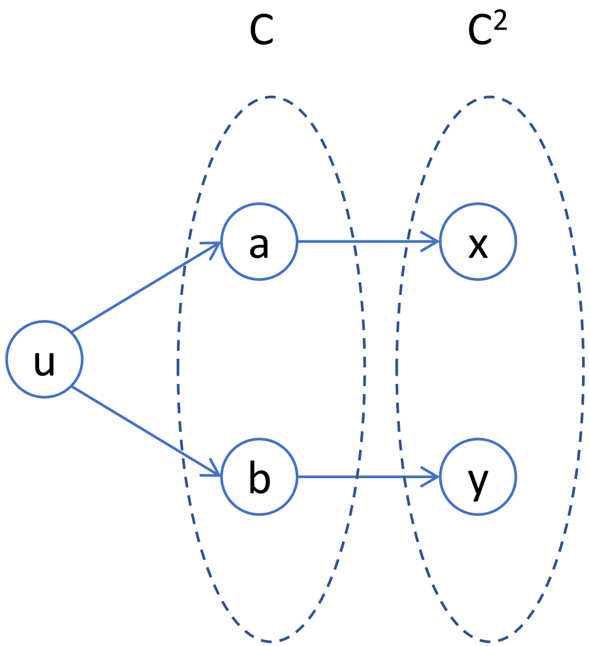


Initialization

s(u,u)=1
s(a,b)=0
s(a,x)=0
s(x,y)=0

Assume C=0.8

Iterate

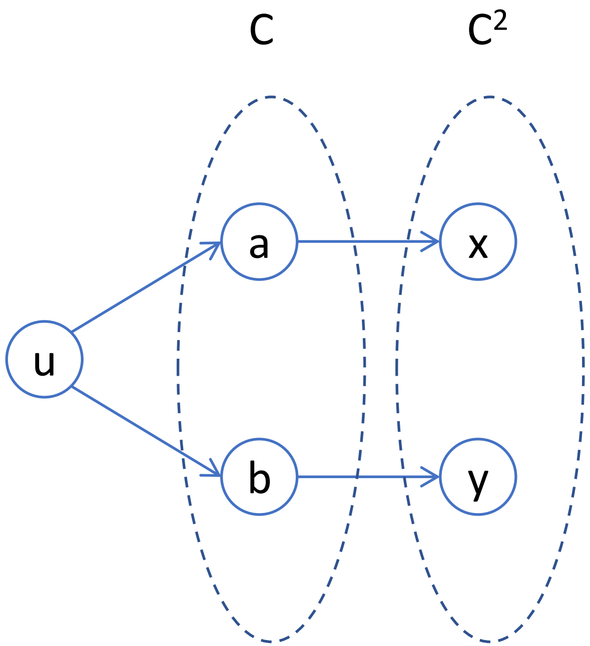


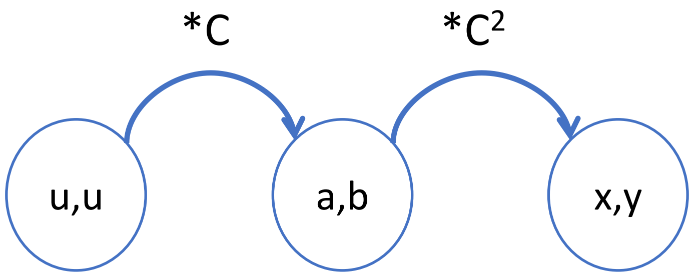
Updated SimRank

s(u,u)=1
s(a,b)=0.8*s(u,u)=0.8
s(a,x)=0.8*s(u,a)=0
s(x,y)=0,8*s(a,b)=0,8*0,8=0,64

Assume C=0.8

SimRank propagation

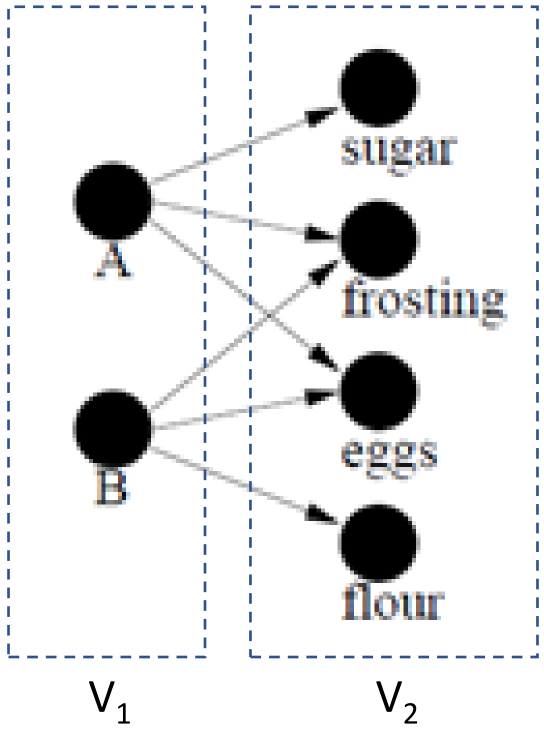




Assume C=0.8

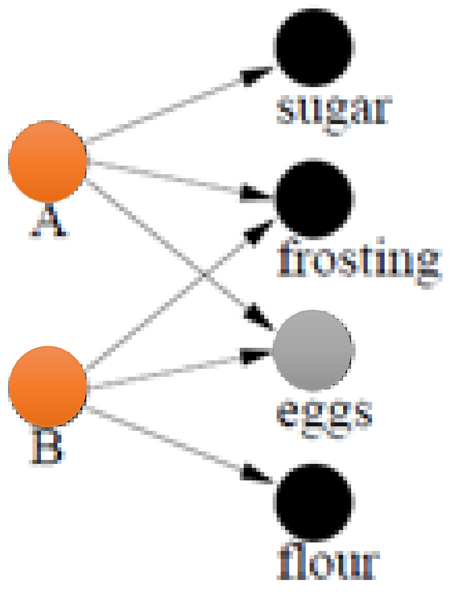
SimRank in bipartite graphs

- Bipartie graph: two disjoint classes of nodes V_1, V_2
  - e.g. V_1={customers}, V_2={items}
  - Edges only between nodes in V_1 to nodes in V_2



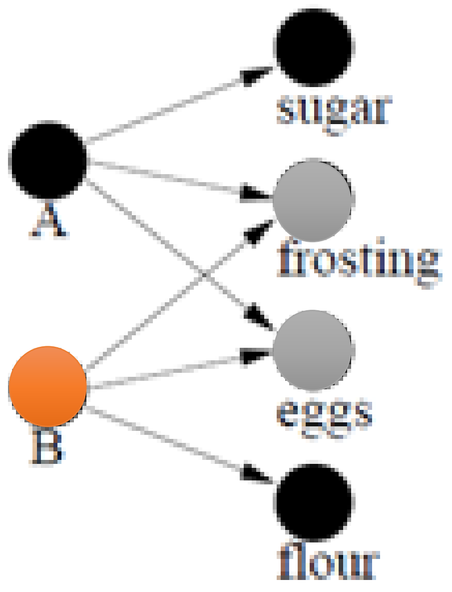
Intuition-1

- People are similar if they purchase similar objects



Intuition-2

- Items are similar if they are purchased by similar people



Bipartite SimRank

- SimRank between persons A and B, (A≠B)

$$s(A,B) = \frac{C_1}{|O(A)||O(B)|} \sum_{i=1}^{|O(A)|} \sum_{j=1}^{|O(B)|} s(O_i(A), O_j(B))$$

- SimRank between items x and y, (x≠y)

$$s(x,y) = \frac{C_2}{|I(x)||I(y)|} \sum_{i=1}^{|I(x)|} \sum_{j=1}^{|I(y)|} s(I_i(x), I_j(y))$$

- The similarity between persons A and B is the average similarity between the items they purchased
  - O(A) are the out-neighbors (items) for person A
- The similarity between items x and y is the average similarity between the people who purchased them

Modified SimRank in bipartite graphs