Entity Resolution and Link Prediction

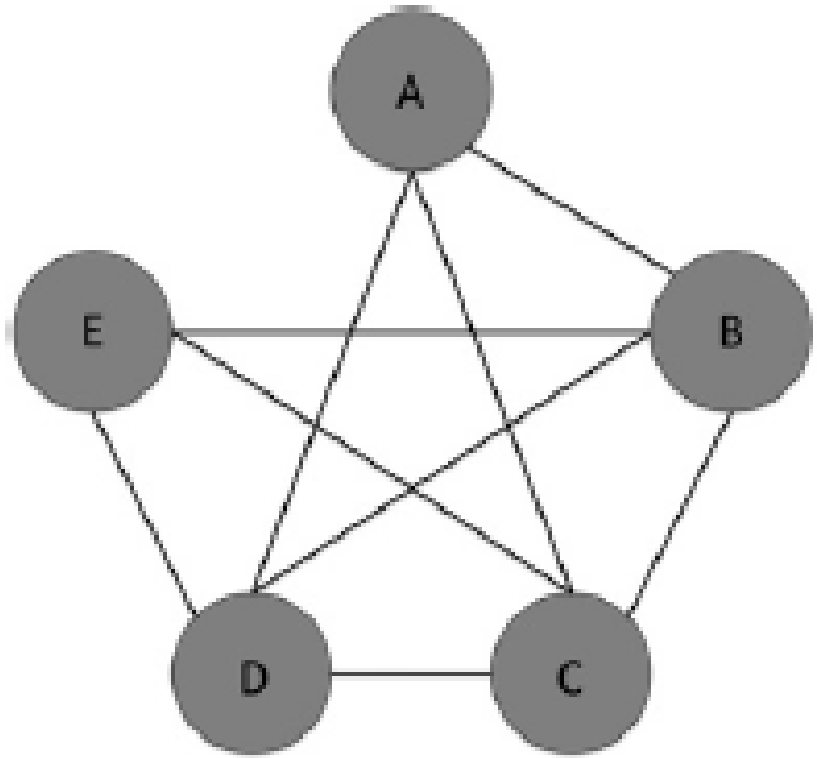
Social Network

Social Network

- Once a network is constructed there is often missing information and duplicate information

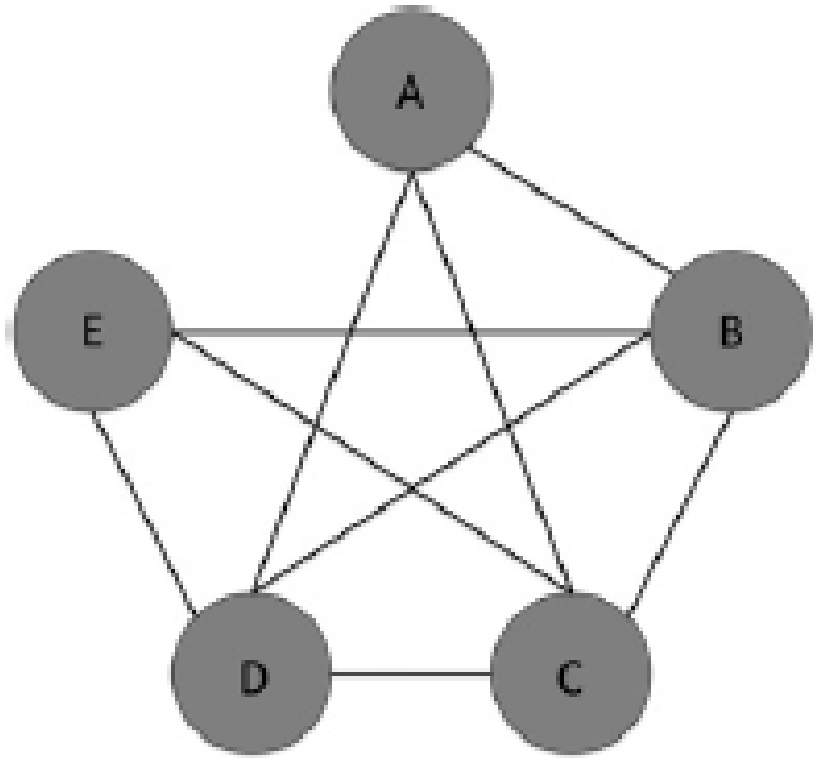
Social Network

- Once a network is constructed there is often missing information and duplicate information



Social Network

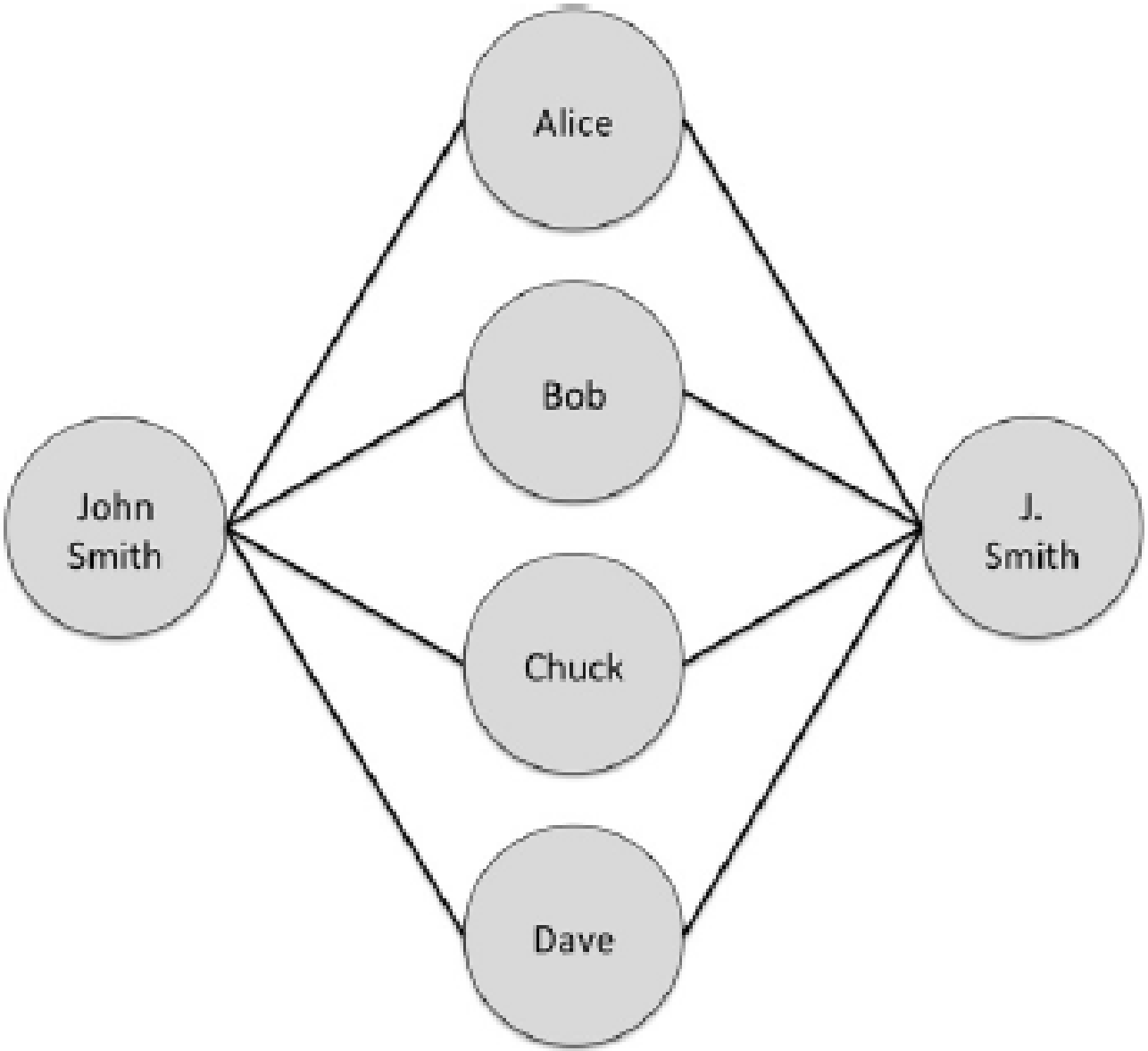
- Once a network is constructed there is often missing information and duplicate information



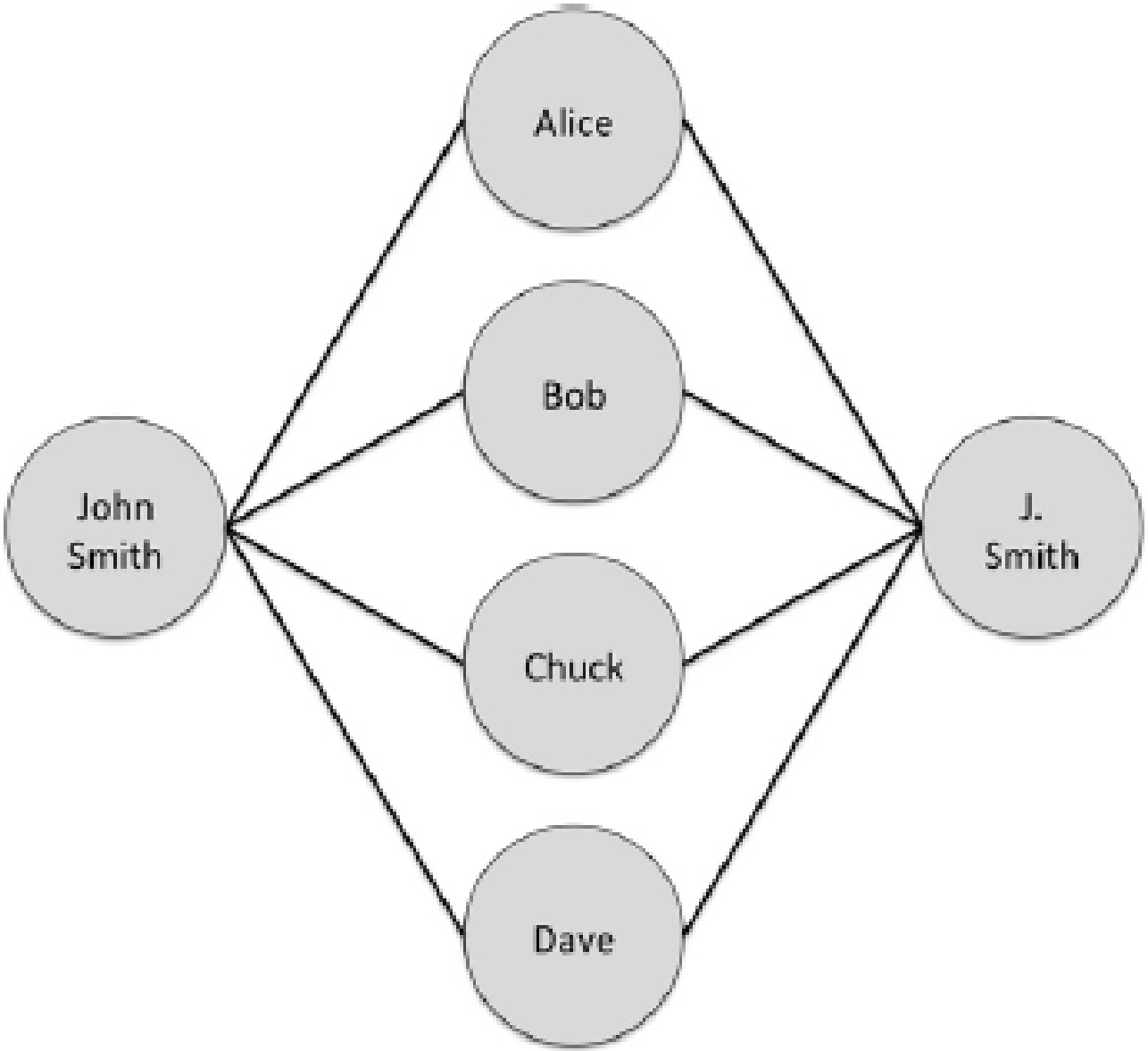
- Link Prediction is method of analysis that detects where missing links should be present

Social Network

Social Network



Social Network



Link Prediction

Link Prediction

- Analyze the network at a set point of time

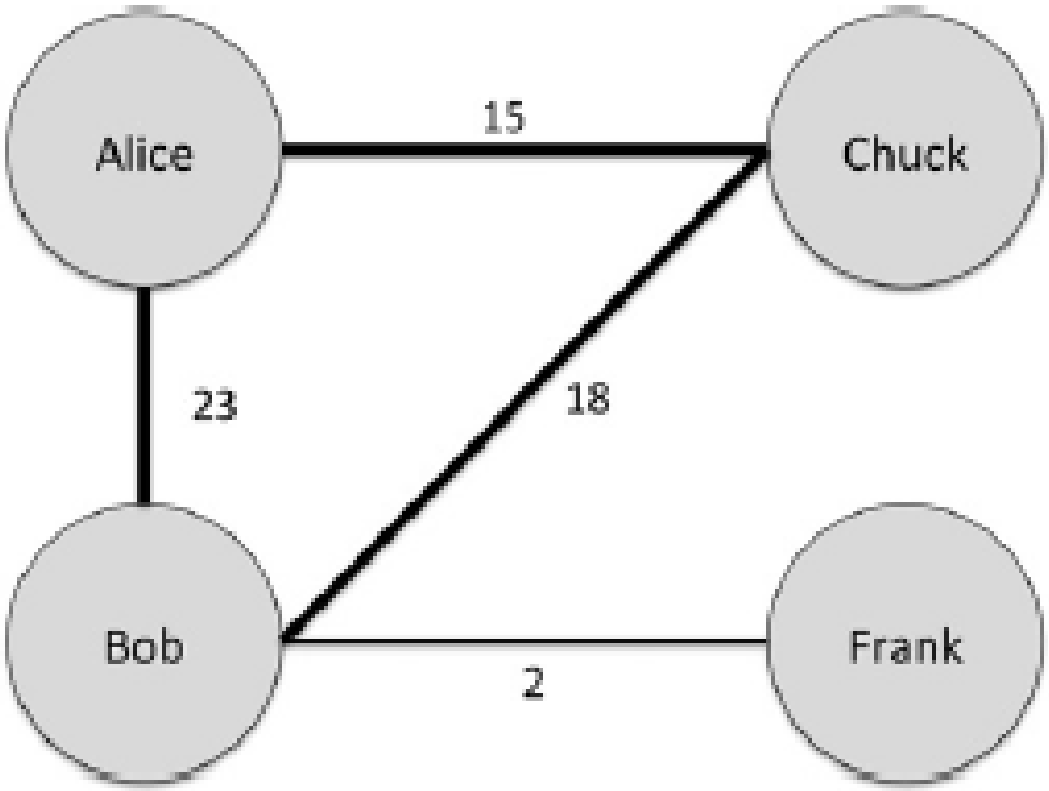
Link Prediction

- Analyze the network at a set point of time
- predict the future links

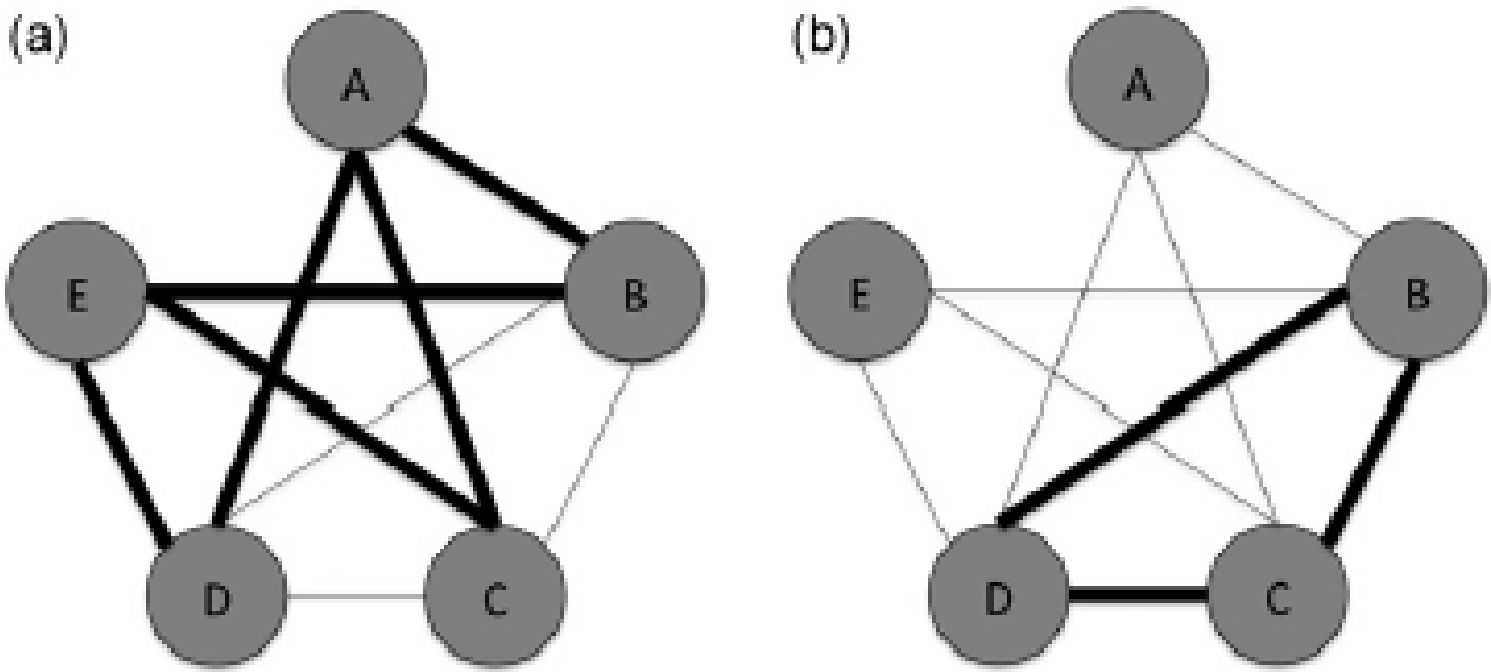
Link Prediction

- Analyze the network at a set point of time
- predict the future links
- useful in data cleaning for "missing links"

Link Prediction example



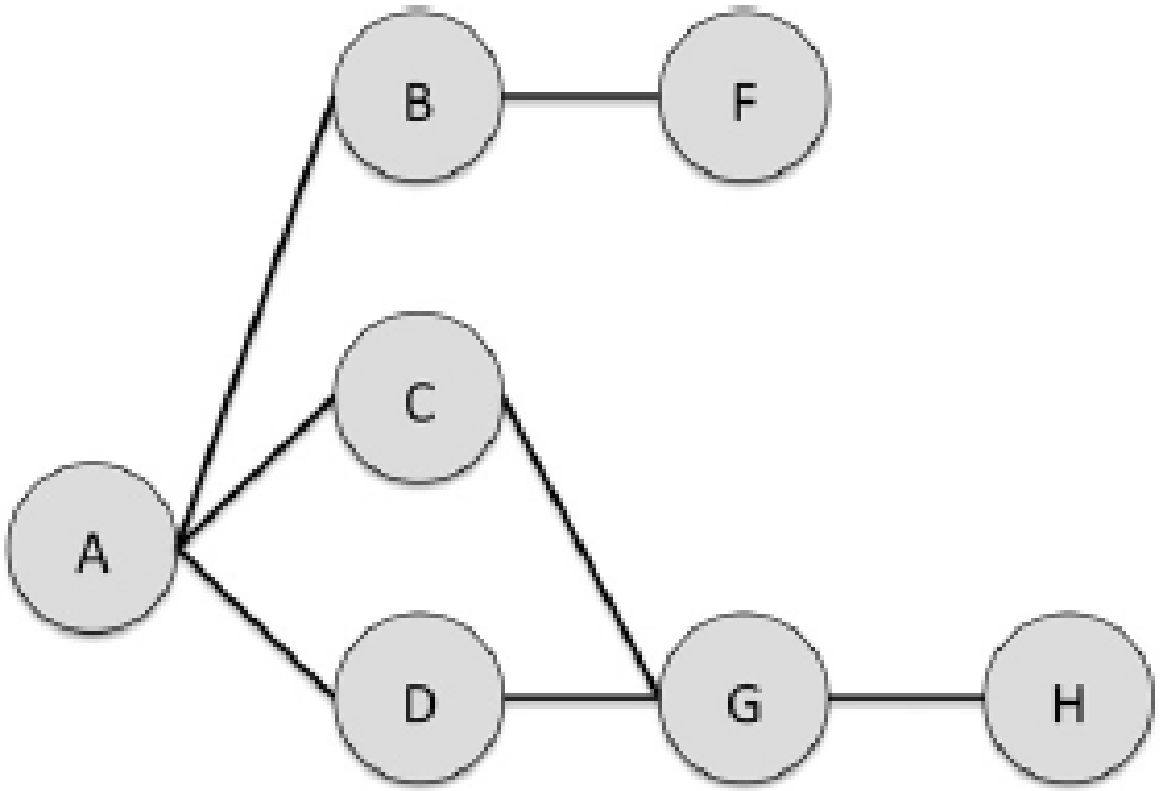
Missing Link



Link Prediction Algorithm

- score(A,B)

Link Prediction



Link Prediction Algorithm 1

score(A,B) = -ShortestPath(A,B)

Link Prediction Algorithm 2

$$\text{score(A,B)} = \text{Neighbours(A)} \cap \text{Neighbours(B)}$$

Link Prediction Algorithm 3

$$\text{score(A,B)} = \frac{|Neighbours(A) \cap Neigbours(B)|}{|Neighbours(A) \cup Neigbours(B)|}$$

This is known as Jaccard Index

Case Study : Jaccard Index

Say there are four nodes, Alice, Bob, Chuck and Dave
Let Alice and Bob be celebrities, each with 1 million friends
Chuck and Dave are average users with 100 friends each.
Say Alice and Bob have 2000 friends in common while Chuck and Dave have only 20 friends in common.

Case Study : Jaccard Index

Say there are four nodes, Alice, Bob, Chuck and Dave
Let Alice and Bob be celebrities, each with 1 million friends
Chuck and Dave are average users with 100 friends each.
Say Alice and Bob have 2000 friends in common while Chuck and Dave have only 20 friends in common.
$score(Alice, Bob) = \frac{2000}{1000000+1000000-2000} = 0.001$

Case Study : Jaccard Index

Say there are four nodes, Alice, Bob, Chuck and Dave
Let Alice and Bob be celebrities, each with 1 million friends
Chuck and Dave are average users with 100 friends each.
Say Alice and Bob have 2000 friends in common while Chuck and Dave have only 20 friends in common.

$score(Alice, Bob) = \frac{2000}{1000000+1000000-2000} = 0.001$

$score(Chuck, Dave) = \frac{20}{100+100-20} = 0.11$

Adamic and Adar, 2003

$$score(A, B) = \sum_{x \in Neighbours(A) \cap Neighbours(B)} \frac{1}{log(|Neighbours(x)|)}$$

Link Prediction Algorithm 5

preferential attachment

$score(A, B) = |Neighbours(A)| * |Neighbors(B)| =$
$degree(A) * degree(B)$

Advanced Link Prediction Techniques

- We can take the average ranking of each node pair from each measure and rank by that value
- The result would be a ranking that considers all the factors
- probabilistic models(Markov Networks)
- some approaches consider node's attributes in addition to network structure
- weighted and directed graphs
- machine learning