UNIT-4

ENTITY RESOLUTION AND LINK PREDICTION

Link Prediction

- link prediction is **the problem of predicting the existence of a link between two entities in a network**.
- Involves analysing the network at a set time and predict the edges that may appear in future.

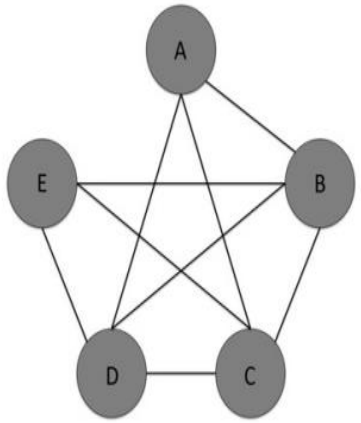


FIGURE 9.1

A network where all pairs of nodes but one are connected.

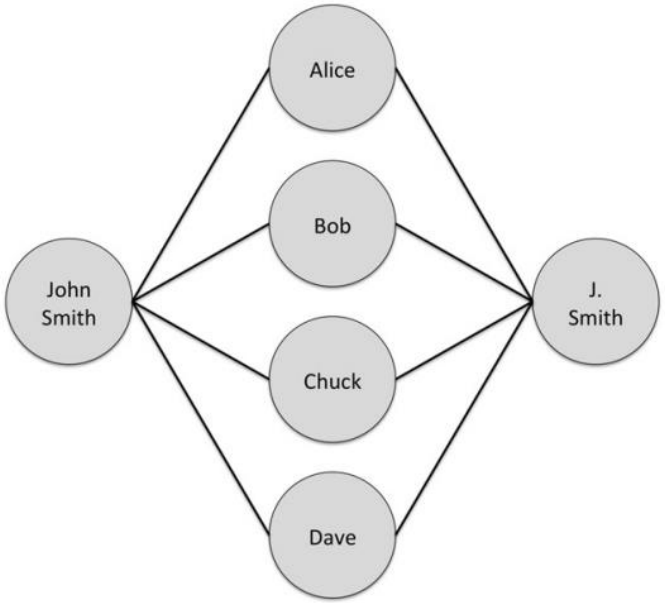


FIGURE 9.2

A network with two nodes, John Smith and J. Smith, who have similar names and acquaintances with no connection to one another. This could suggest that they are actually the same person.

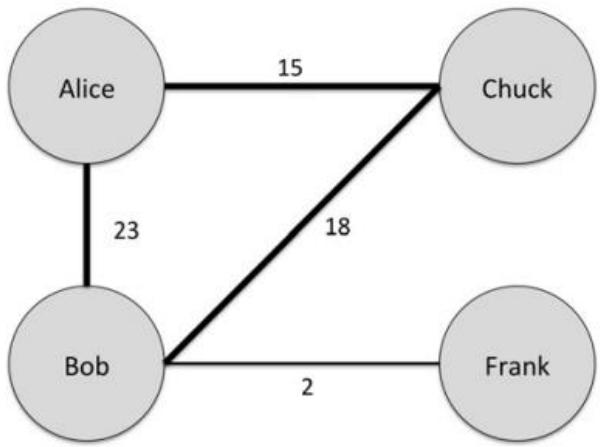


FIGURE 9.3

A network showing the frequency with which Alice, Bob, Chuck, and Frank attend meetings together.

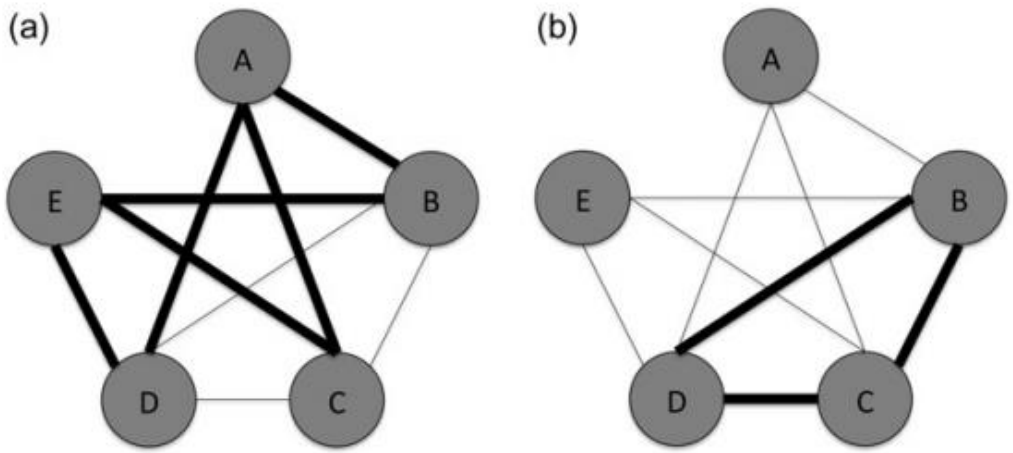


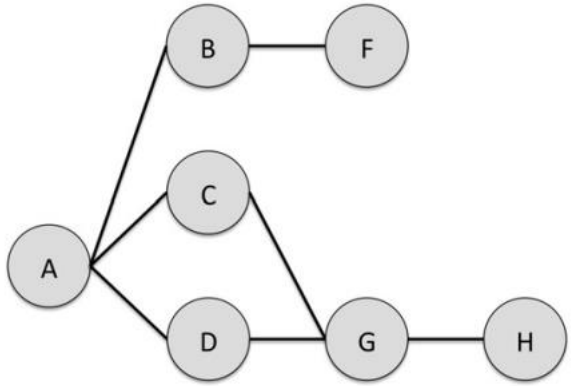
FIGURE 9.4

Two variations of the graph in Figure 9.1 where edge thickness indicates tie strength. In (a), nodes A and E have many shared strong ties, while in (b) they only share weak ties.

Mathematical notation

- A set is a collection of items.
- In graphs, the neighbors of a node are a set.
- For example, the neighbors of node A in Figure

are {B, C, D}.



- Let **Neighbors(A)** indicate the set of A's neighbors.
- If a set is written with vertical bars on either side, that refers to the size of the set.
- **|Neighbors(A)|** means the size of the set of A's neighbors. Since A has three neighbors, |Neighbors(A)|= 3.
- **|Neighbors(A)|= degree(A).**
- Intersection
- Union

- To sum the degree of each node who is neighbors with A

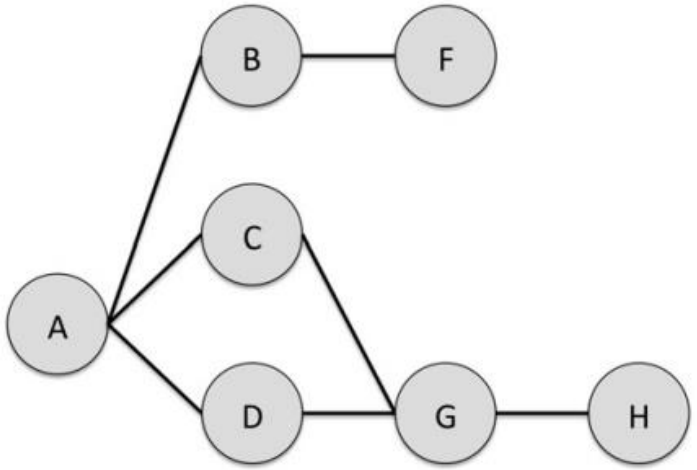
$$\sum_{x \in Neighbors(A)} degree(x)$$

- x∈Neighbors(A). That means x represents each item from the set.

Computing score

- One of the simplest ways to score the similarity or closeness of two nodes is to use the shortest path length between them.
- Nodes that are close to one another are more likely to create a relationship.
- However, as the average shortest path length increases, we want the score to decrease because nodes that are far apart (with a high average shortest path length) are less likely to be connected.
- We can use the negative value of the shortest path, so closer nodes have higher scores. Score(A,B) = shortestPath(A,B)

Pair	Score: -Shortest Path Length
A,F	2
A,G	2
B,C	2
B,D	2
C,D	2
C,H	2
D,H	2
A,H	3
B,G	3
C,F	3
D,F	3
B,H	4
F,G	4
F,H	5

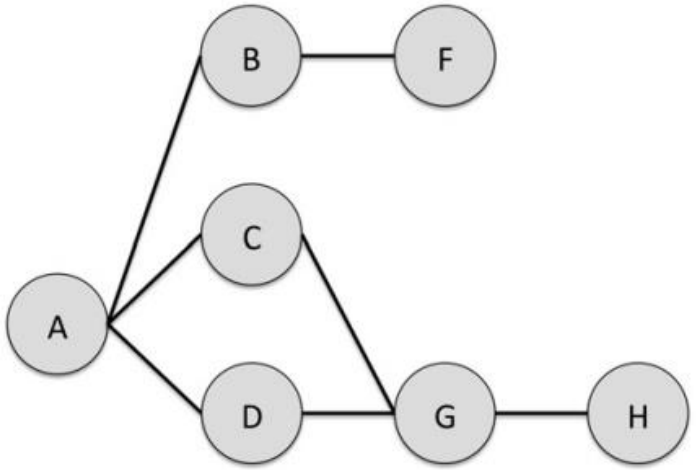


Many nodes are tied with a high score of 2.
If a simple rule is used to predict that edges will occur between nodes with the highest scores, then all these pairs (A,F), (A,G), (B,C), (B,D), (C,H), and (D,H) would have predicted edges between them.

- Another way of computing scores that uses more information from the network structure is to count the number of common neighbors between the two nodes in a pair.
- For the pair (A,B), we can represent this as the intersection of the set of nodes that are neighbors of A and the set of nodes that are neighbors of B.

$$score(A, B) = Neighbors(A) \cap Neighbors(B)$$

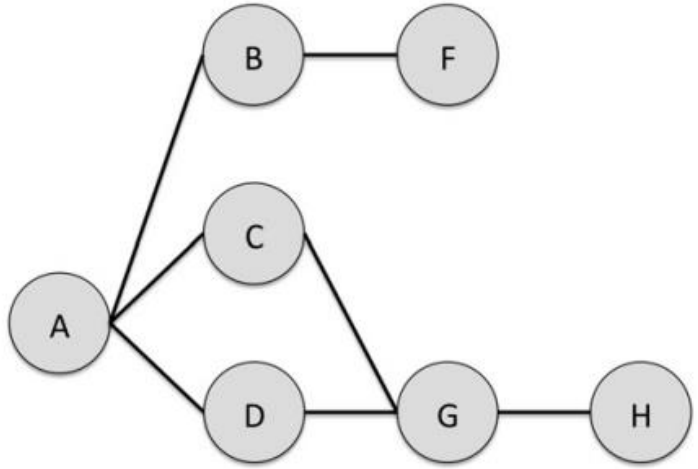
Pair	Score: Common Neighbors
A,G	2
C,D	2
A,F	1
B,C	1
B,D	1
C,H	1
D,H	1
A,H	0
B,G	0
B,H	0
C,F	0
D,F	0
F,G	0
F,H	0



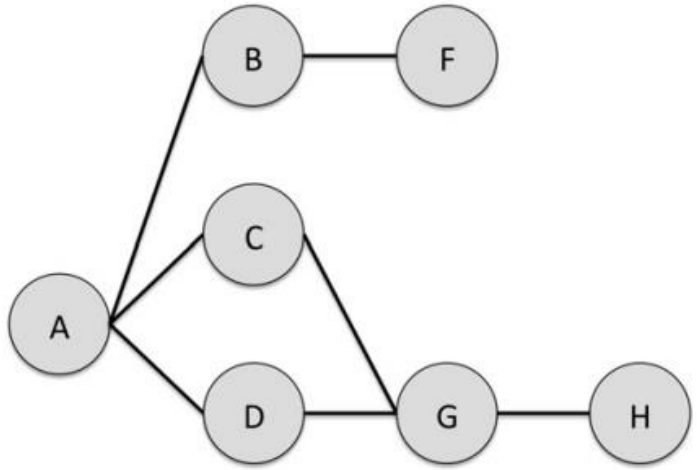
Two pairs, (A,G) and (C,D), have the high score.
Thus, these would be the only edges predicted when we apply this algorithm

- The **Jaccard Index** counts the total number of friends in common and divides that by the total number of people who are friends of either node.
- Nodes A and G have two friends in common.
- The total number of nodes who are friends with either A or G is four: nodes B, C, D, and H.
- Note that we do simply add the number of nodes who are friends with A (3) to the number of nodes who are friends with G (3), because this would count their mutual friends twice (nodes C and D). Instead, we are taking the union of their friends

$$score(A,B) = \frac{|Neighbors(A) \cap Neighbors(B)|}{|Neighbors(A) \cup Neighbors(B)|}$$



Pair	Score: Jaccard Index
C,D	1
C,H	0.50
D,H	0.50
A,G	0.50
A,F	0.33
B,C	0.33
B,D	0.33
A,H	0
B,G	0
B,H	0
C,F	0
D,F	0
F,G	0
F,H	0



- Nodes C and D have two common neighbors (A and G). Since these are the only neighbors of C and D, their score is 2/2 = 1.
- Thus, in this network, we would predict that the next edge appears between nodes C and D.

Case Study 1:

- Consider 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. Now say Alice and Bob have 2,000 friends in common while Chuck and Dave have only 20 friends in common.
- *The size of the union is the sum of the degrees minus the size of the intersection*.

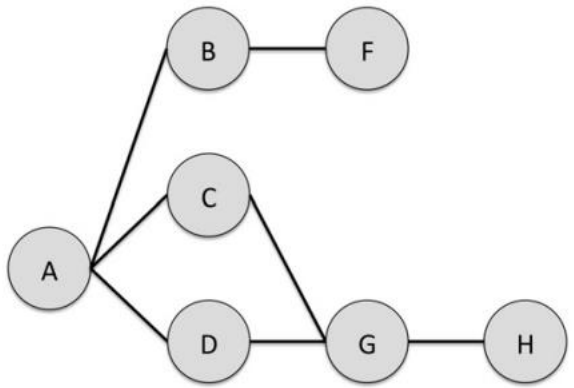
$$score(Alice, Bob) = \frac{2,000}{(1,000,000 + 1,000,000) \quad 2,000} = \frac{2,000}{1,998,000} = 0.001$$

$$score(Chuck, Dave) = \frac{20}{(100 + 100) \quad 20} = \frac{20}{180} = 0.11$$

- This example brings up another problem. What if the 20 people Chuck and Dave know in common are also celebrities?
- Adamic and Adar (2003) proposed amethod for dealing with this issue. They look at common friends and assign a score that gives more weight to people who have a few friends.

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

Pair	Score: Adamic/Adar
A,G	6.64
C,D	4.19
A,F	3.32
B,C	2.10
B,D	2.10
C,H	2.10
D,H	2.10
A,H	0.00
B,G	0.00
B,H	0.00
C,F	0.00
D,F	0.00
F,G	0.00
F,H	0.00

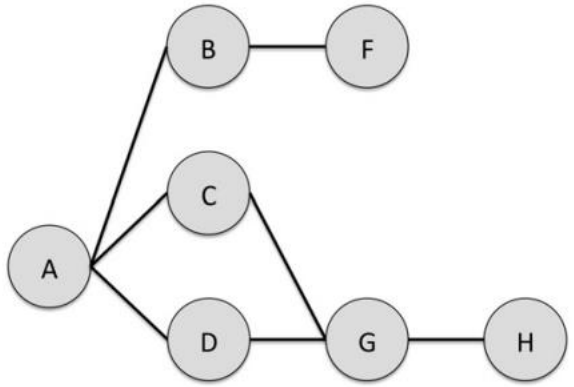


The clear winner here is (A,G). This method predicts that the next link to be added is between these nodes

- **Preferential attachment:**
- This network principle states that nodes with a high degree are more likely to gain new links.
- Popular nodes are more likely to gain new friends than less popular nodes.
- When predicting edges, preferential attachments suggest that nodes with high degree are more likely to gain new edges.

$$score(A,B) = |Neighbors(A)|^{*}|Neighbors(B)| = degree(A)^{*}degree(B)$$

Pair	Score: Preferential Attachment
A,G	9
B,G	6
B,C	4
B,D	4
C,D	4
A,F	3
A,H	3
F,G	3
B,H	2
C,F	2
C,H	2
D,F	2
D,H	2
F,H	1



$$score(A,B) = |Neighbors(A)|^{*}|Neighbors(B)| = degree(A)^{*}degree(B)$$

With this measure, we would predict that the next edge to appear will be between nodes A and G.

Advanced link prediction techniques

- There are many ways to make the link prediction more sophisticated.
- One could begin by combining the measures discussed earlier.
- For example, 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 described earlier.
- There are also probabilistic models for link prediction that are very successful. These often rely on a technique called Markov Networks . Some approaches consider nodes' attributes in addition to network structure. They can also work with weighted and directed graphs.
- Machine learning has also been effective when applied to this problem

Entity resolution

- **Entity Resolution** is a technique to identify data records in a single data source or across multiple data sources that refer to the same real-world entity and to link the records together.
- **Entity resolution** is a technique that tries to identify nodes that represent the same entity and then to merge them together.

- Most of them involve looking at the data

about the nodes, including their

attributes and relationships.

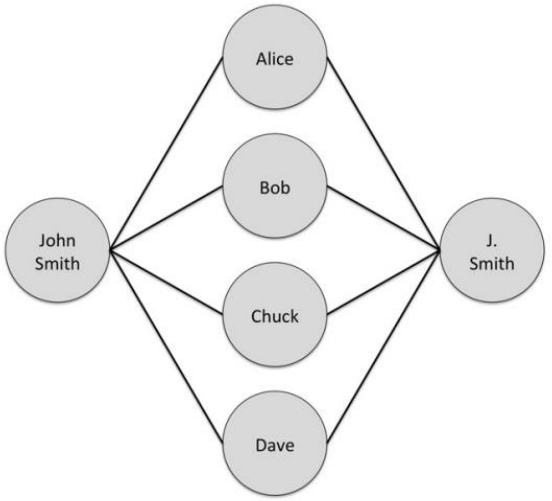
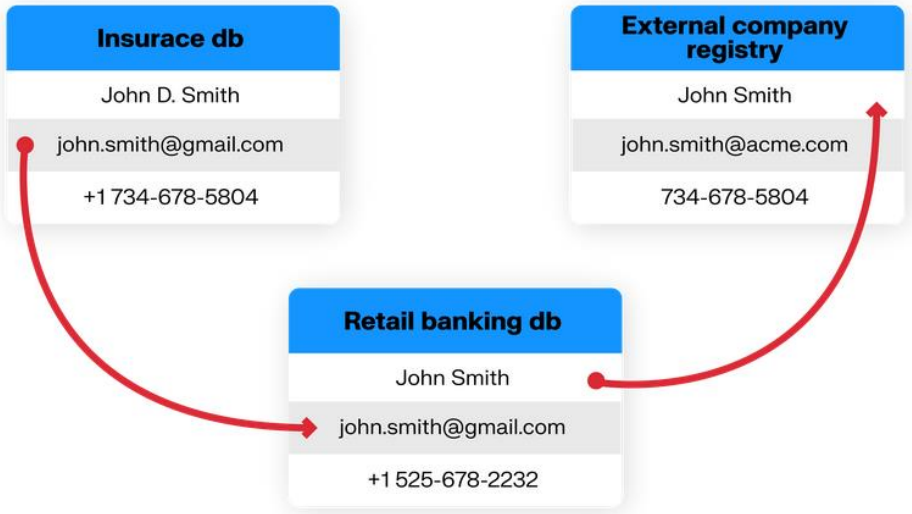


FIGURE 9.2

A network with two nodes, John Smith and J. Smith, who have similar names and acquaintances with no connection to one another. This could suggest that they are actually the same person.



- A corporate service provider can use entity resolution to resolve organization names despite the different representations, misspellings, abbreviations, and typographical errors.

Susheel Enterprises	Sushil Enterprises
Joylukkas	Joylukas
McDonald's	McD

- An insurance company needs to reconcile spelling variations, short forms, synonyms, and different word orders to correlate its data.
- Finance and banking firms require to match different numbers and percentage representations.

Table 9.1 Sample Personal Data for Four People

First Name	Last Name	Address	Birthday	SSN
J	Smith	123 Main St	1/6/68	123-12-1234
John	Smith	54 Elm St	January 1968	123-12-1234
Robert	Smith	123 Main St	1/6/1968	
JA	Smith	54 Elm St	March 1968	

Scoring Techniques:

- To create score for pair of nodes, we can consider similarity on set of attributes.
- Assign score=1 for attributes with match.
- Matching on some attributes is more important than others. So, assign more weight to such attributes. Ex:SSN of a person.
- To create a score for pair of nodes, determine that they match on given attribute and have weight for each attribute.
- Add positive weight for each attribute where the nodes match, and subtract the weight where nodes do not match.

How to find weight?

- The **weights must be higher** for attributes that are more definitive, like the SSN, and lower for attributes that are more commonly shared, like the month of birth.
- This is done with values called **u** and **m** probabilities.
- The **u probability** indicates the probability that two nodes will match on an attribute by chance.
  - For example, the probability that two nodes have the same birth month is 1/12. Thus the **u** probability for birth month equals 1/12 or 0.083.
  - *The probability of two nodes having the same last name is more complex to compute because the probability varies based on the last name itself.*
    - *For example, "Smith" is the most common last name in the United States, representing about 1% of all citizens' last names. Thus, the u probability for matching on the last name "Smith" is 0.01.*

- The **m** probability is the probability that two nodes that represent the same person will have the same value.
  - Often we expect this value will be 1.
  - For example, two nodes that are the same should have the same birthday, gender, SSN, and so on.
- However, the m value is not always 1. In some cases, like address or phone number, two nodes may indeed represent the same person but have different values.
- For example, one node could have personal/home information, and the other could have work information. Also, there may be missing attribute data.
  - For example, in Table, several nodes are missing SSNs. Thus, they could represent the same person, but if one has an SSN and the other does not, the values will not match.

- Setting the **m** probabilities will depend on the data in hand.
- In data given in table **m probability for SSN** is 0.95 (assuming there is more data than what is shown in), the m probability for address is 0.6, and the m probability for birth month is 0.98.
- **There will actually be two weights for each attribute**.
- The first is how much weight we add to the score if there is a match, and the second is how much weight we subtract from the score if there is no match. The common formulas are as follows:

For a match:

$$w = \ln(m/u)/\ln(2)$$

For a nonmatch:

$$w = \ln\left(\frac{1-m}{1-u}\right)/\ln(2)$$

Using the values we discovered above, the weight for a match on birth month would be:

$$\ln(0.98/0.083)/\ln(2) = 2.469/0.693 = 3.56$$

The weight for a no-match on birth month would be:

$$\ln\left(\frac{1-0.98}{1-0.083}\right)/\ln(2) = \ln\left(\frac{0.02}{0.917}\right)/\ln(2) = \quad 3.825/0.693 = \quad 5.520$$

- Perform this calculation for every attribute in the table.
- Then, check for matches and add the appropriate weights for a match or nonmatch to compute a final score.

Incorporating network data

- To quantify how similar these nodes are to one another structurally, examine their egocentric networks and compare them.
- Specifically, we want to compare the neighbors of one node to the neighbors of the other.
- Thus, we can use many of the same scoring mechanisms from link prediction to quantify how similar a pair of nodes are to one another.
- For entity resolution, the number of common neighbors, the Jaccard Index, the Adamic/Adar method, and preferential attachment all compared the neighbors of one node with those of another.
- The results from these similarity measures can be used in addition to attribute data.
- For example, if two nodes are very similar in their attribute data but have very little similarity in the network, we can reduce the similarity score.
- A high similarity on the network may make up for lower similarity in attribute data as well. Network and attribute data can be considered as separate steps, or the network data score can receive its own weight for use in the sum above.

More sophisticated entity resolution

- There are some ways to iterate on the relatively simple methods introduced earlier.
- One approach is to allow for partial matches.
- Ex: "John Smith" and "J Smith" example, while their first names are not an exact match, they are close. Since "J" is the correct first initial for "John," we may label this a partial match. Then, instead of adding the weights for items that match, we can add part of the weight for a partial match. For example, if we say "J" is a 0.3 match for "John," then we could add 0.3 times the weight for a name match to the score. We would also have the option of subtracting 0.7 times the non-match score.

$$p^* w_{match} + (1 - p) w_{non\ \ match}$$

- Consider the weight for a matching first name is 5.5 and the weight for a non- matching first name is 3.2.
- If we did not give any credit for a partial match, then we would simply subtract 3.2 from the score. But if there is a 0.3 match on the first name, then the score becomes

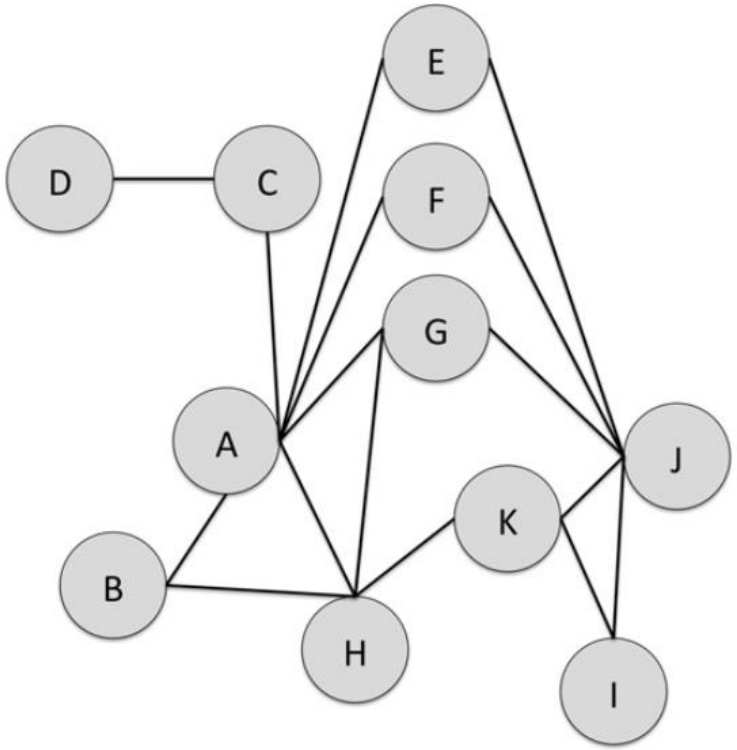
$$0.3^*5.5 + 0.7^* \quad 3.2 = 1.65 \quad 2.24 = \quad 0.59$$

This partial match allows to give much more credit to the pair, subtracting only 0.59 instead of 3.2.

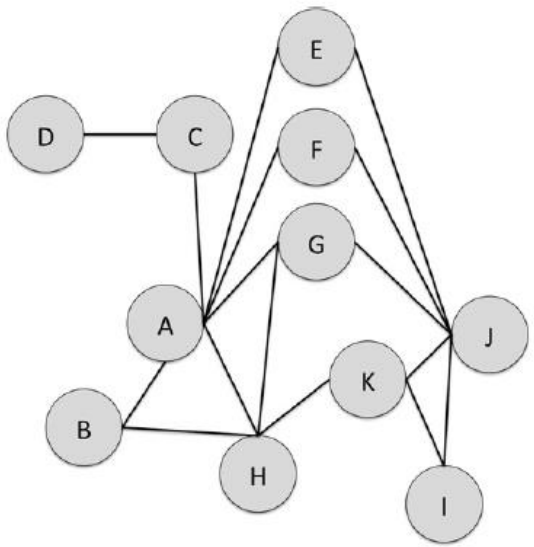
More sophisticated entity resolution

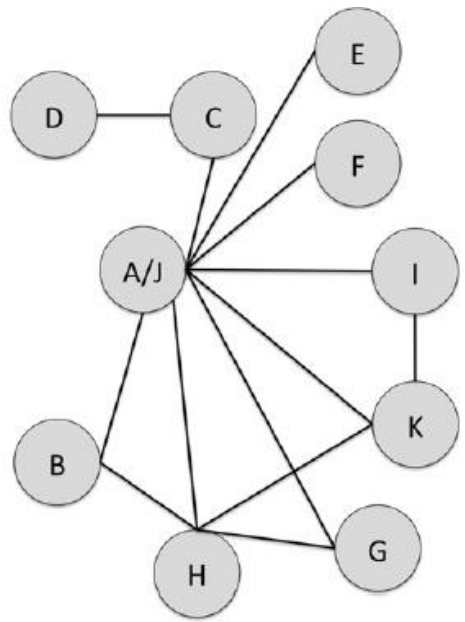
- Nodes A and J have three common neighbors: nodes E, F, and G.
- Nodes B and D have no common neighbors. Nodes E and I have one node, J, as a common neighbor. And so on..

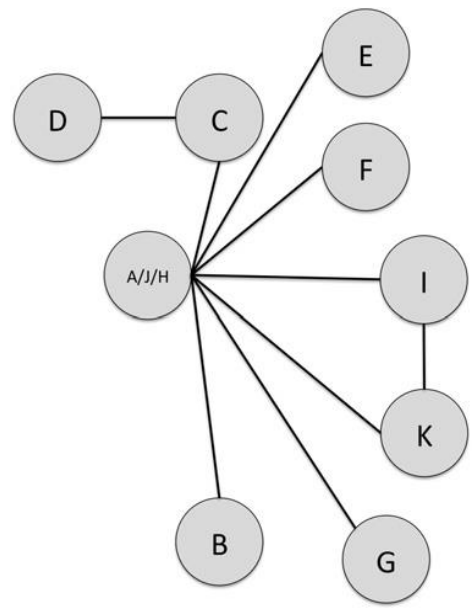
- ***A graph in which consider whether or not to merge nodes. The examples will consider merging A and J, B and D, and E and I.***



- A second,more sophisticated step is that we can do repeated iterations of entity resolution..







- A second,more sophisticated step is that we can do repeated iterations of entity resolution..

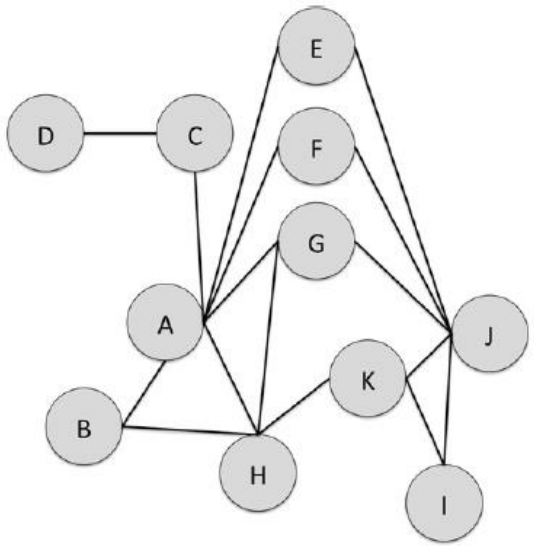


Table 9.2 Values for each Example Node Pair and the Associated Similarity Measures

	Node Pair		
	A,J	**B,D**	**E,I**
Common Neighbors	3	0	1
Jaccard Index	0.38	0	0.33
Adamic/Adar	9.97	0	1.43
Preferential Attachment	25	1	4

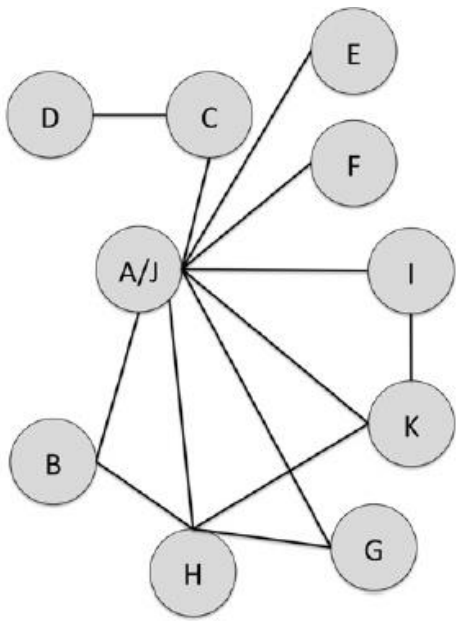


Table 9.3 Measures of Network Similarity for Nodes on the Merged Network Shown in Figure 9.7

	Node Pair			E,I
	Previous A/J	**A/J, H**	**B,D**	
Common Neighbors	3	3	0	1
Jaccard Index	0.38	0.43	0.00	0.50
Adamic/Adar	9.97	9.97	0.00	1.11
Preferential Attachment	25	32	1	2

Note that the values for the pair E and I have also changed because of the merger in the network.

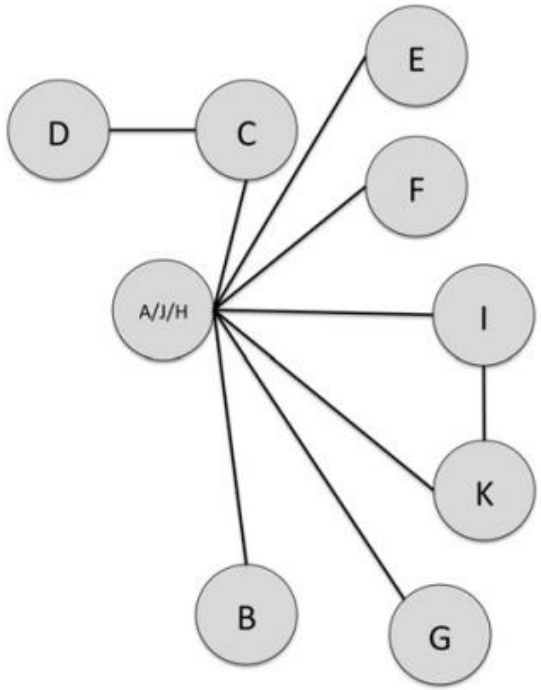


Table 9.3 Measures of Network Similarity for Nodes on the Merged Network Shown in Figure 9.7

	Node Pair			E,I
	Previous A/J	**A/J, H**	**B,D**	
Common Neighbors	3	3	0	1
Jaccard Index	0.38	0.43	0.00	0.50
Adamic/Adar	9.97	9.97	0.00	1.11
Preferential Attachment	25	32	1	2

Note that the values for the pair E and I have also changed because of the merger in the network.

Link prediction: Case study—Friend recommendation

FIGURE 9.9

A suggestion about people to follow made by Twitter.

- Many social networking and social media websites have a feature that recommends friends.
- For example, above figure shows Twitter's "Who to follow" recommendation.
- There are many techniques, and **link prediction** is one way to do it.

- Considers all unconnected pairs of nodes in the network and generates a score for each.
- Those scores can be used to add the top-scoring link to the graph, or they can be considered a ranked list of potential edges to add.
- For friend recommendation, consider all edges all possible edges for a specific user.
- When that user logs in, the system can compute a score for each pair comprising the user and every other node in the network. Then, the pairs can be sorted from highest to lowest score, and the other node in the top-scoring edges becomes a recommended friend.
  - For large networks, computing scores for every pair of nodes can be computationally expensive and take s a long time.
  - For example, Facebook has over a billion users. Running 1 billion calculations takes a long time, especially if the system needs to get the friend list for every person.
  - If the system uses this as a limit on number of common neighbors, then the only nodes that need to be considered as candidate friends for the user are the users' friends' friends. That greatly cuts back on the number of possible pairs to score, making the computation much faster.

- Note that link prediction results are not necessarily the only thing to consider when recommending friends.
- Looking at similarity of node attributes can add valuable information.
- While the interesting attributes will be different from those in entity resolution, the techniques for using them may be similar.
- For example, when recommending friends, we might look for people with matches on interests, educational background, favorite sports teams, and so on. We can create weights for matches on each of those attributes and use them in a score, just as we used weights on matching personal information to conduct entity resolution. Combining these attribute-based insights with the link prediction results will often lead to better friend suggestions

Entity resolution: Case study—Finding duplicate account

- When people sell things online on sites such as eBay, or Amazon, the transaction requires that the buyers trust them.
- The seller's reputation is extremely important for the transaction to go well.
- When sellers develop a poor reputation, a common "solution" is to open another account.
- In some cases, sellers will develop good reputations in many accounts by selling small items, leveraging that reputation to sell a few big items at which point they defraud the buyers, absconding with the money and closing the account.

- To protect buyers, companies that host online sales want to ensure that people are not maintaining multiple accounts without an obvious link between them.
- Knowing which accounts belong to the same person allows the company to track the good and bad actions of each unique person and to have the power to suspend all the accounts if the seller does something very bad on one of them, or if the sum of bad behavior across the accounts crosses some threshold.
- Entity resolution works well for this task. User attributes, like financial information and addresses, are often very distinctive and can help identify the accounts' owner. Network information can also be included, especially when the accounts are linked to the same products or customers