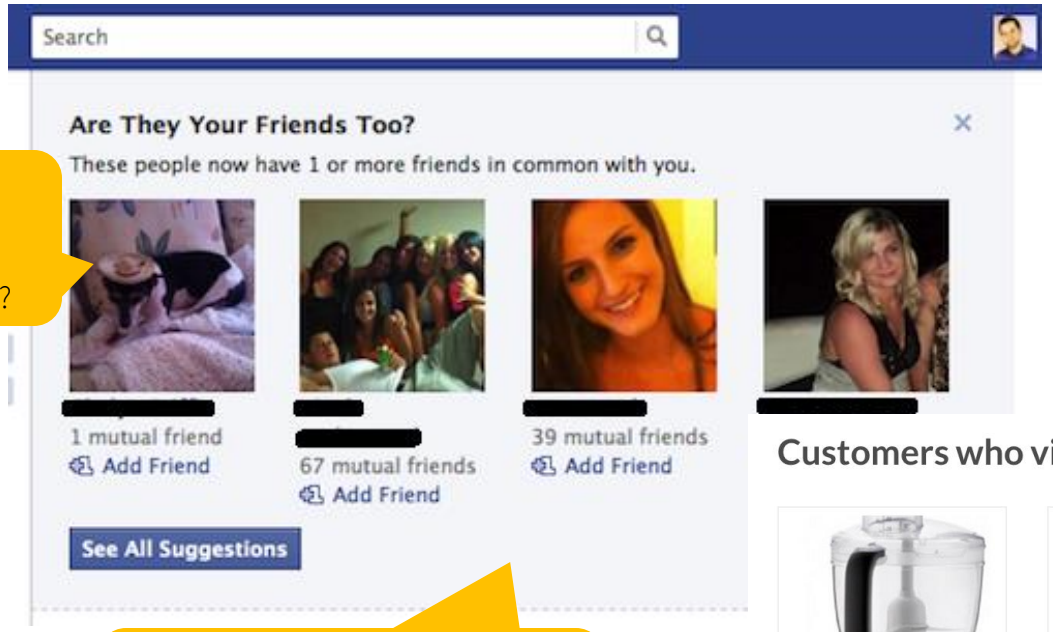
Link Prediction in Social Networks

Predicting links in social networks based on network characteristics

University of Zurich[UZH]

Stefan Bublitz, Luca Gaegauf, Pascal Sutter, Salome Lang

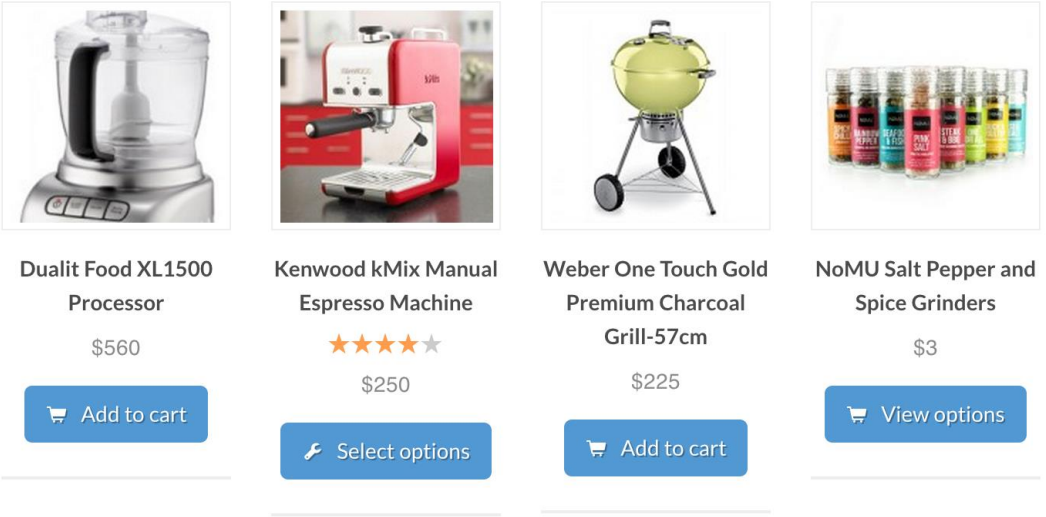
You have probably encountered link prediction before…



Is this dog your friend?

Link prediction is used in online social networks.

Link prediction is used by online retailers for product recommendations.



Link Prediction
Agenda

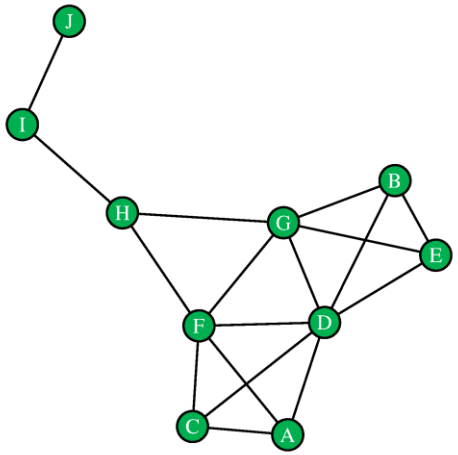
What do we want to do?

The goal:

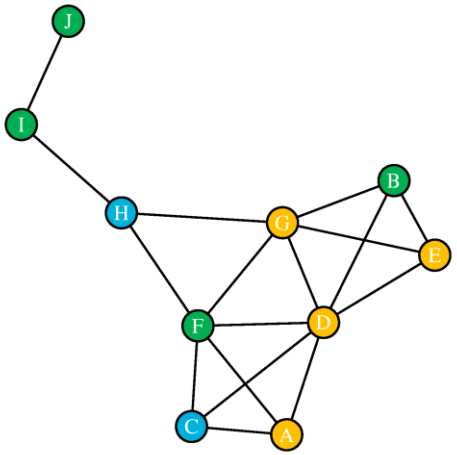
Take the snapshot of a given network and predict likely links between the existing nodes.

Based on:

the network itself, and…



… observed attributes of the nodes.



Link prediction is about predicting the most likely links

Definition

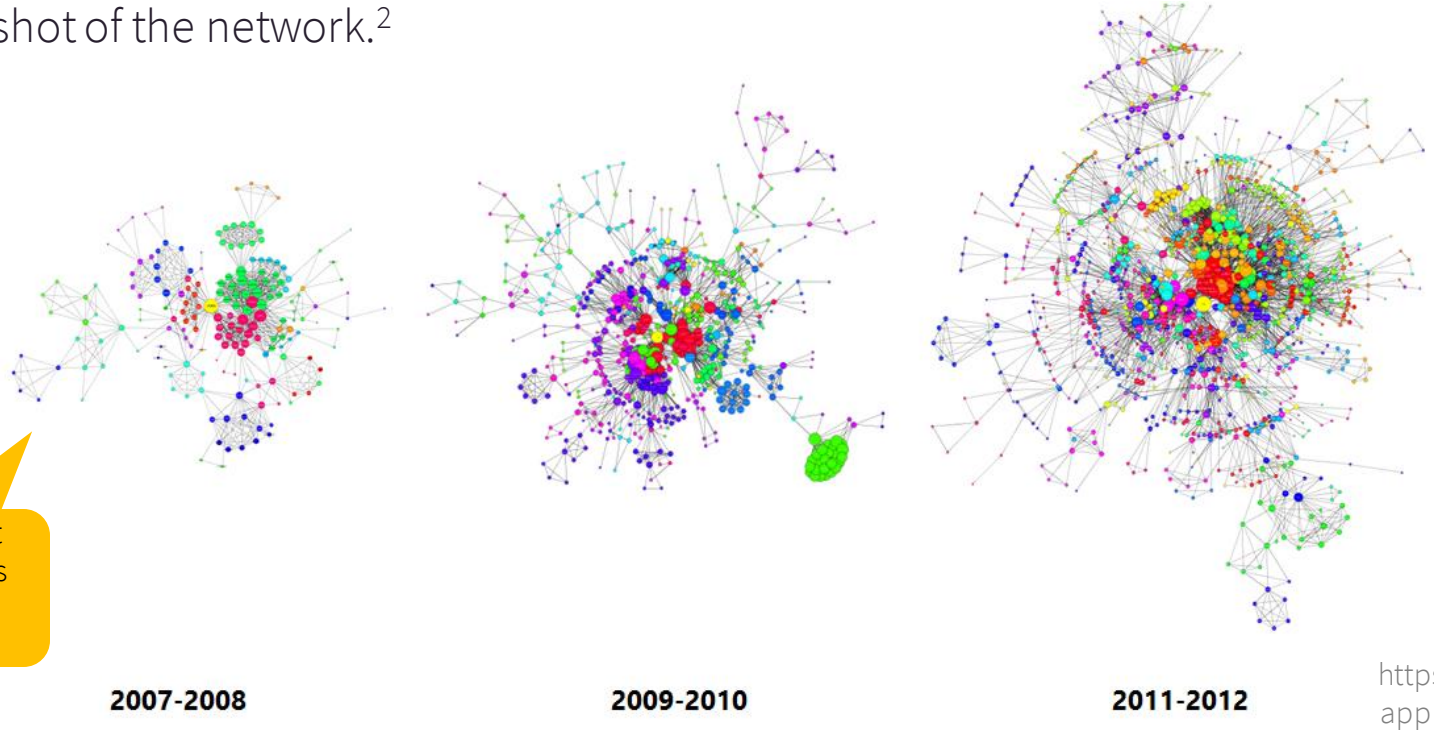
Link prediction estimates the likelihood of the existence of a link between two nodes, based on observed links and the attributes of nodes.[1,2]

Accordingly, the likelihood that another person in a social network is your friend can be estimated from

1. **Observed links in the network**: the number of links, clusters, cliques, etc.
2. **Attributes of the nodes**: common characteristics such as age, gender, interests, location, etc.

Link prediction methods can be used for two main purposes

1. to **detect unobserved links** that are missing or hidden in the observed network.
2. to **predict future links**, thus the future network evolution, based on the present snapshot of the network.[2]



The largest components in Apple's inventor network over a 6-year period

https://www.kenedict.com/apples-internal-innovation-network-unraveled-part-1-evolving-networks/

Link Prediction
Agenda

There are three major approaches to link prediction

The three approaches differ in the **level of analysis** from which they infer the probability of links between the existing nodes:

1. **Similarity-based methods** focus on the similarity of each pair of nodes in the network. This is the mainstream approach to link prediction.
2. **Maximum likelihood methods** focus on the network structure as a whole, assuming that networks are organized according to certain principles.
3. **Probabilistic models** focus on the unobserved underlying structure of an observed network. [1]

Many link prediction algorithms can be classified according to the three approaches

Approach	Algorithm
Similarity-based	Local similarity indices
	Global similarity indices
	Quasi-local indices
Maximum Likelihood	Hierarchical structure models
	Stochastic block models
Probabilistic	Probabilistic relational models
	Probabilistic entity relationship models
	Stochastic relational models

In the context of link prediction, methods are often called algorithms. We use the two terms interchangeably.

We will focus on the similarity-based approach and introduce you to the following algorithms

Approach	Algorithm	Presented here
Similarity-based	Local similarity indices	**Jaccard similarity**
	Global similarity indices	**Katz index**
	Quasi-local indices	**Local random walk and superposed random walk**
Maximum Likelihood	Hierarchical structure models	**Hierarchical structure models**
	Stochastic block models	
Probabilistic	Probabilistic relational models	
	Probabilistic entity relationship models	
	Stochastic relational models	

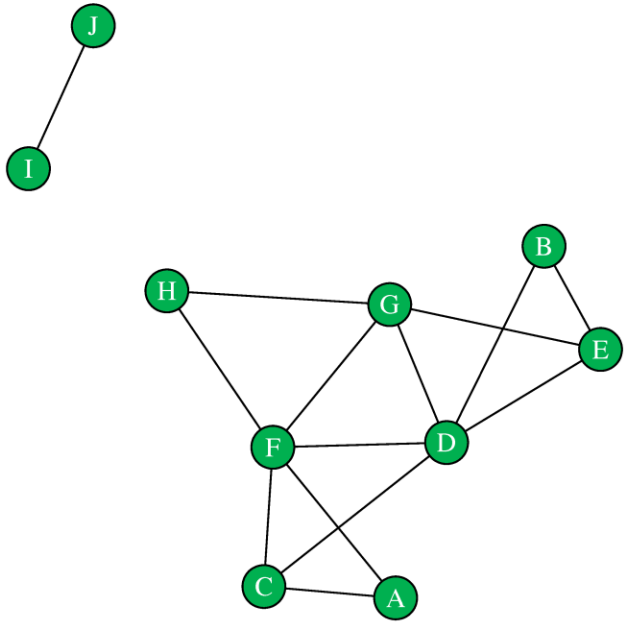
The different algorithms can only be evaluated with respect to the already existing links

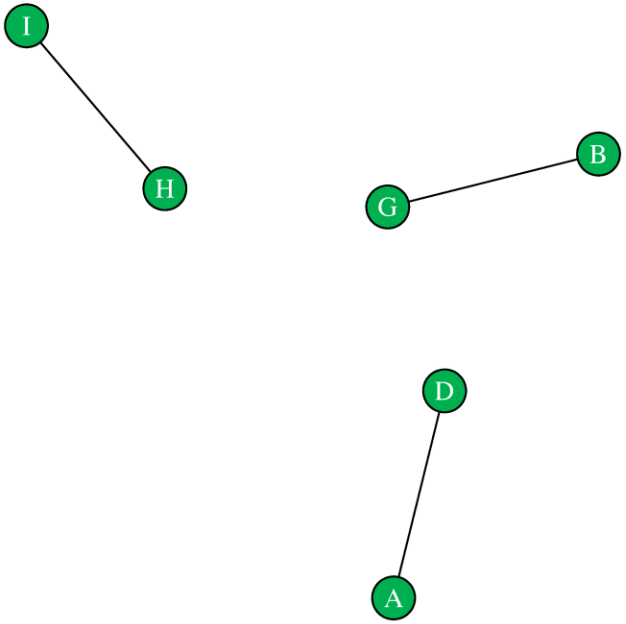
1. **Problem**: It is not possible to compare the accuracy of the algorithms in predicting missing or future links because these links are not available at the time of analysis.
2. Instead, we **cut** a part of the **edges of the observed network**, apply the algorithm to the new network and evaluate how well it predicts the edges we have cut.
3. This corresponds to splitting the edges into a **train and a test set**.

Split the edges of the kite network

Training data

Test data





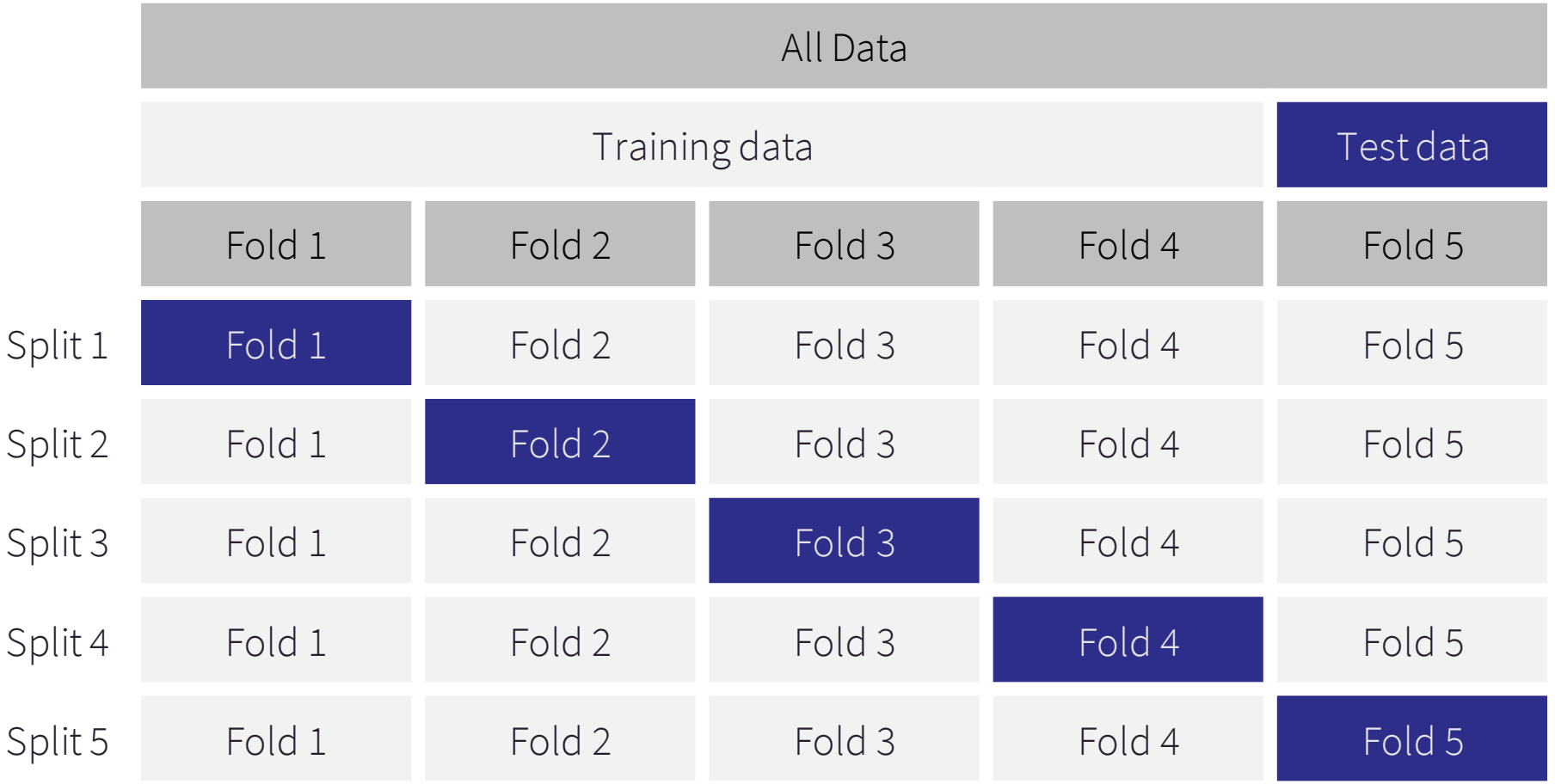
How should we split the edges then?

Partitioning the edges randomly is not the best solution

1. **Problem**: In repeated random sub-sampling, some edges may never enter the train set, whereas others may enter it several times. This results in **biased estimates**.
2. Particularly in a network context, it makes more sense to use **K-fold cross-validation**:
    1. Randomly split the edges in K subsets.
    2. Fit the algorithm to (K-1) subsets and evaluate how well it predicts the edges in the subset that is left.
    3. Repeat the last step K times.

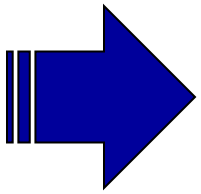
K-fold cross-validation has two advantages: **all links are used** for both training and testing, which results in **unbiased estimates**.

K-fold cross-validation can be illustrated as follows:



Similarity based approaches focus on the similarity of pairs of nodes

Similarity based algorithms are a class of algorithms where each **node pair** is assigned a score that describes their similarity to each other.



High similarity scores of node pairs with an unobserved link correspond to high likelihood of a link existing.

Similarity based approaches build on one important assumption

1. People tend to socialize with others that have **similar preferences**.
2. By the above assumption, people who are **friends in a social network** are likely to have similar preferences.
3. Unconnected friends of friends are also likely to have similar preferences since they share a friend with similar preferences.
4. Therefore, the **network structure reveals information about the nodes**, e.g. similar preferences that are not observable for an outsider.

First, we will present the different similarity scores and then how to use the scores for link prediction.

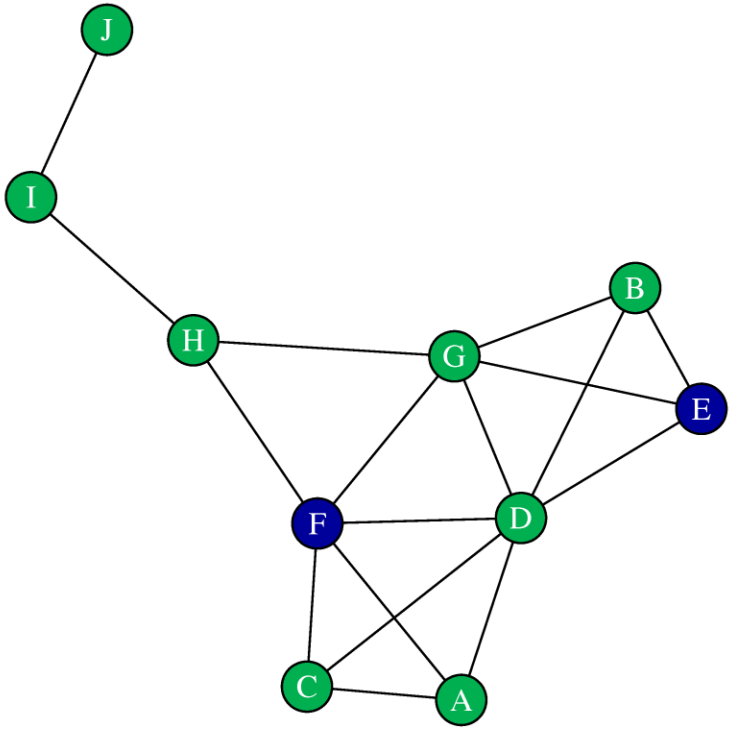
Jaccard similarity is a local measure

The **Jaccard similarity** of two nodes counts the number of common neighbours and normalizes by the number of nodes that are neighbours of at least one of the two nodes.

$$Jaccard_{ij} = \frac{\sum_{k=1}^{n} A_{ij} A_{kj}}{\sum_{k=1}^{n} A_{ik} + \sum_{k=1}^{n} A_{jk} - \sum_{k=1}^{n} A_{ik} A_{kj}}$$

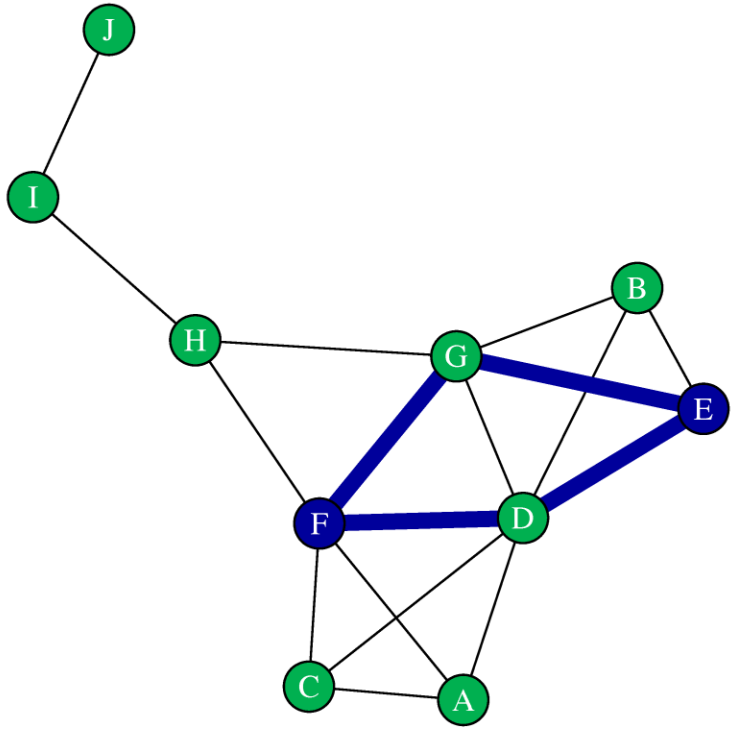
Nodes with high similarity share a large proportion of the neighbours.

To calculate the Jaccard similarity of nodes E and F focus on their direct neighbours



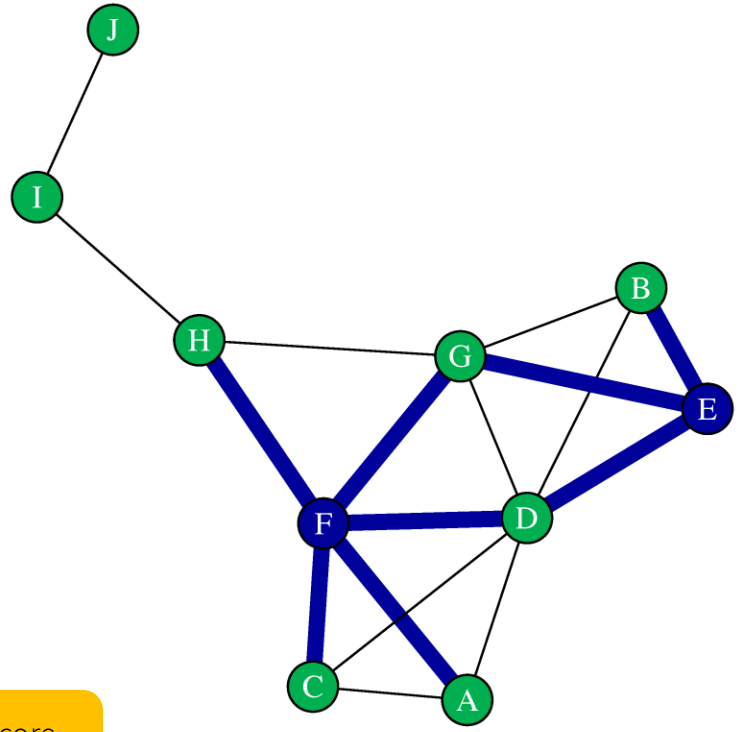
$$Jaccard_{EF} = \frac{\sum_{k=1}^{n} A_{EF} A_{kF}}{\sum_{k=1}^{n} A_{Ek} + \sum_{k=1}^{n} A_{Fk} - \sum_{k=1}^{n} A_{Ek} A_{kF}}$$

Nodes E and F share 2 neighbours: D and G



$$Jaccard_{EF} = \frac{2}{\sum_{k=1}^{n} A_{Ek} + \sum_{k=1}^{n} A_{Fk} - \sum_{k=1}^{n} A_{Ek} A_{kF}}$$

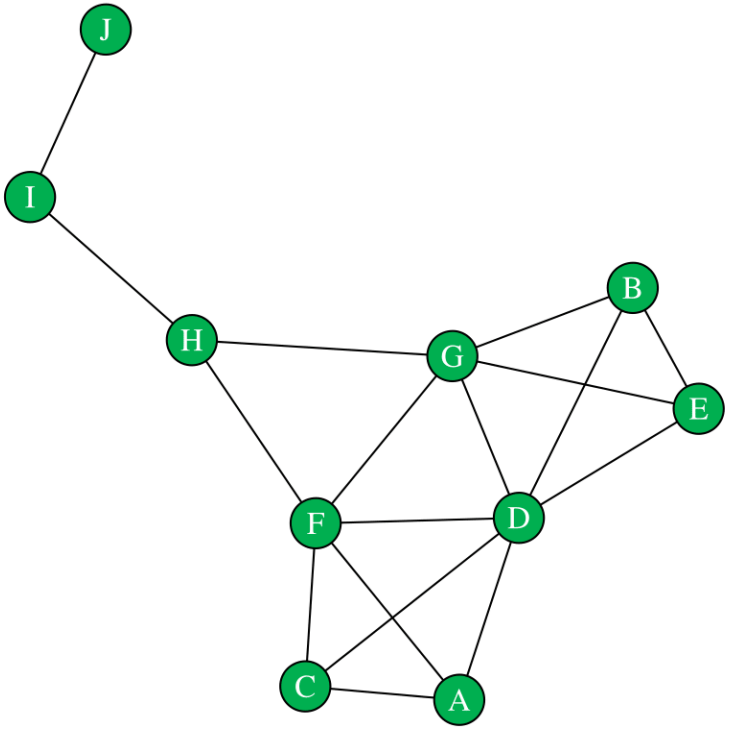
Nodes E and F have a total of 6 neighbours between themselves: A, B, C, D, G, and H



The Jaccard score of **E**, and **F** is 1/3.

$$Jaccard_{EF} = \frac{2}{6}$$

The Jaccard similarity is subsequently calculated for all node pairs



How to use these scores to predict links will be explained after all models have been presented

Katz similarity is a global measure

The **Katz similarity** of two nodes aggregates the amount of *n*-length paths between them.

Longer paths are exponentially damped to give the shorter paths more weight.

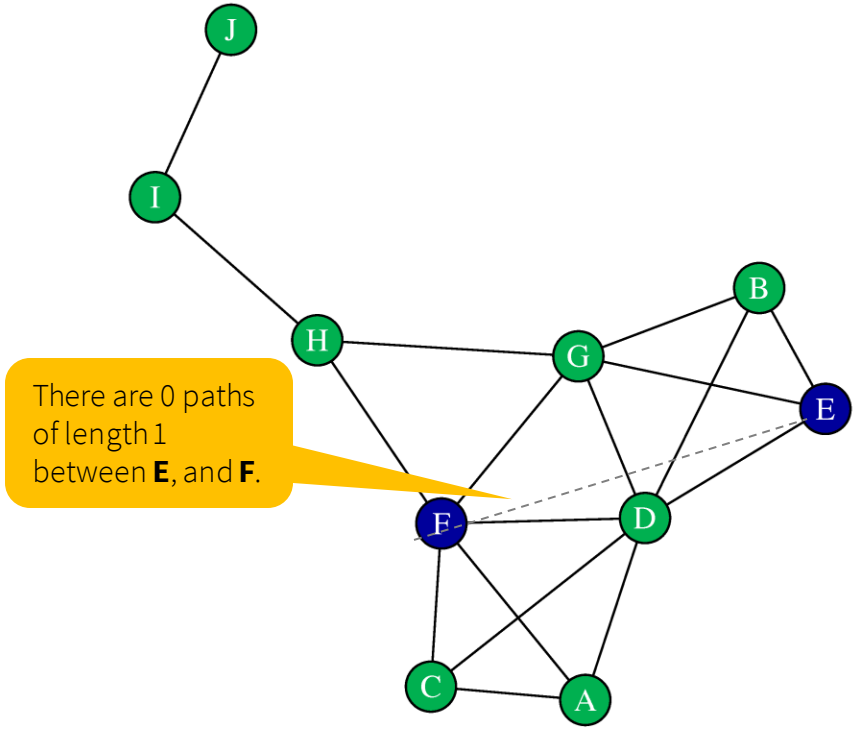
$$Katz_{ij} = \sum_{l=1}^{\infty} \beta^n \cdot \left| paths_{ij}^{\langle n \rangle} \right| = \beta A_{ij} + \beta^2 A^2{}_{ij} + \beta^3 A_{ij}{}^3 + \ldots$$

The **damping factor** specifies how much to dampen the effect of long paths.

$\mathbf{A^n_{ij}}$ is number of *n*-step paths between *i*, and *j*.

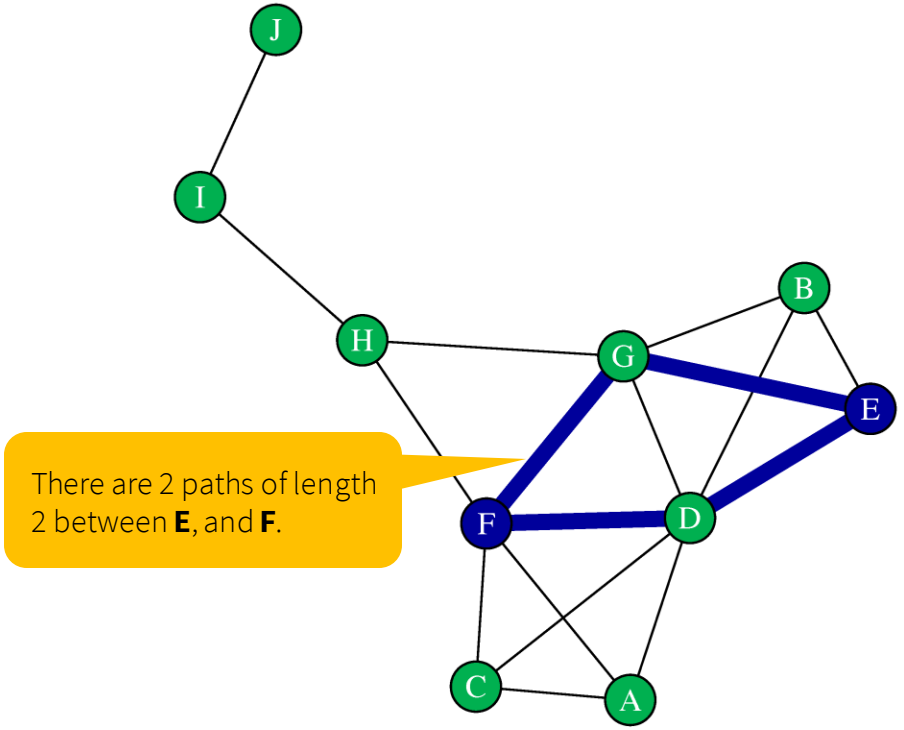
Nodes with high Katz scores have many shorter paths connecting them than nodes with low scores.

To calculate the Katz similarity of nodes E and F, first consider the direct path between E and F



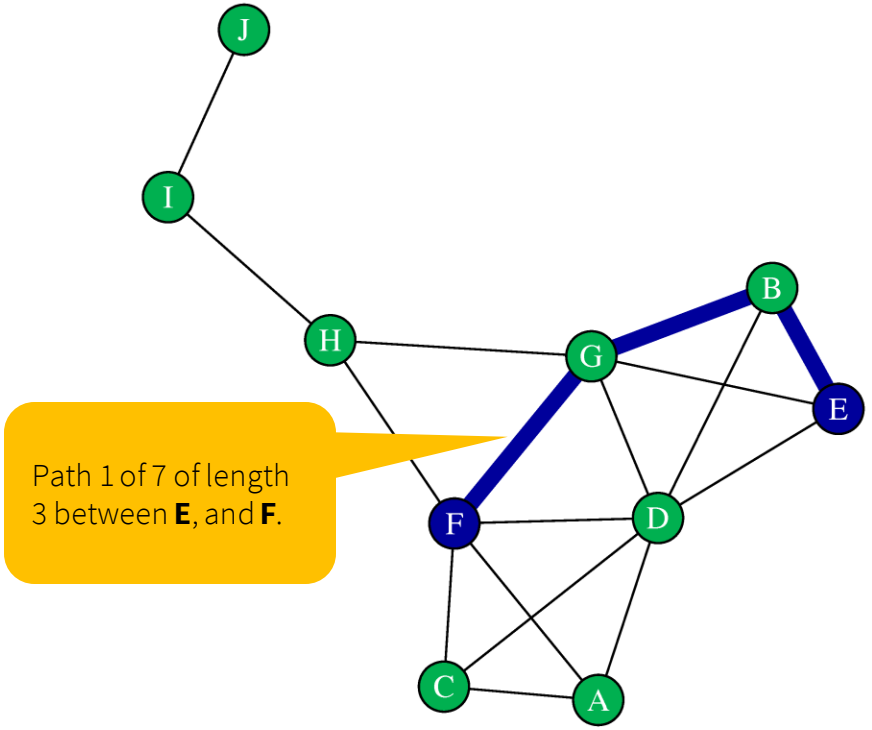
$$Katz_{EF} = \beta \cdot 0 + \beta^2 A^2{}_{EF} + \beta^3 A_{EF}{}^3 + \ldots$$

There are 2 paths over 1st degree neighbours



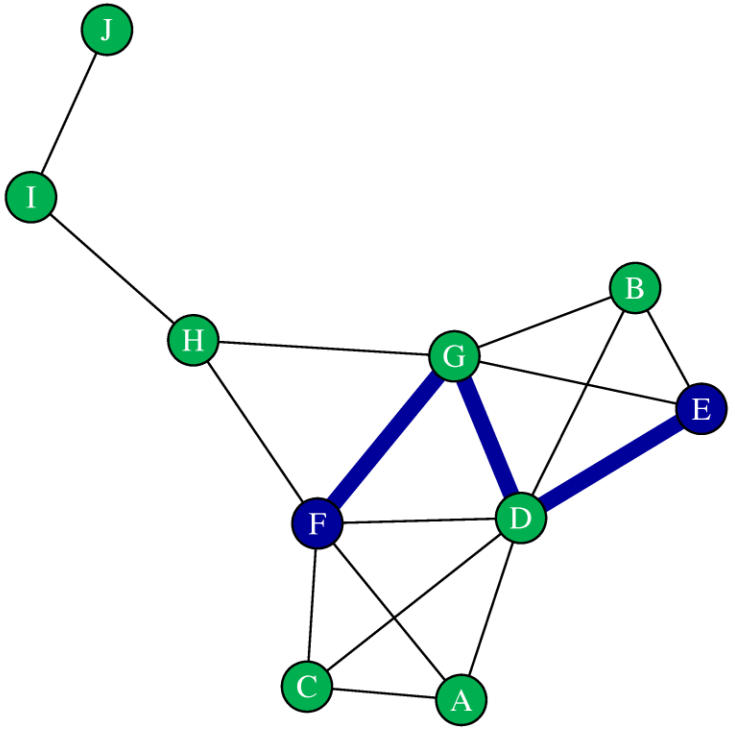
$$Katz_{EF} = \beta \cdot 0 + \beta^2 \cdot 2 + \beta^3 A_{EF}{}^3 + \ldots$$

There are 7 paths over 2nd degree neighbours
Path 1



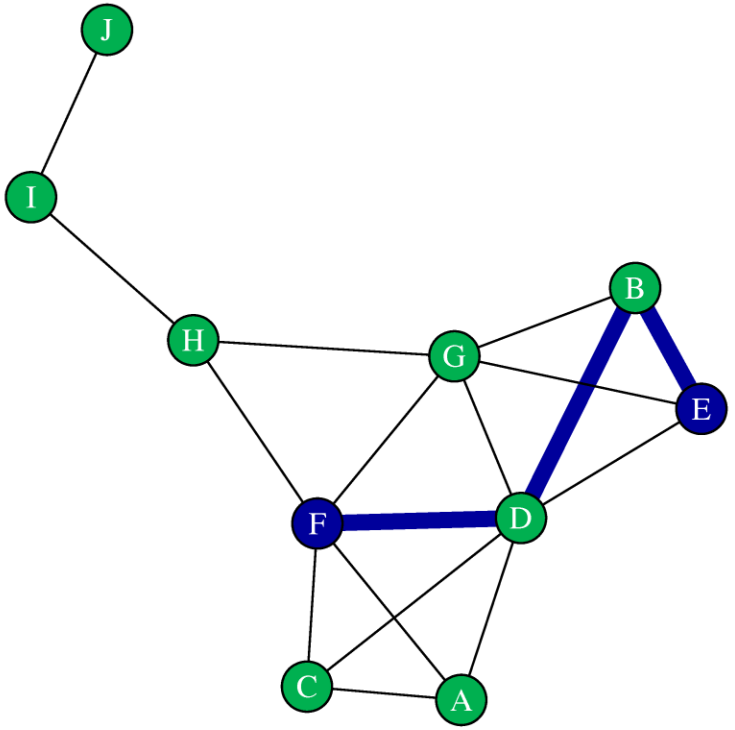
$$Katz_{EF} = \beta \cdot 0 + \beta^2 \cdot 2 + \beta^3 \cdot (1) + \ldots$$

There are 7 paths over 2nd degree neighbours
Path 2



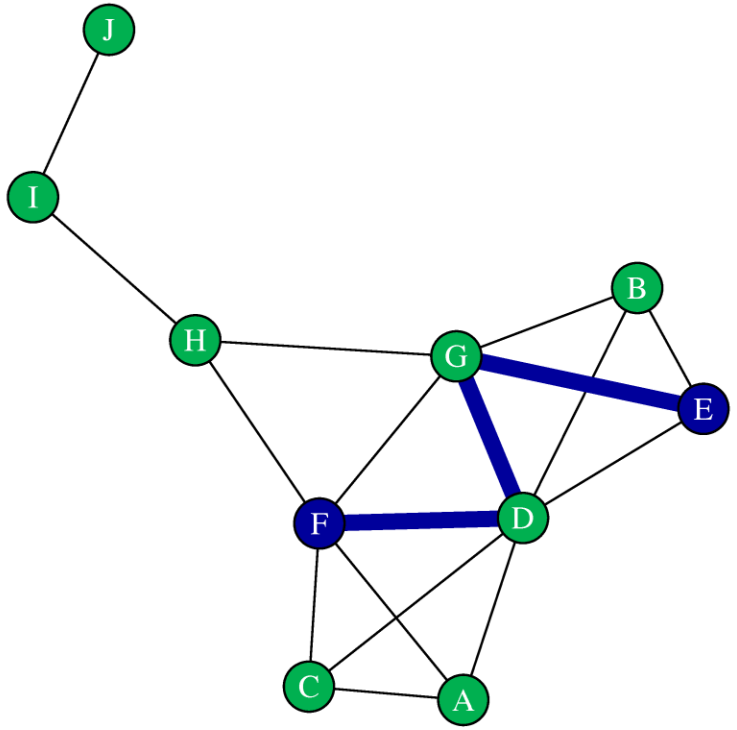
$$Katz_{EF} = \beta \cdot 0 + \beta^2 \cdot 2 + \beta^3 \cdot (1+1) + \ldots$$

There are 7 paths over 2nd degree neighbours
Path 3



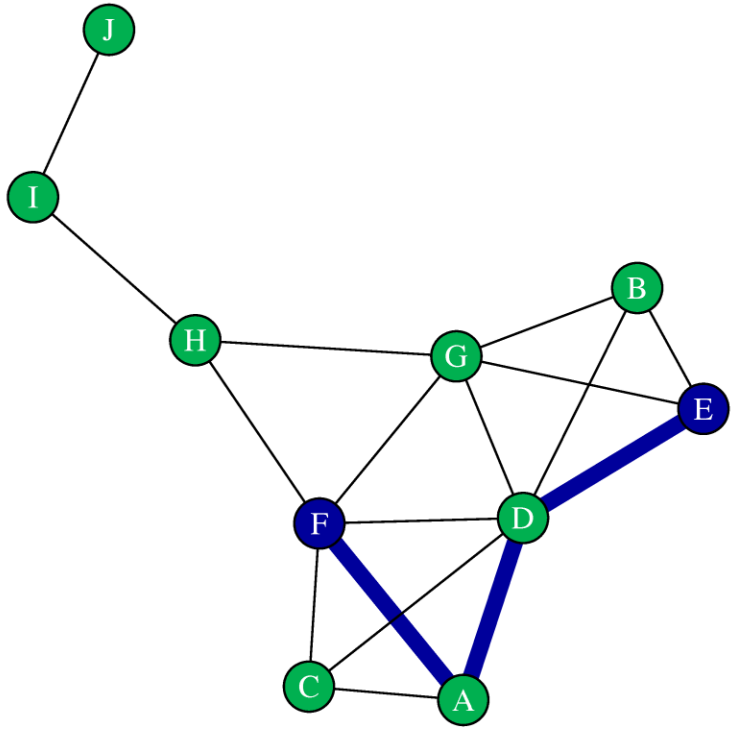
$$Katz_{EF} = \beta \cdot 0 + \beta^2 \cdot 2 + \beta^3 \cdot (1+1+1) + \ldots$$

There are 7 paths over 2nd degree neighbours
Path 4



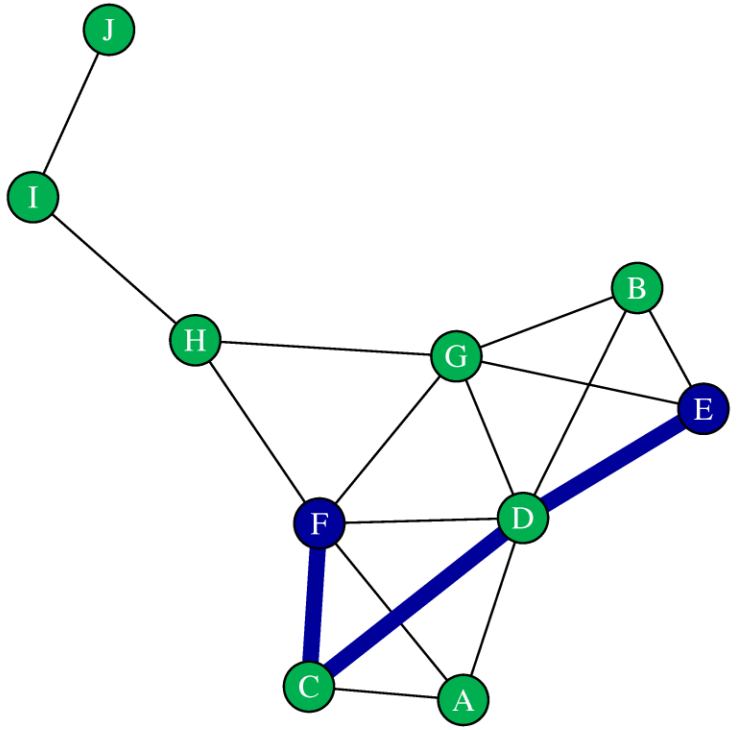
$$Katz_{EF} = \beta \cdot 0 + \beta^2 \cdot 2 + \beta^3 \cdot (1+1+1+1) + \ldots$$

There are 7 paths over 2nd degree neighbours
Path 5



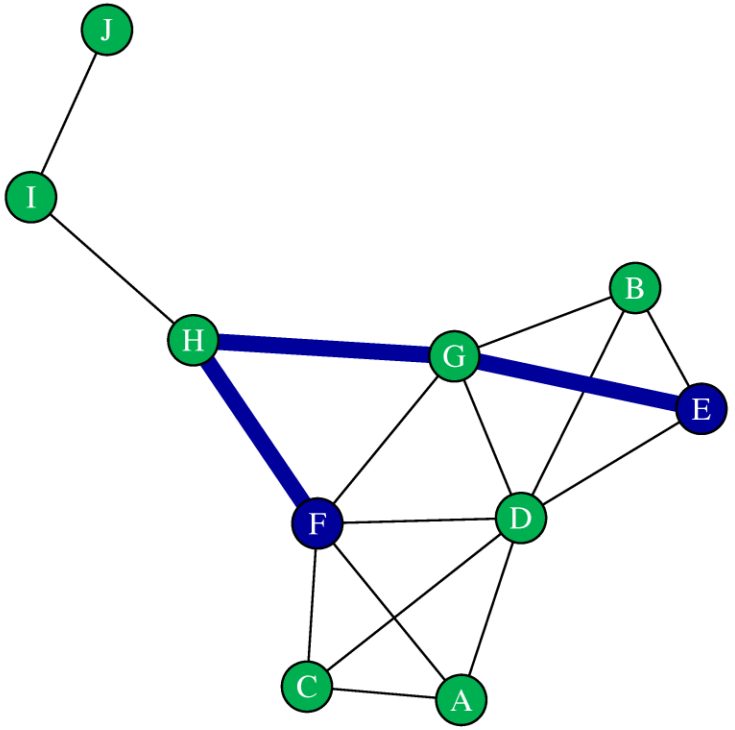
$$Katz_{EF} = \beta \cdot 0 + \beta^2 \cdot 2 + \beta^3 \cdot (1+1+1+1+1) + \ldots$$

There are 7 paths over 2nd degree neighbours
Path 6



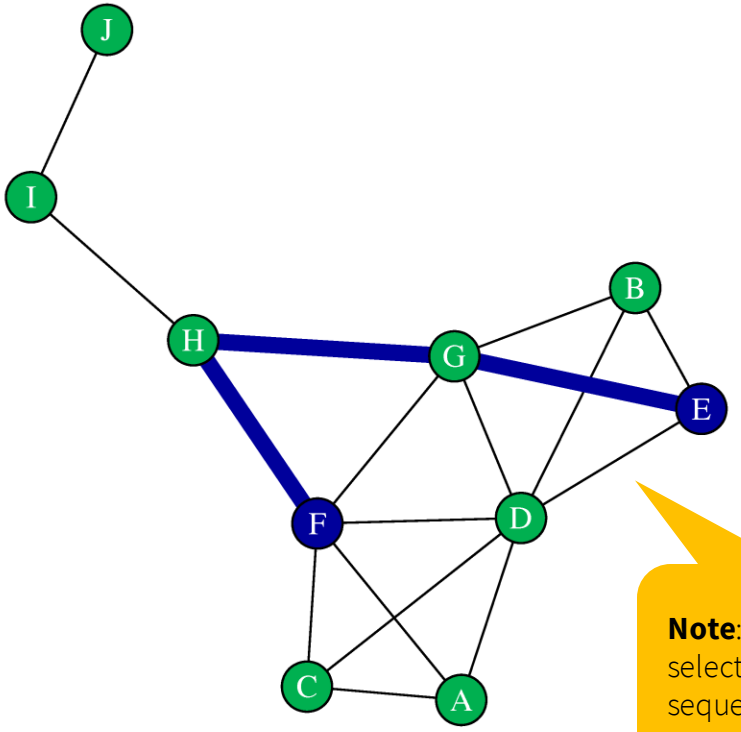
$$Katz_{EF} = \beta \cdot 0 + \beta^2 \cdot 2 + \beta^3 \cdot (1+1+1+1+1+1) + \ldots$$

There are 7 paths over 2nd degree neighbours
Path 7



$$Katz_{EF} = \beta \cdot 0 + \beta^2 \cdot 2 + \beta^3 \cdot (1+1+1+1+1+1+1) + \ldots$$

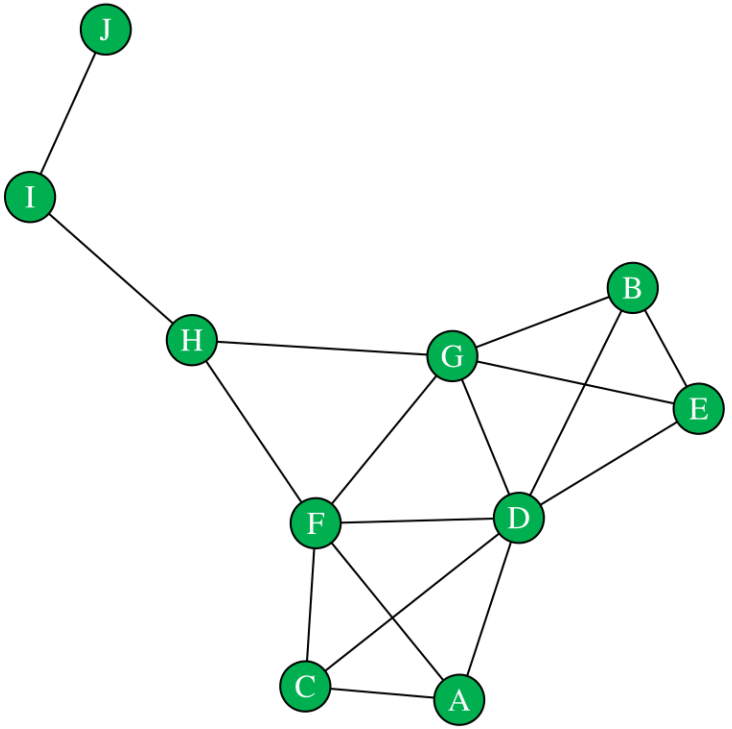
This is continued to infinity



Note: It is important to select a β such that the sequence converges.

$$Katz_{EF} = \beta \cdot 0 + \beta^2 \cdot 2 + \beta^3 \cdot 7 + \ldots$$

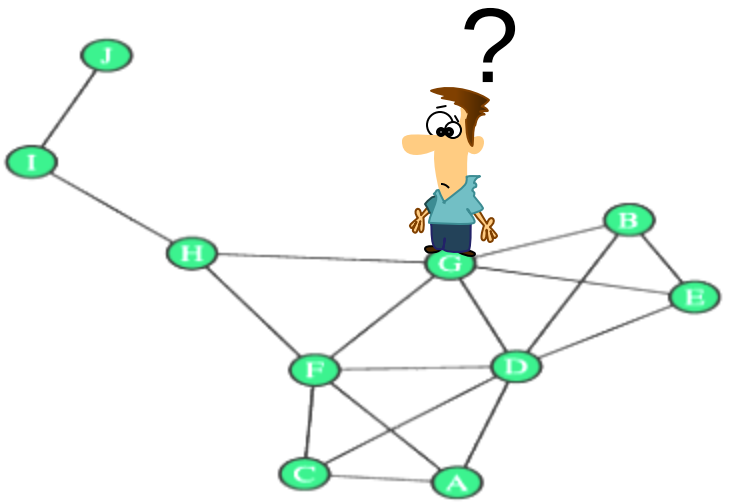
The Katz similarity is subsequently calculated for all node pairs



How to use these scores to predict links will be explained after all models have been presented.

Random walk scores are quasi-local measures

1. Imagine a being who walks along the paths of a network. The walker's memory is long enough to get him from point A, to point B along a path, however, once the walker arrives, he forgets where he came from, and where he was going. The next path he chooses to walk down is **completely independent** of where he just came from.
2. We call this being a **random walker.**
3. We apply 2 random walk measures:
   - Local random walk
   - Superposed random walk



Local random walk indicates the similarity of pairs of nodes as well

The **local random walk** score of two nodes is the sum of the probability that a random walker placed at node *i* will reach node *j* in *n* steps, and the probability that a random walker placed at node *j* will reach *i* in *n* steps, weighted by the starting node's portion of degree.

n is determined exogenously.

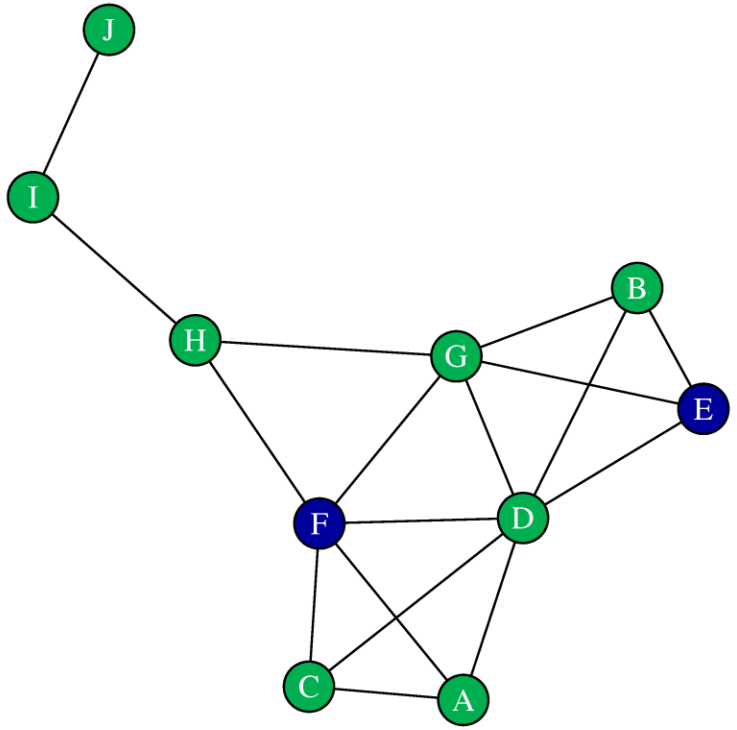
$$LRW^{n}{}_{ij} = q_i \pi^{\langle n \rangle}{}_{ij} + q_j \pi^{\langle n \rangle}{}_{ji}, \quad \text{where} \quad q_i = \frac{k_i}{2|E|}$$

k_i is the degree of *i*.

E is the total number of edges. **2|E|** is the sum of all degree centralities.

Nodes in a similar community are more likely to be reached by a random walker.

Calculating local random walk between nodes E and F for n = 3

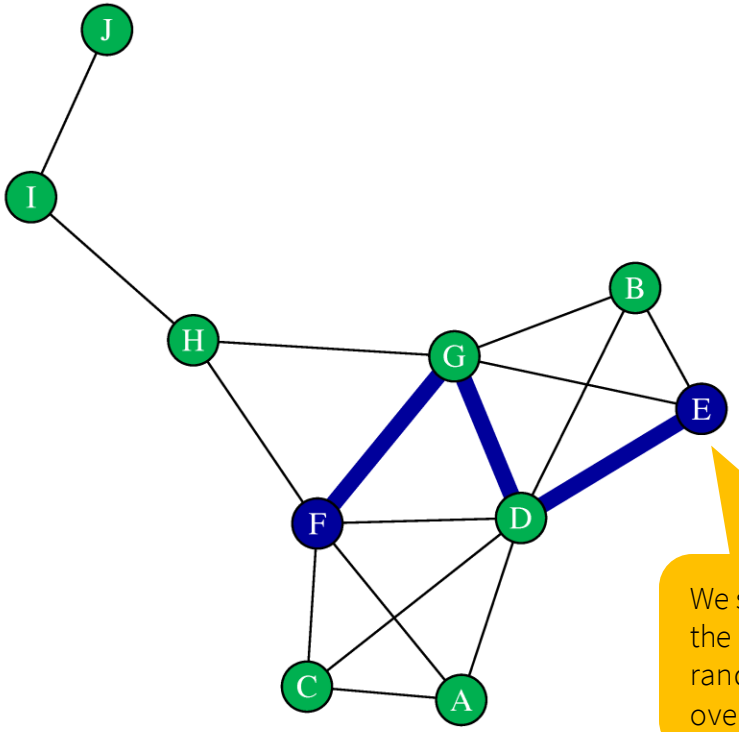


We will calculate the probability that a random walker placed at **E**, will reach **F** in 3 steps.

$$LRW^{3}_{EF} = \frac{k_E}{2|E|}\pi^{\langle 3\rangle}_{EF} + \frac{k_F}{2|E|}\pi^{\langle 3\rangle}_{FE}$$

Choose a 3-step path between E and F

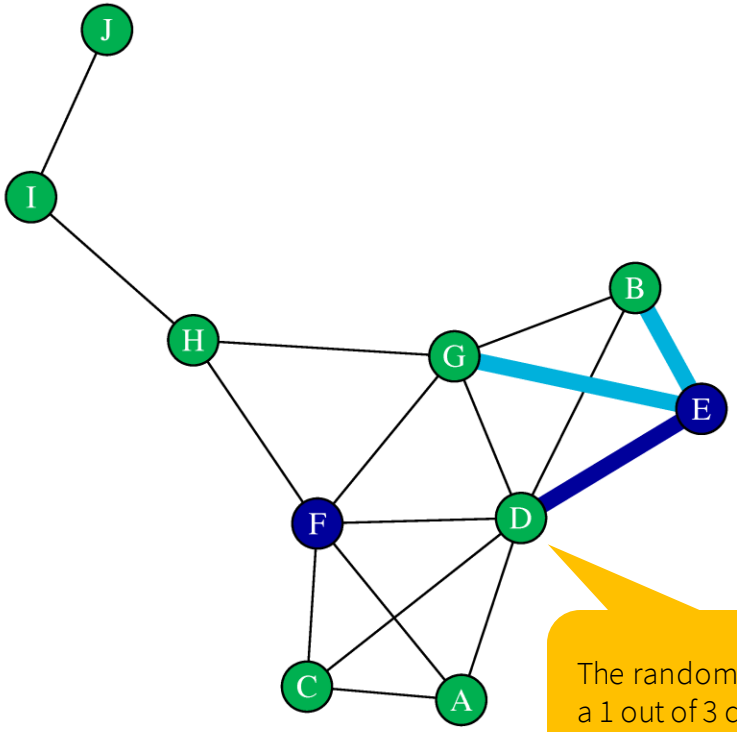
E-D-G-F



We start with calculating the probability that a random walker will reach **F** over the path **E**-**D**-**G**-**F**.

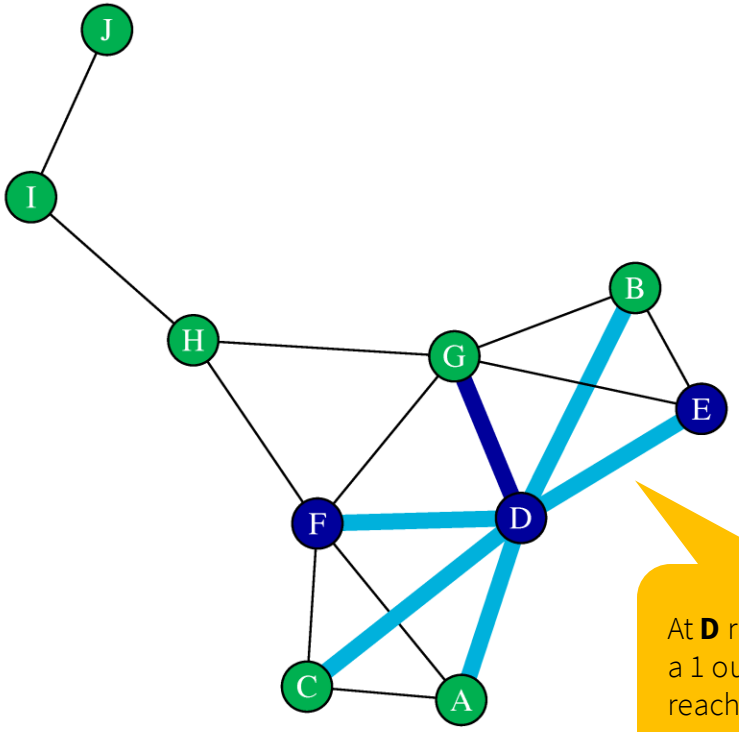
$$LRW^{3}{}_{EF} = \frac{k_E}{2|E|}\pi^{\langle 3\rangle}{}_{EF} + \frac{k_F}{2|E|}\pi^{\langle 3\rangle}{}_{FE}$$

Walking along E-D-G-F
Random walker starts at E



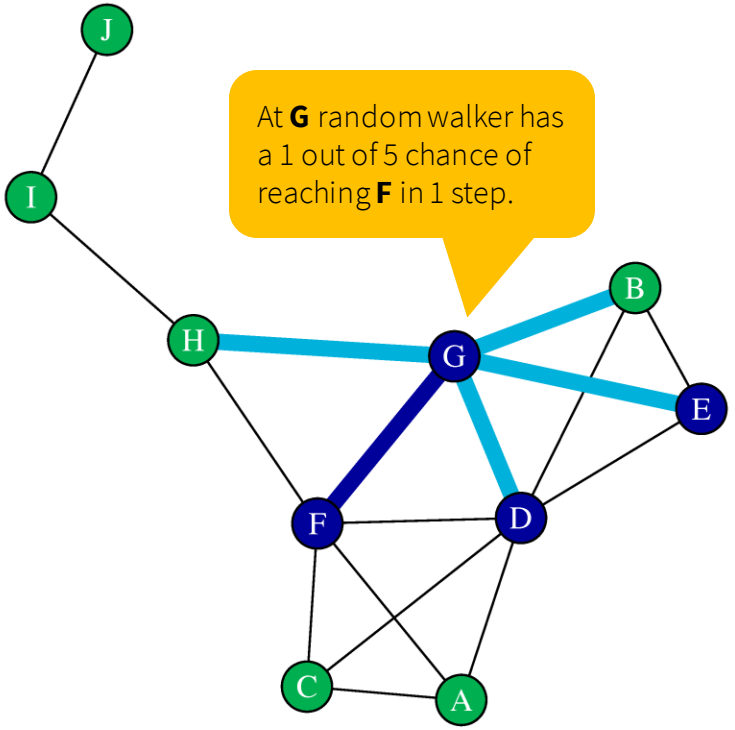
$$LRW^{3}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{3}\right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}_{FE}$$

Walking along E-D-G-F
From D to E



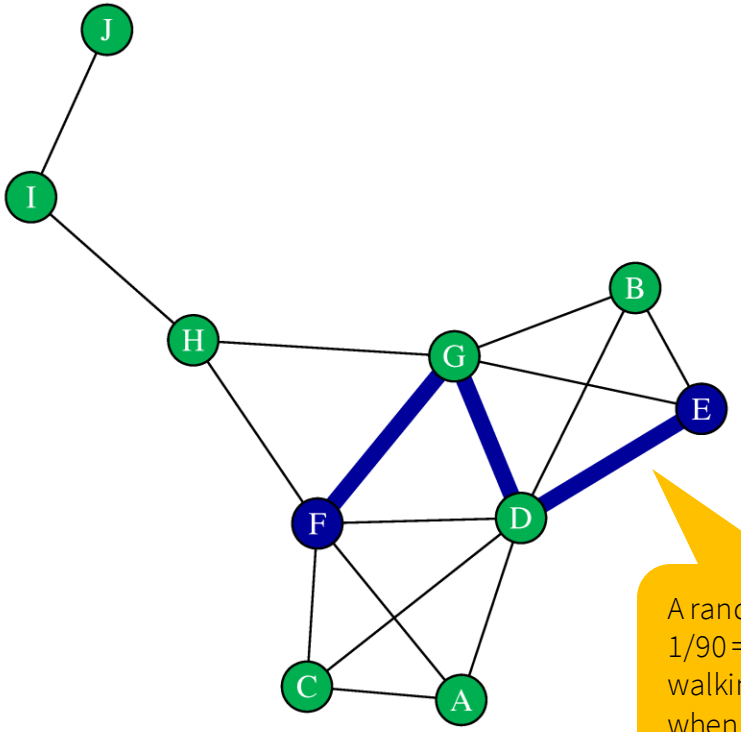
$$LRW^{3}{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{3} \cdot \frac{1}{6} \right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}{}_{FE}$$

Walking along E-D-G-F
From G to F



$$LRW^{3}{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{3} \cdot \frac{1}{6} \cdot \frac{1}{5} \right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}{}_{FE}$$

The probability of walking along E-D-G-F

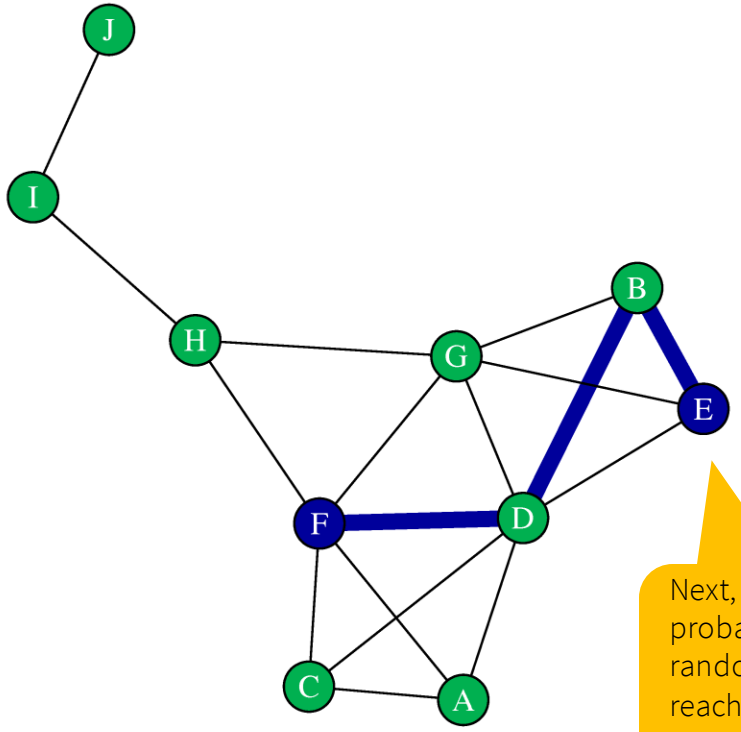


A random walker has a 1/90 = 1.11% chance of walking along **E**-**D**-**G**-**F**, when starting at **E**.

$$LRW^{3}{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90}\right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}{}_{FE}$$

Choose another 3-step path between E and F

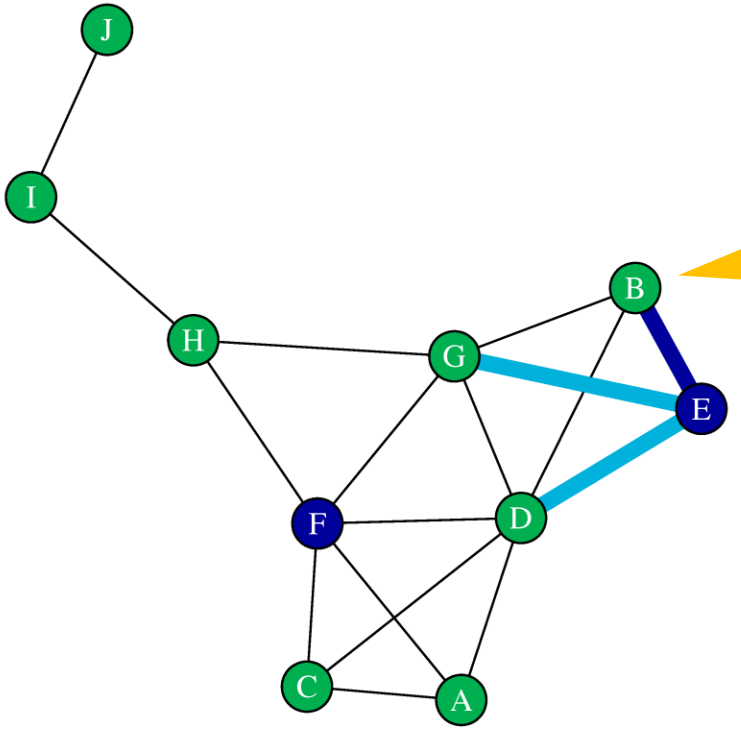
E-B-D-F



Next, we calculate the probability that a random walker will reach **F** over the path **E**-**B**-**D**-**F**.

$$LRW^{3}{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90}\right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}{}_{FE}$$

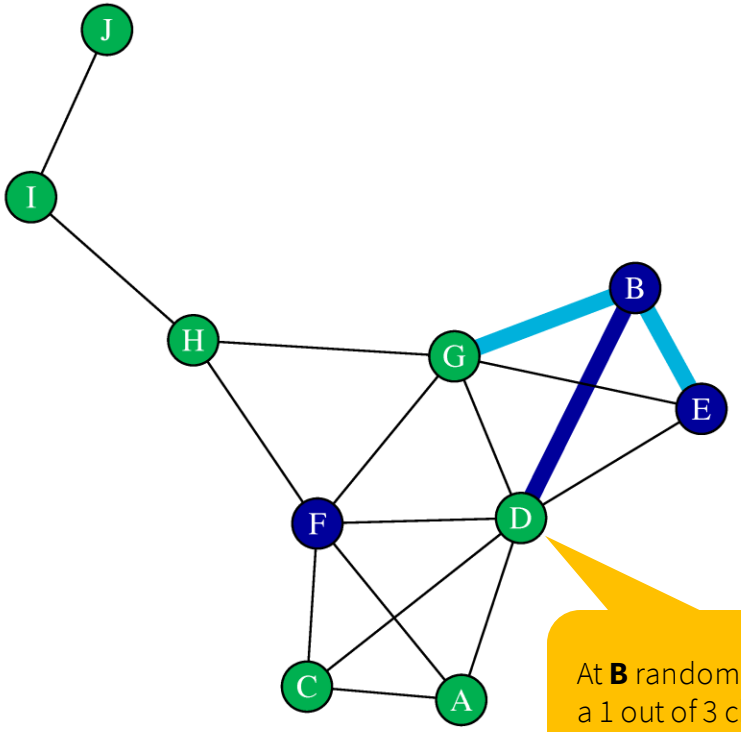
Walking along E-B-D-F
Random walker starts at E



The random walker has a 1 out of 3 chance of reaching **B** in 1 step.

$$LRW^{3}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{3} \right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}_{FE}$$

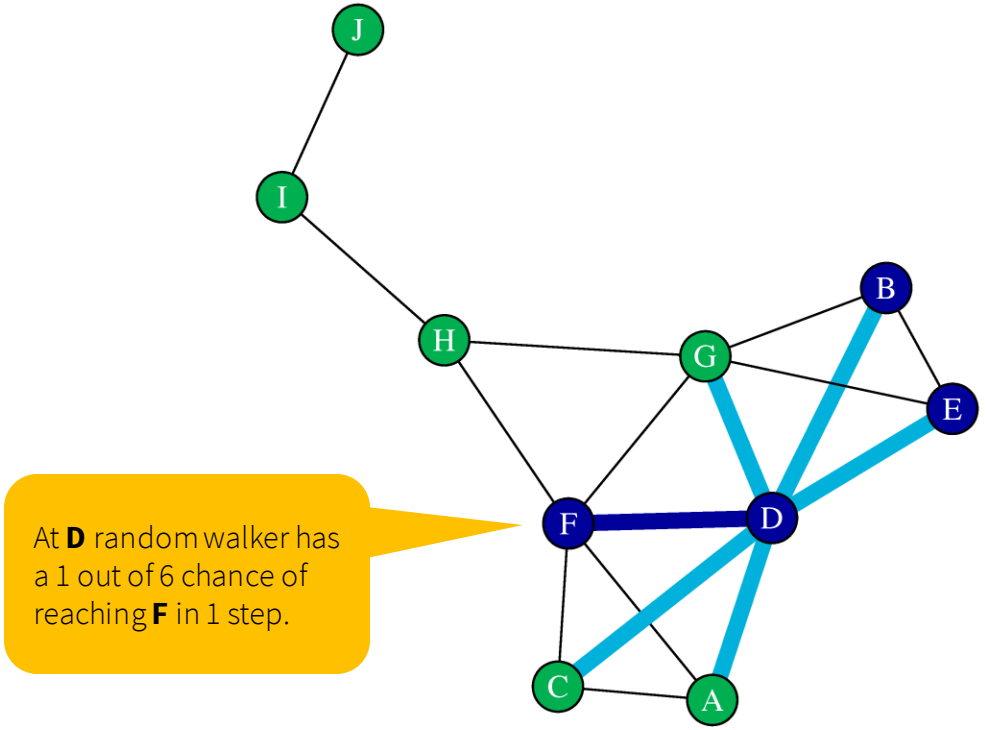
Walking along E-B-D-F
From B to D



At **B** random walker has a 1 out of 3 chance of reaching **D** in 1 step.

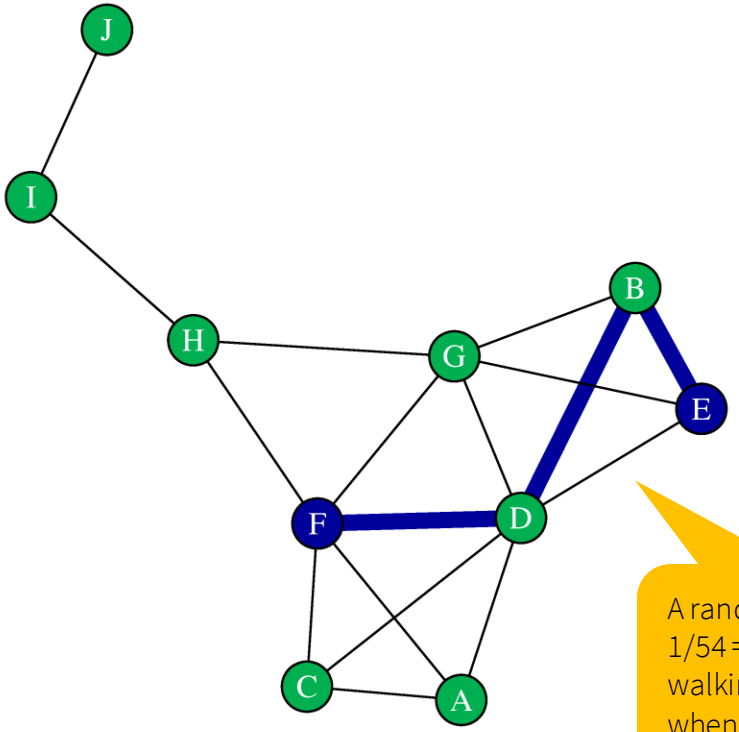
$$LRW^{3}{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{3} \cdot \frac{1}{3} \right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}{}_{FE}$$

Walking along E-B-D-F
From D to F



$$LRW^{3}{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{6} \right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}{}_{FE}$$

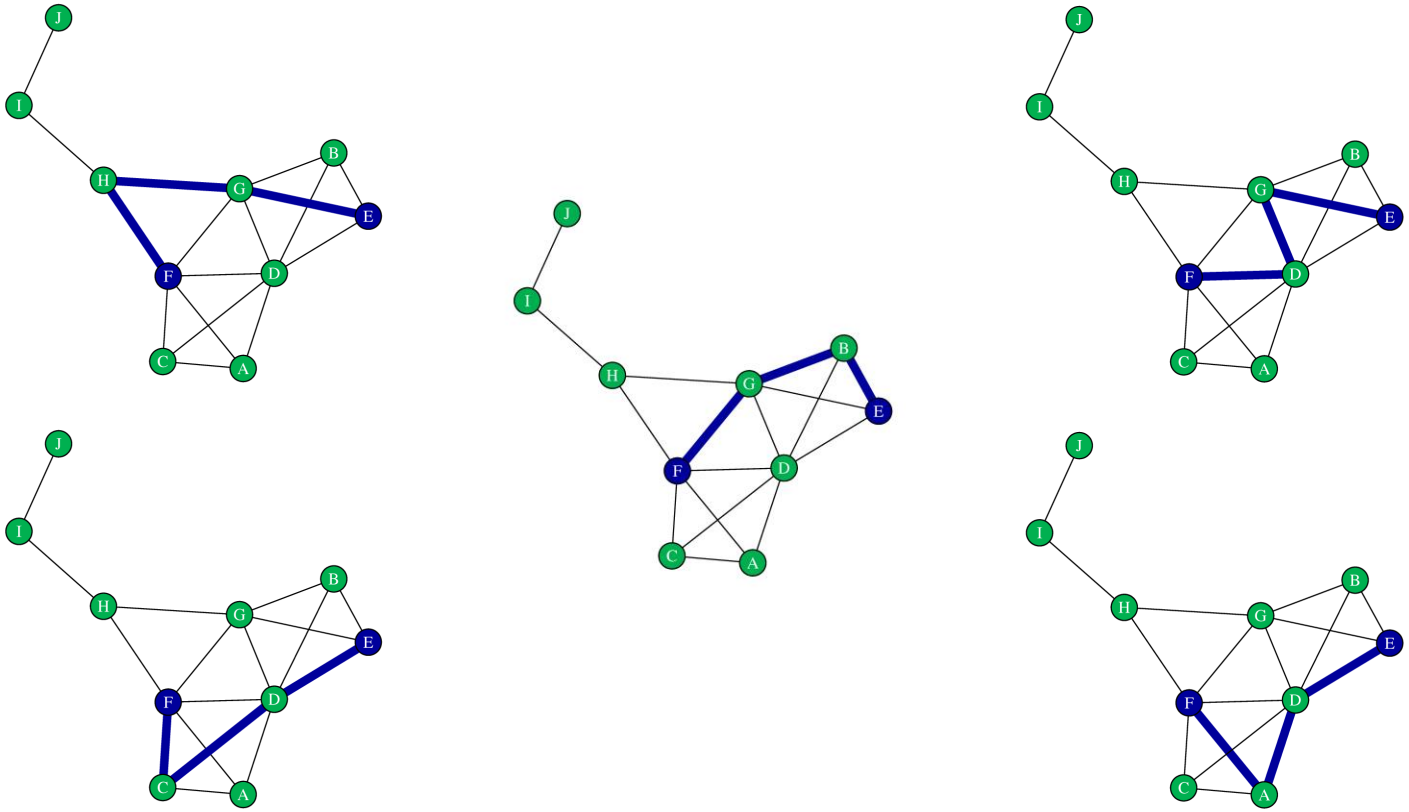
The probability of walking along E-B-D-F



A random walker has a 1/54 = 1.85% chance of walking along **E**-**B**-**D**-**F**, when starting at **E**.

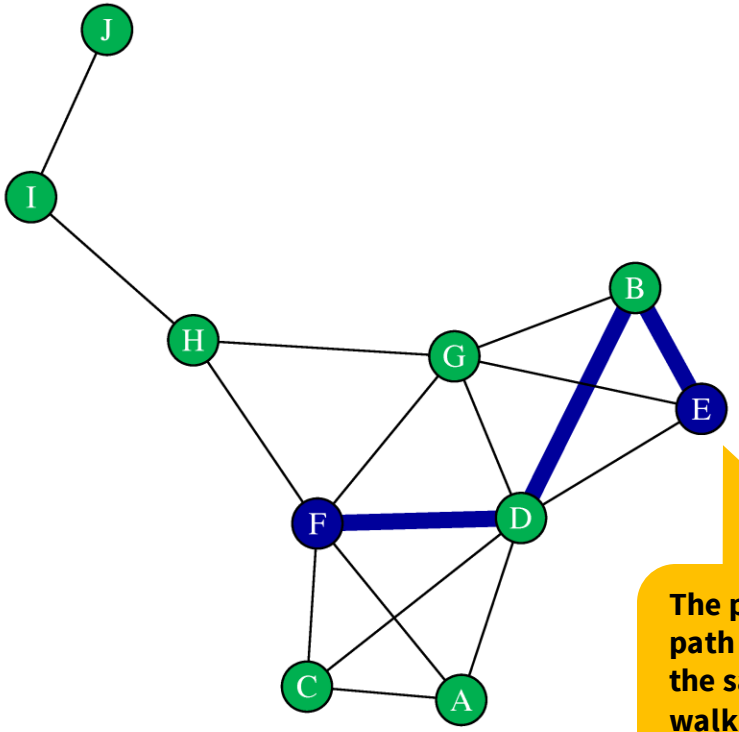
$$LRW^{3}{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{54} \right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}{}_{FE}$$

This is subsequently done for all 3-step paths connecting E and F



$$LRW^{3}{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{54} + \ldots \right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}{}_{FE}$$

The same thing is done for all 3-step paths connecting F and E

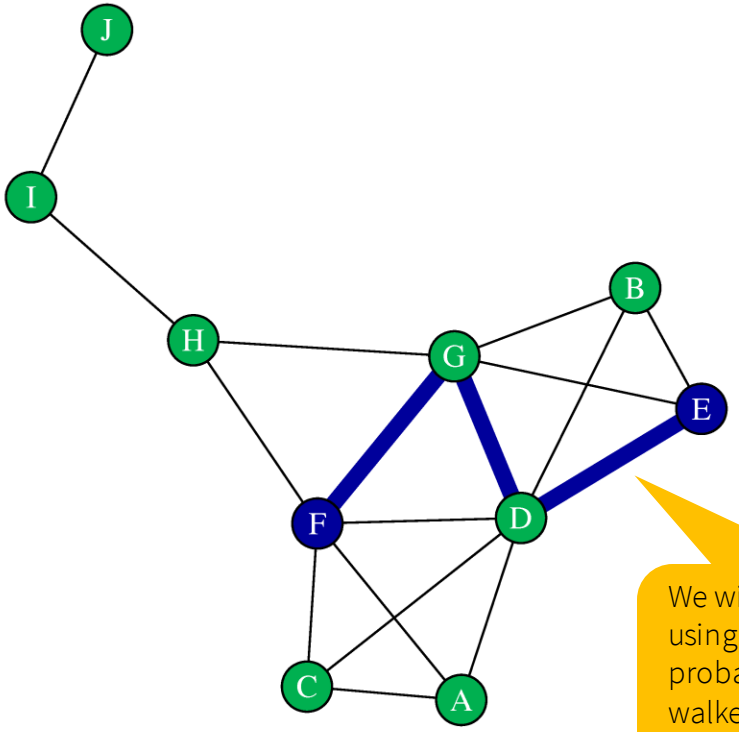


The probability of walking a path connecting E to F, is not the same as the probability of walking the same path in reverse!

$$LRW^{3}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{54} + \ldots \right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}_{FE}$$

Choose the a 3-step path between F and E

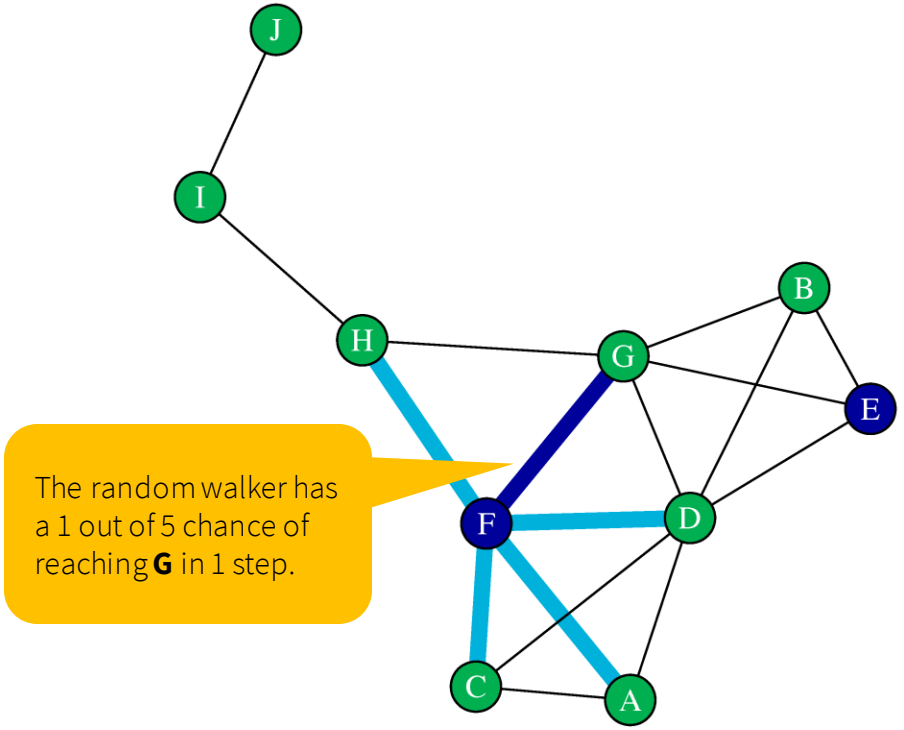
F-G-D-E



We will demonstrate this using by calculating the probability that a random walker placed at **F**, will reach **E** in 3 step, over **F**-**G**-**D**-**E**.

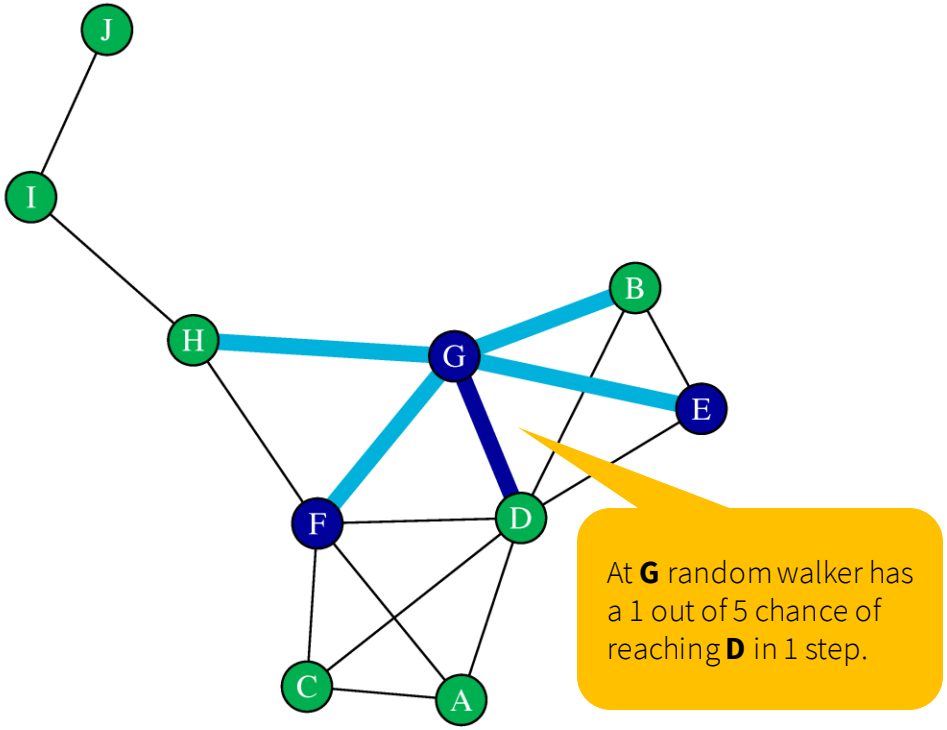
$$LRW^{3}{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{54} + \ldots \right) + \frac{k_F}{2|E|} \pi^{\langle 3 \rangle}{}_{FE}$$

Walking along F-G-D-E
Random walker starts at F



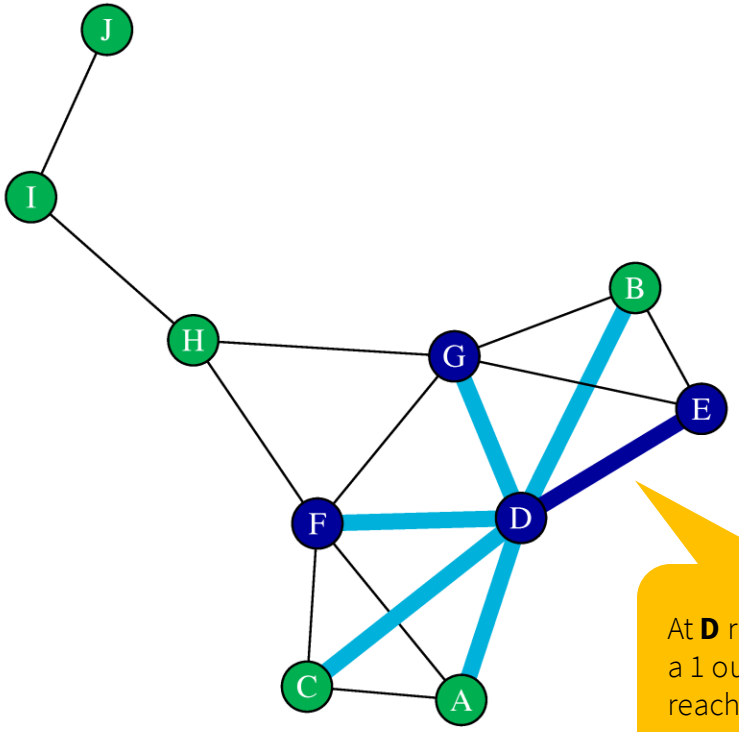
$$LRW^3{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{54} + \ldots \right) + \frac{k_F}{2|E|} \left(\frac{1}{5} \right)$$

Walking along F-G-D-E
From G to D



$$LRW^3{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{54} + \ldots \right) + \frac{k_F}{2|E|} \left(\frac{1}{5} \cdot \frac{1}{5} \right)$$

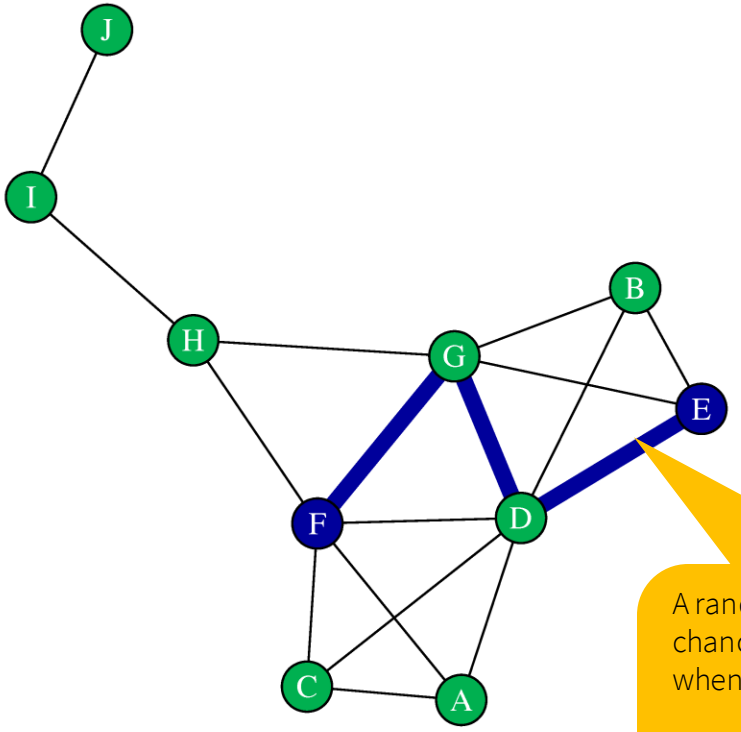
Walking along F-G-D-E
From D to E



At **D** random walker has a 1 out of 6 chance of reaching **E** in 1 step.

$$LRW^3{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{54} + \ldots \right) + \frac{k_F}{2|E|} \left(\frac{1}{5} \cdot \frac{1}{5} \cdot \frac{1}{6} \right)$$

The probability of walking along F-G-D-E

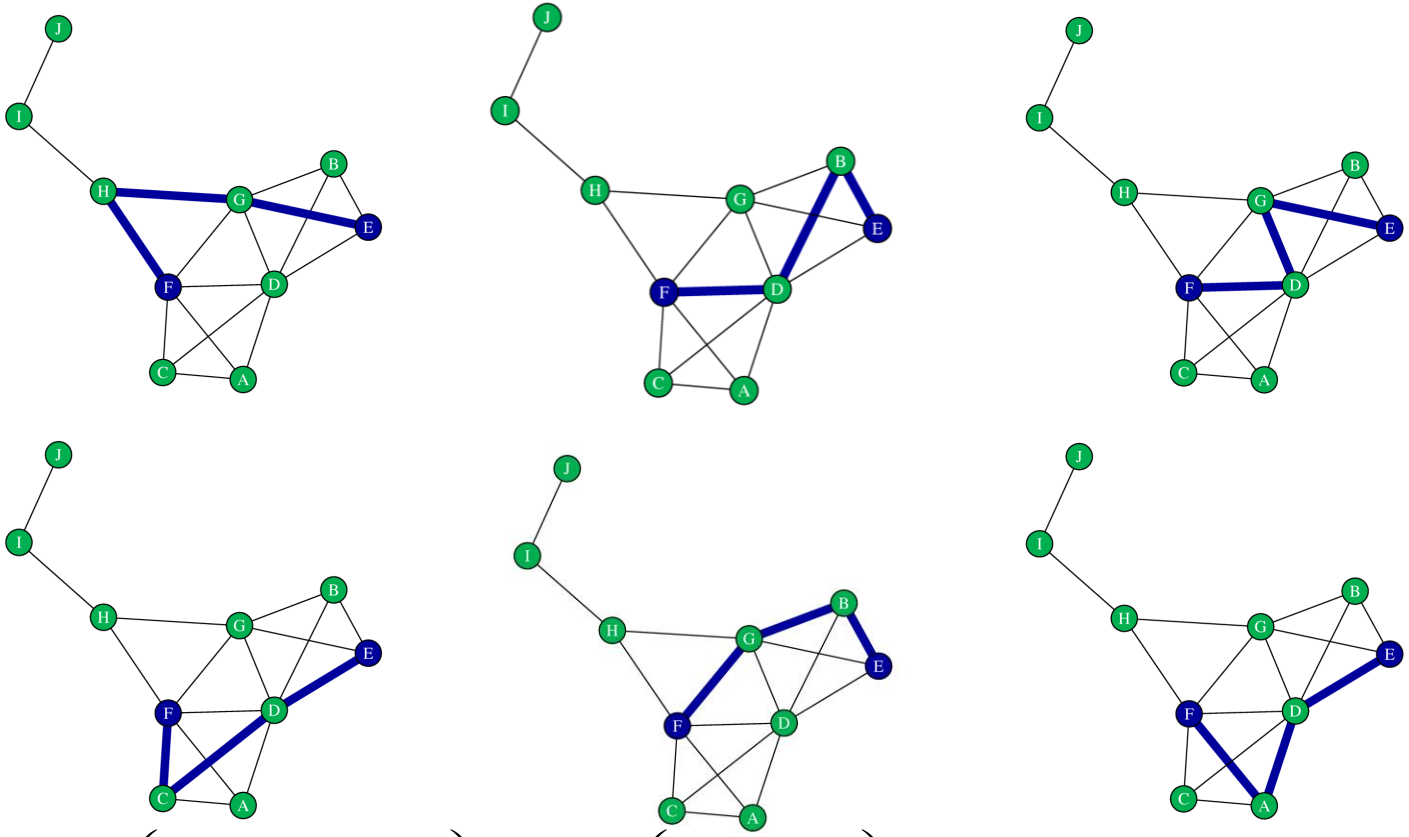


A random walker has a 1/160 = 0.67% chance of walking along **F**-**G**-**D**-**E**, when starting at **F**.

Note: that this is not the same probability of walking from **E**-**D**-**G**-**F** (=1/90).

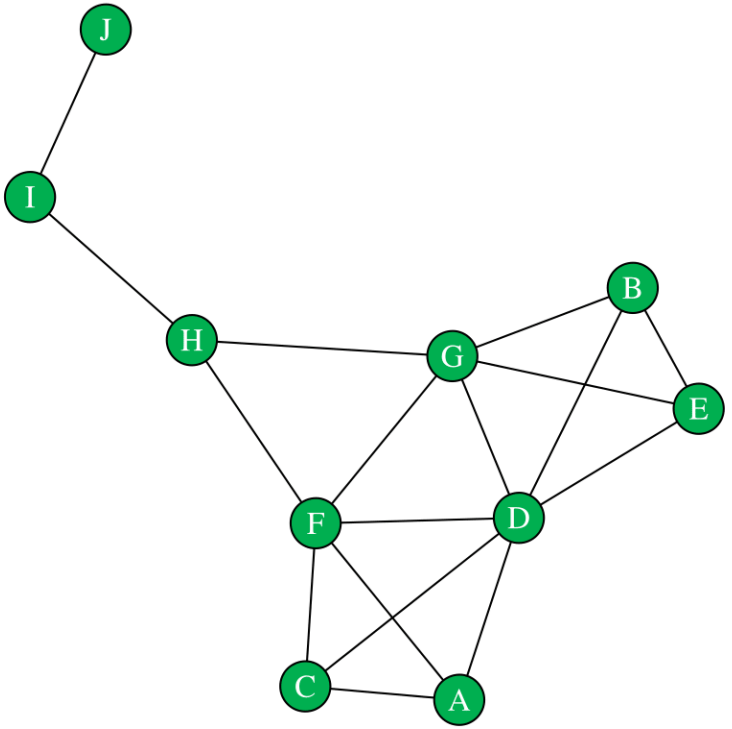
$$LRW^3{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{54} + \ldots \right) + \frac{k_F}{2|E|} \left(\frac{1}{150} \right)$$

This is subsequently done for all 3-step paths connecting F and E



$$LRW^3{}_{EF} = \frac{k_E}{2|E|} \cdot \left(\frac{1}{90} + \frac{1}{54} + \ldots \right) + \frac{k_F}{2|E|} \left(\frac{1}{150} + \ldots \right) = 0.0048$$

The LRW^3 score is subsequently calculated for all node pairs



How to use these scores to predict links will be explained after all models have been presented.

Superposed random walk is an extension of the local random walk

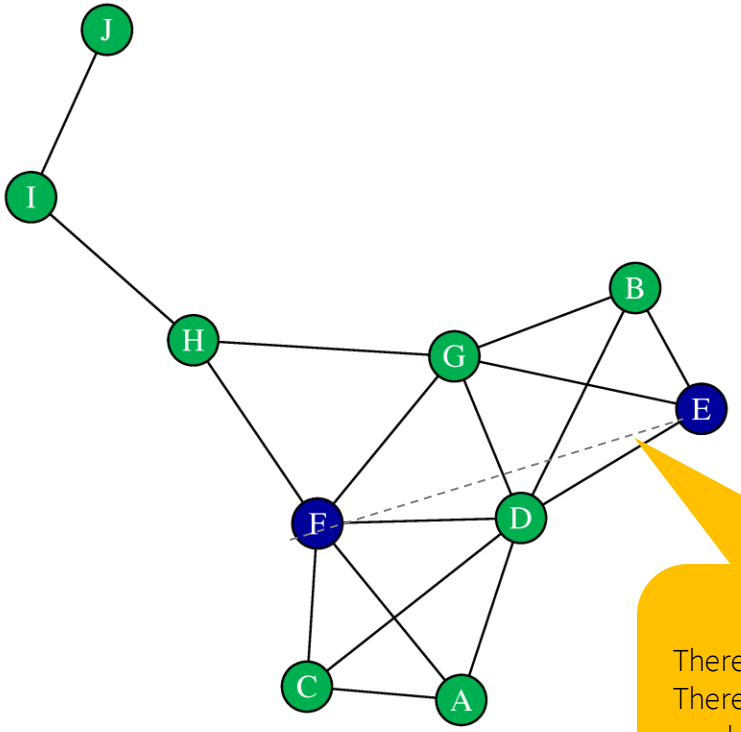
The **superposed random walk** score for nodes *i* and *j* aggregates the local random walk scores for all paths of length *n*, and fewer.

$$SRW^{n}_{ij} = \sum_{k=1}^{n} LRW_{ij}^{k}$$

In addition to the **LRW³** (see above), **LRW¹** and **LRW²** need to be calculated for all node pairs.

$$SRW^{3}_{EF} = LRW^{1}_{EF} + LRW^{2}_{EF} + LRW^{3}_{EF}$$

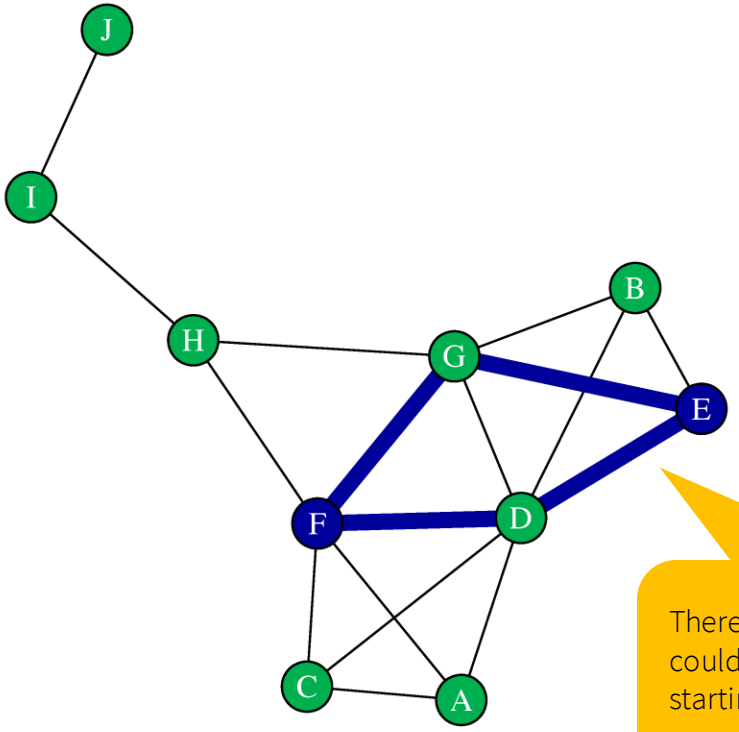
Calculating superposed random walk between nodes E and F for n = 1



There are no paths between **E** and **F**. Therefore there is no possible path a random walker could walk to reach **F** from **E**, or **E** from **F** in 1 step.

$$SRW^{3}_{EF} = 0 + LRW^{2}_{EF} + 0.0048$$

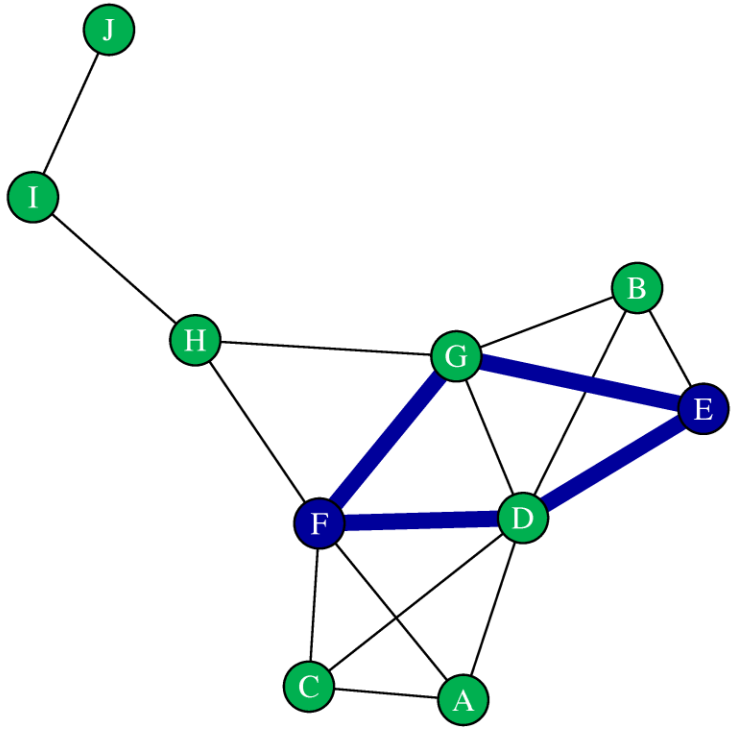
Calculating superposed random walk between nodes E and F for n = 2



There are 2 paths a random walker could travel to reach **F** (respectively **E**) starting from **E** (respectively **F**): over **D**, or over **G**.

$$SRW^{3}_{EF} = 0 + LRW^{2}_{EF} + 0.0048$$

Using the same approach demonstrated on 3-step paths, calculate the LRW²



$$SRW^{3}_{EF} = 0 + 0.0216 + 0.0048 = 0.0264$$

Finally: How to use the scores for link prediction

Input

Unconnected node pairs	Score
A-D	0.162
A-F	0.158
A-G	0.144
B-C	0.113
...	...
J-K	0.004

Set threshold at *n* predicted links.

Output

Unconnected node pairs	Score	Link prediction
A-D	0.162	1
A-F	0.158	1
A-G	0.144	1
...	...	...
B-C	0.010	0
J-K	0.004	0

These edges are predicted, i.e. we infer that they are missing in the observed network or will evolve in the future.

These edges do and will not exist according to our model and the threshold we set.

MLE estimates the connectivity parameter by maximizing the likelihood of the observed structure

MLE is a procedure of finding the value of one or more parameters of a given statistic.

Applied to a network

$$G = (V, E),$$

Maximum Likelihood Estimation tries to determine the probability p_{AB}, that any two nodes $A, B \in V$ are connected.

One MLE based algorithm is the hierarchical structure model

1. The Hierarchical Structure Model (HSM) focuses on the **hierarchical structure** of a network to predict missing links.
2. Accordingly, this algorithm yields the best predictions for networks which exhibit a **clear hierarchical structure**.
3. The model utilizes sampling of so-called **dendrograms** to derive the probability that two unconnected nodes within a network are connected.
4. The calculated probabilities are then ranked and links are predicted for the pairs of nodes with the highest probabilities.

The hierarchical structure of the network is represented by dendrograms

When sampling the dendrograms, the likelihood of a diagram is:

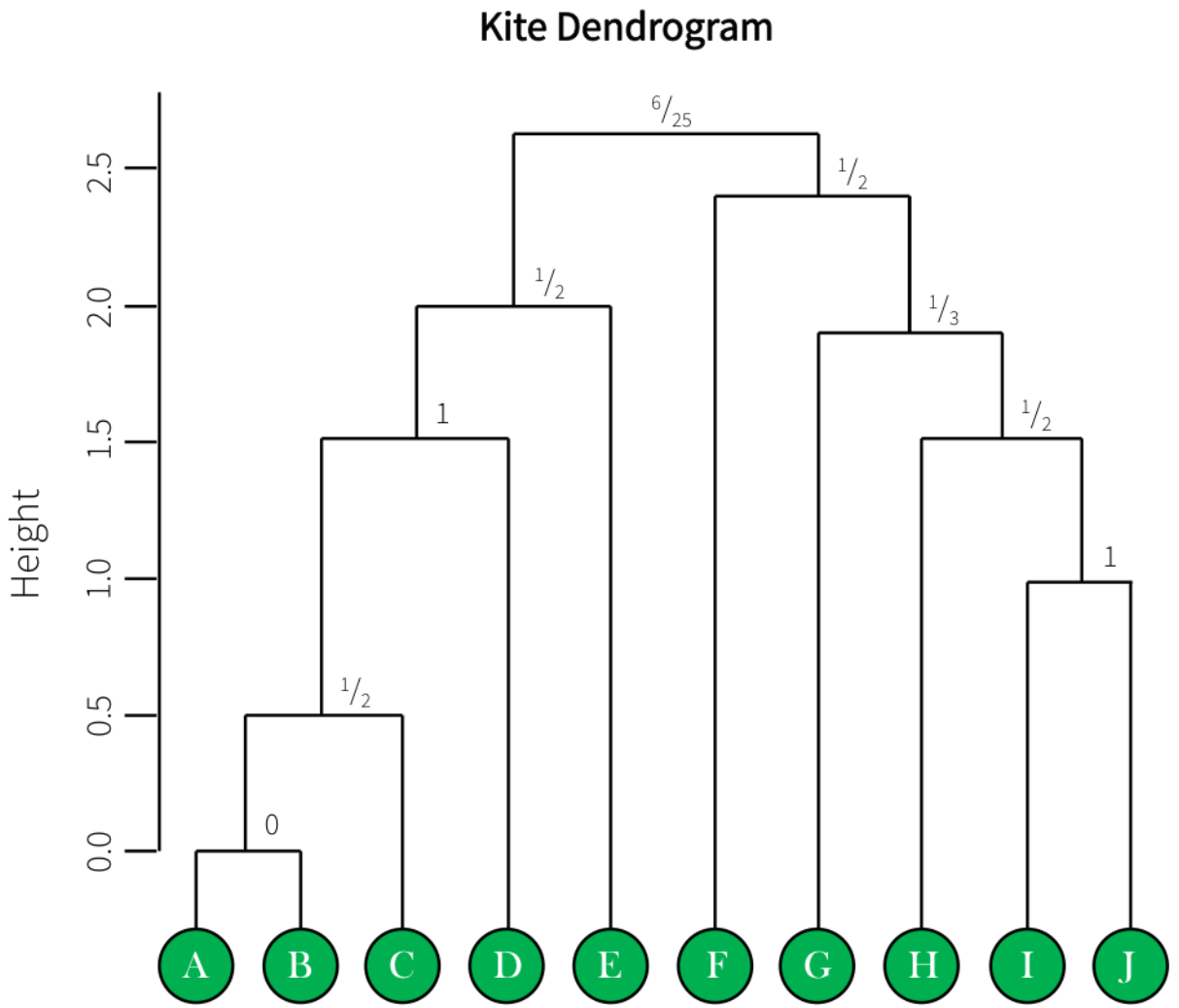
$$\mathcal{L}(D) = \prod_{r \in D} \left[\bar{p}_r^{\,\bar{p}_r} (1 - \bar{p}_r)^{1 - \bar{p}_r} \right]^{L_r R_r}$$

Where the merging probabilities $\bar{p}_r$ denoted in the diagram are calculated as follows

$$\bar{p}_r = \frac{E_r}{L_r R_r}$$

Here, E_R is the number of edges between both groups and L_r, R_r the number of vertices in the left respectively right subtrees.

This is a possible dendrogram for the kite network…



…and this is how the connecting probabilities were calculated (on the example of nodes I and J)

1. $L_r = 1$, because the Left-hand cluster (here containing node *I*) contains 1 node
2. $R_r = 1$, because the Right-hand cluster also contains 1 node (node *J*)
3. At last, $E_r = 1$, because there exists only 1 link connecting the left-hand and right-hand cluster. Therefore

$$\bar{p}_r = \frac{E_r}{L_r R_r} = \frac{1}{1 * 1} = 1$$

4. Given all merging probabilities $\bar{p}_r$, the likelihood of the dendrogram on the previous slide to be sampled is proportional to:

$$4 * \frac{1}{2} * 4 * \left(\frac{1}{2}\right)^2 * \left(\frac{1}{3}\right)^1 * \left(\frac{2}{3}\right)^2 * \left(\frac{6}{25}\right)^6 * \left(\frac{19}{25}\right)^{19} \approx 7.698e-08$$

Probabilistic Models focus on the unobserved underlying structure of an observed network

1. **Idea:** Imagine you would like to predict the body height of humans with a regression. You would fit a model with a set of variables (e.g. age, height of parents, etc.) to the observed body heights and estimate the according parameters. Then you can express every potential body height as a probability distribution conditional on the estimated parameters.

2. The probabilistic link prediction methods follow the **same basic idea**: To every observed network, there is an underlying unobserved structure that can be formulized in a model with a set of parameters. Then every non-existent link can be described with a probability distribution conditional on these parameters.

One class of probabilistic methods for link prediction are the probabilistic relational models (PRM)

1. Instead of using a single graph to model, PRMs use three graphs for every network:
    1. The data graph (also called the network's skeleton),
    2. The model graph,
    3. The inference graph (also called the ground graph).
2. An important feature of PRMs is that they **group both nodes and edges into different types**, i.e. the model is particularly useful for bi-, tri- and more-partite networks.

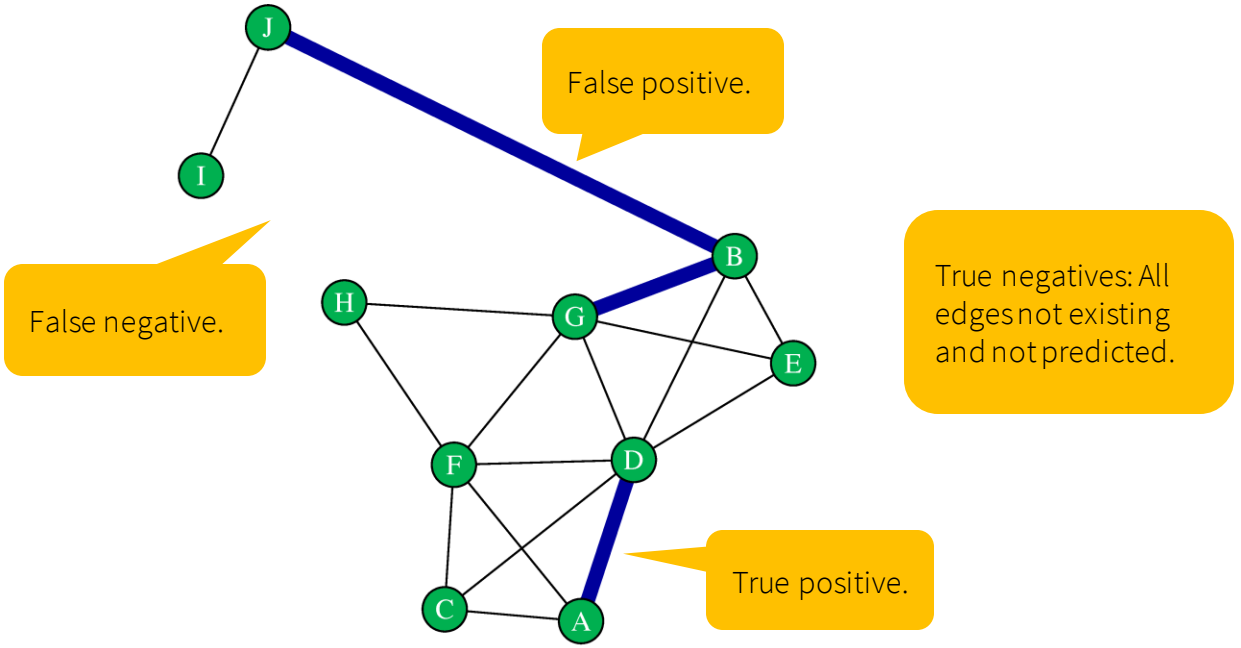
➡ We don't apply a probabilistic model in detail here.

Link Prediction
Agenda

Evaluating our classifiers' performance

1. For every unobserved link between nodes we make a prediction based on the similarity measures.

2. Using the out-of-sample edges, we can evaluate how well the classifier performed.



In a first step we can use a confusion matrix to evaluate our classifier

	Link exists	No link exists
Link predicted	True positive (TP)	False positive (FP)
No link predicted	False negative (FN)	True negative (TN)

Each cell records the amount of cases (e.g. of false negatives).

A trade off exists between false positives and false negatives

1. The results presented in the confusion matrix are dependent on the threshold.
2. The threshold you set alters how many false positives, versus false negatives you get.
3. Depending on the situation the effect of false positives can be worse or better than the effect of false negatives.

➡ How do you chose the best threshold?

First let's looks at some statistics used to measure the performance of a classifier

A list of statistics with which the performance of a classifier can be evaluated can be found at the link below.

The most important ones are:

Accuracy

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

How many were correctly classified in total?

True positive rate (Sensitivity, recall)

$$TPR = \frac{TP}{TP + FN}$$

How many existing links were correctly classified?

True negative rate (Specificity)

$$TNR = \frac{TN}{FP + TN}$$

How many non-existing links were correctly classified?

https://en.wikipedia.org/wiki/Sensitivity_and_specificity

First let's looks at some statistics used to measure the performance of a classifier

A list of statistics with which the performance of a classifier can be evaluated can be found at the link below.

The most important are:

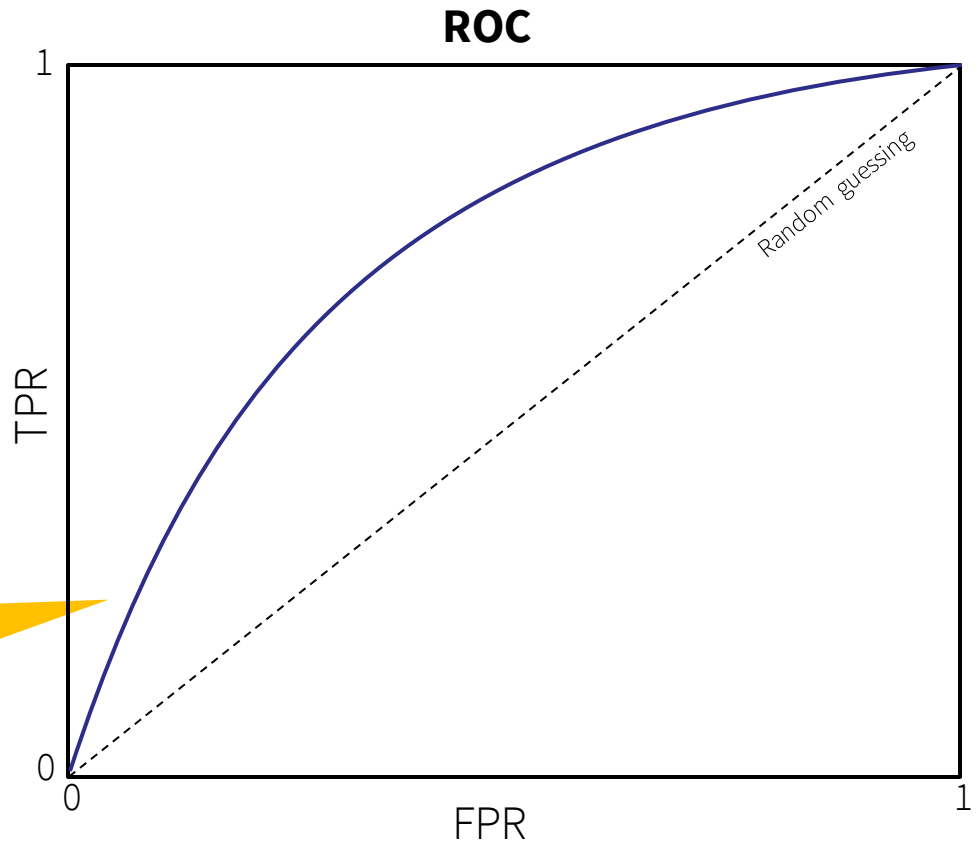
Accuracy	True positive rate (Sensitivity, recall)	True negative rate (Specificity)
$ACC = \frac{TP + TN}{TP + TN + FP + FN}$	$TPR = \frac{TP}{TP + FN}$	$TNR = \frac{TN}{FP + TN}$

We can calculate the false positive rate (FPR) = 1 – TNR = $\frac{FP}{FP + TN}$

How many non-existing links were **incorrectly** classified?

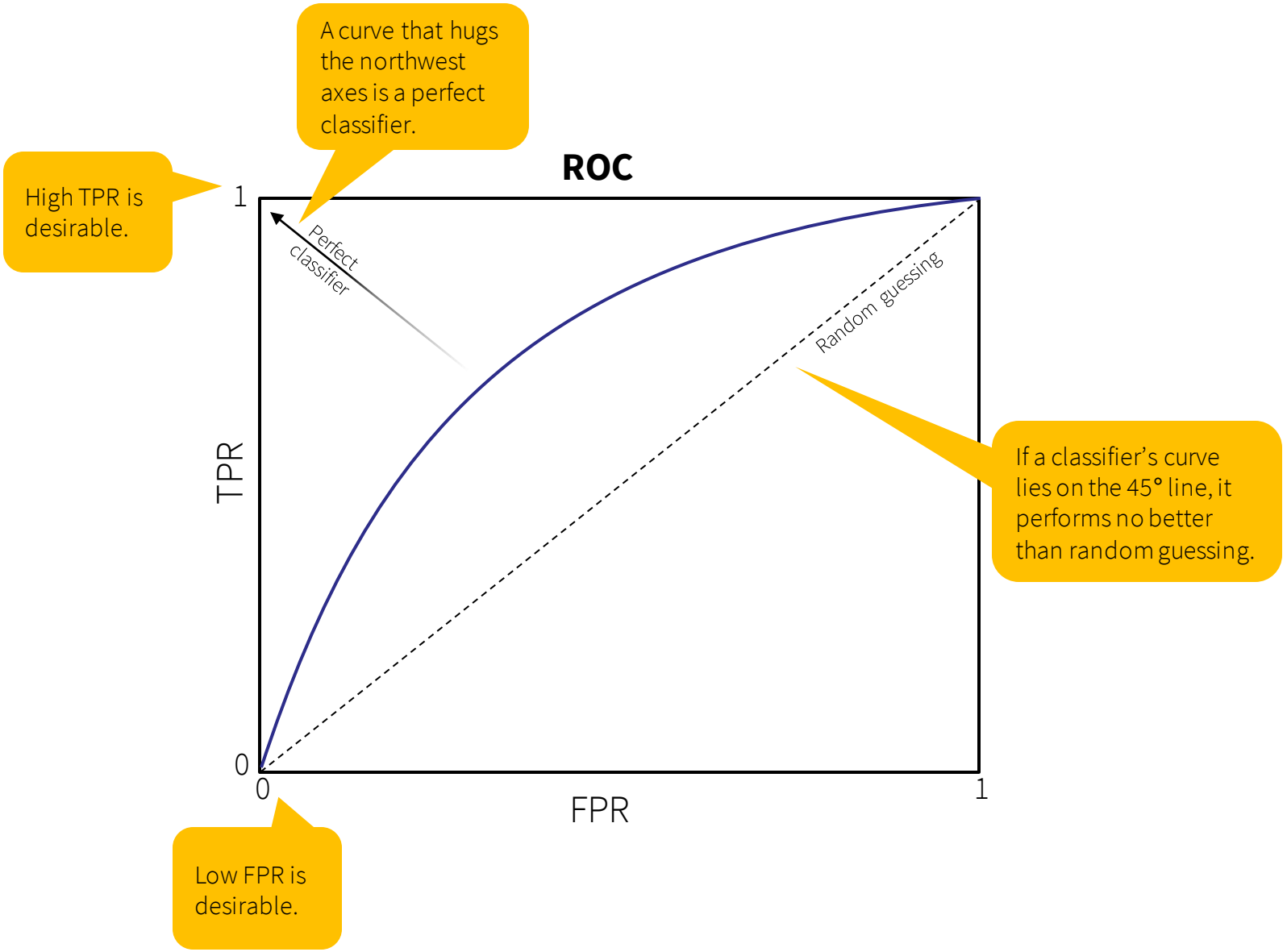
https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Receiver operating curve (ROC) plots the trade off between the TPR and the FPR if we vary the threshold



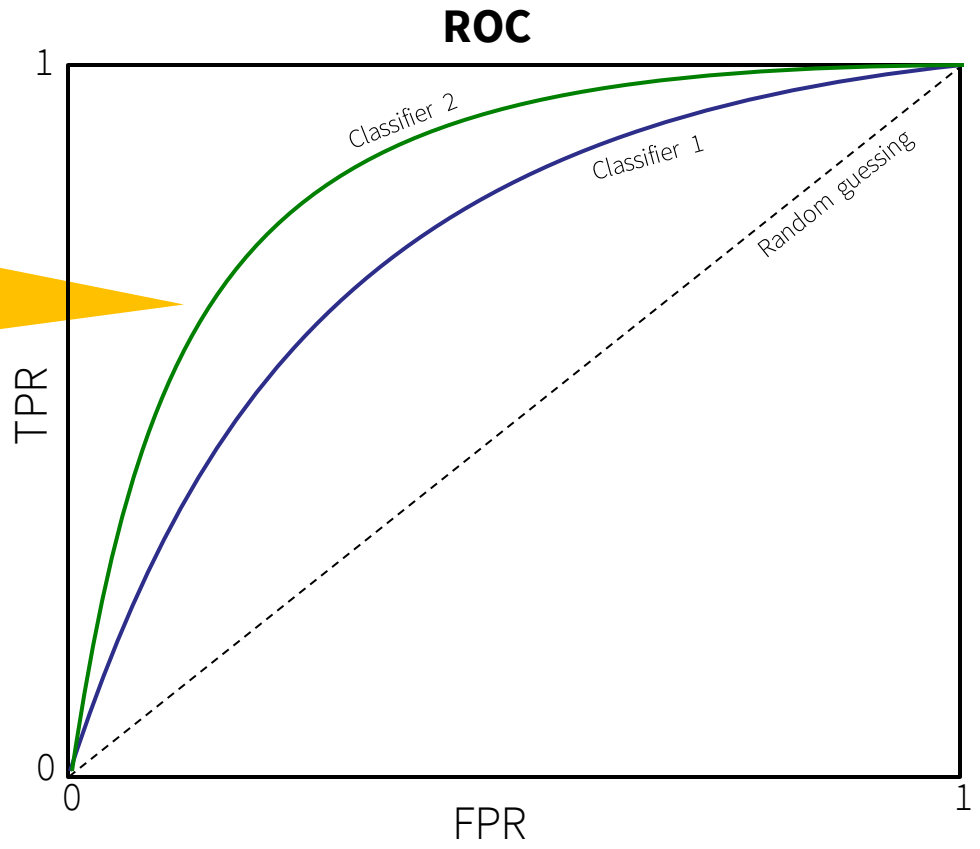
We can plot the **ROC** by varying the classification threshold between 0, and 1.

How to read an ROC

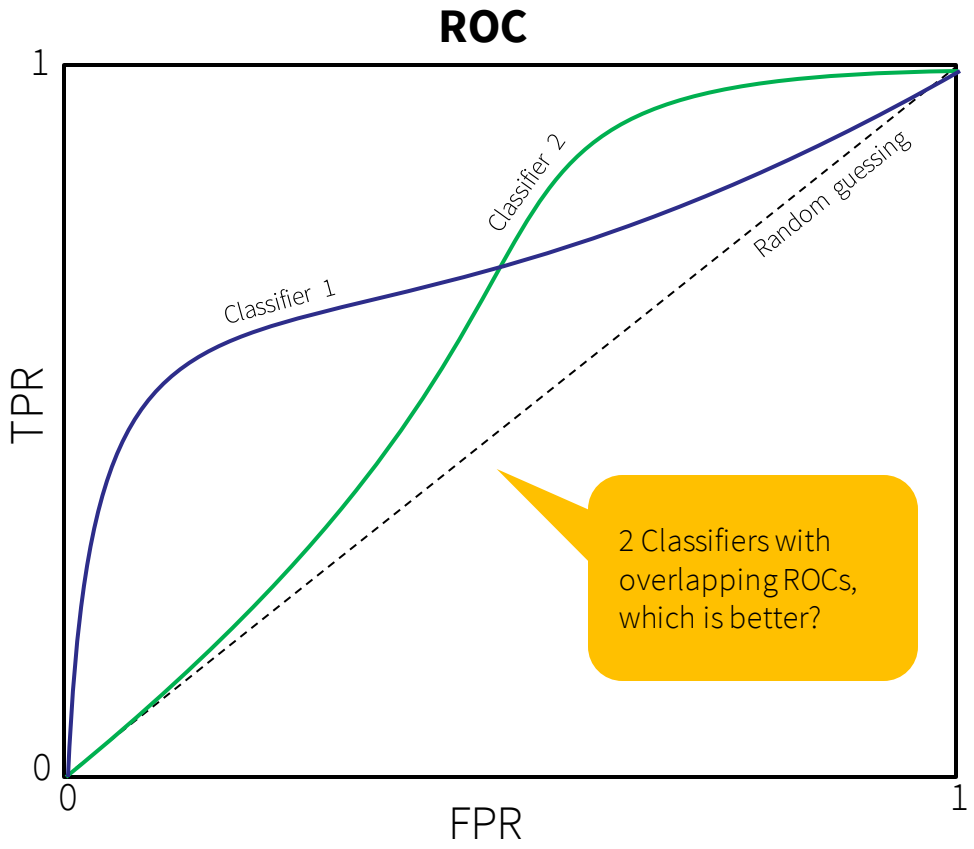


Comparing multiple classifiers

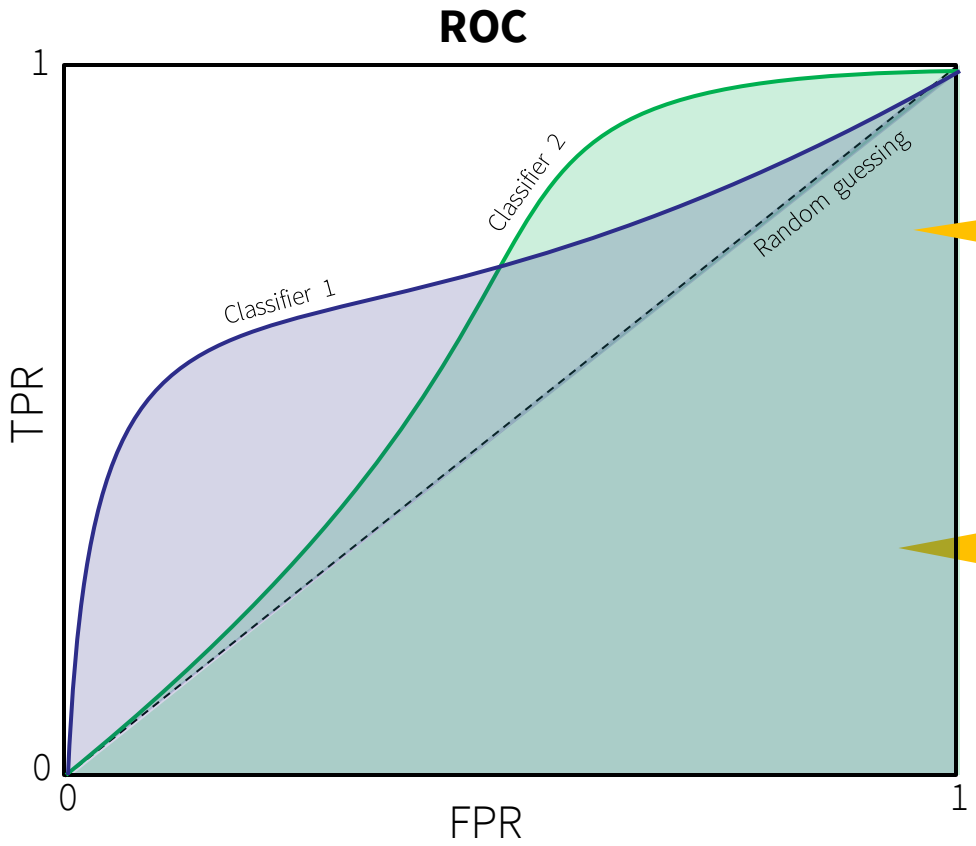
Multiple classifiers can be plotted and compared. The tighter the curve hugs the axes the better its performance.



So, which classifier is better?



So, which classifier is better?
Use the area under the (ROC) curve



The area under the curve (AUC) can be used to evaluate the classifiers based on their ROC.

Perfect classifiers have an AUC = 1. The closer the AUC is to 1, the better the performance of the classifier is.

Which method predicts the best? Use the AUC Value

1. The resulting AUC Value of different predicting methods can be compared.
2. The higher the AUC (closer to 1), the better is the underlying predicting method.
3. **Network specifics** are likely influence the methods' performance.
4. The **computation time** that R needs to process the method is also a relevant information, as effectivity could play an important role in the field.

To evaluate the performance of the presented algorithms, we apply them to a real-world network

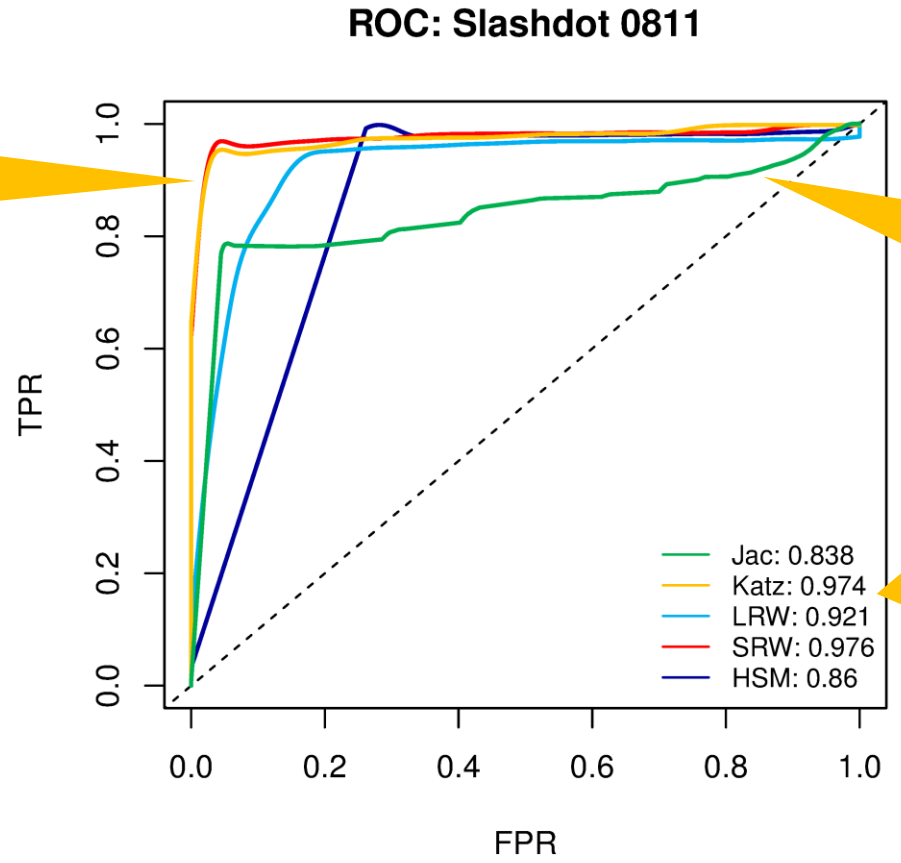
1. **Slashdot** is a technology-related news website known for its specific user community.
2. We took a random subset of 200 nodes.
3. Users in the network are linked if they are friends (or foes).

Network Descriptives	
Number of nodes	200
Number of edges	1592
Clustering coefficient	0.1080015
Degree heterogeneity	5.283642
Largest connected component	200/1

https://snap.stanford.edu/data/soc-Slashdot0811.html

This is how the various link prediction algorithms compare

The Katz score, and the superposed random walk display the best performance.



The weakest performance is displayed by the Jaccard, followed by the hierarchical structure model.

AUC scores are noted in the legend.

Comparing Katz and superposed random walk: Accuracy vs. runtime

The Katz global similarity measure, and the superposed random walk both display good performance:

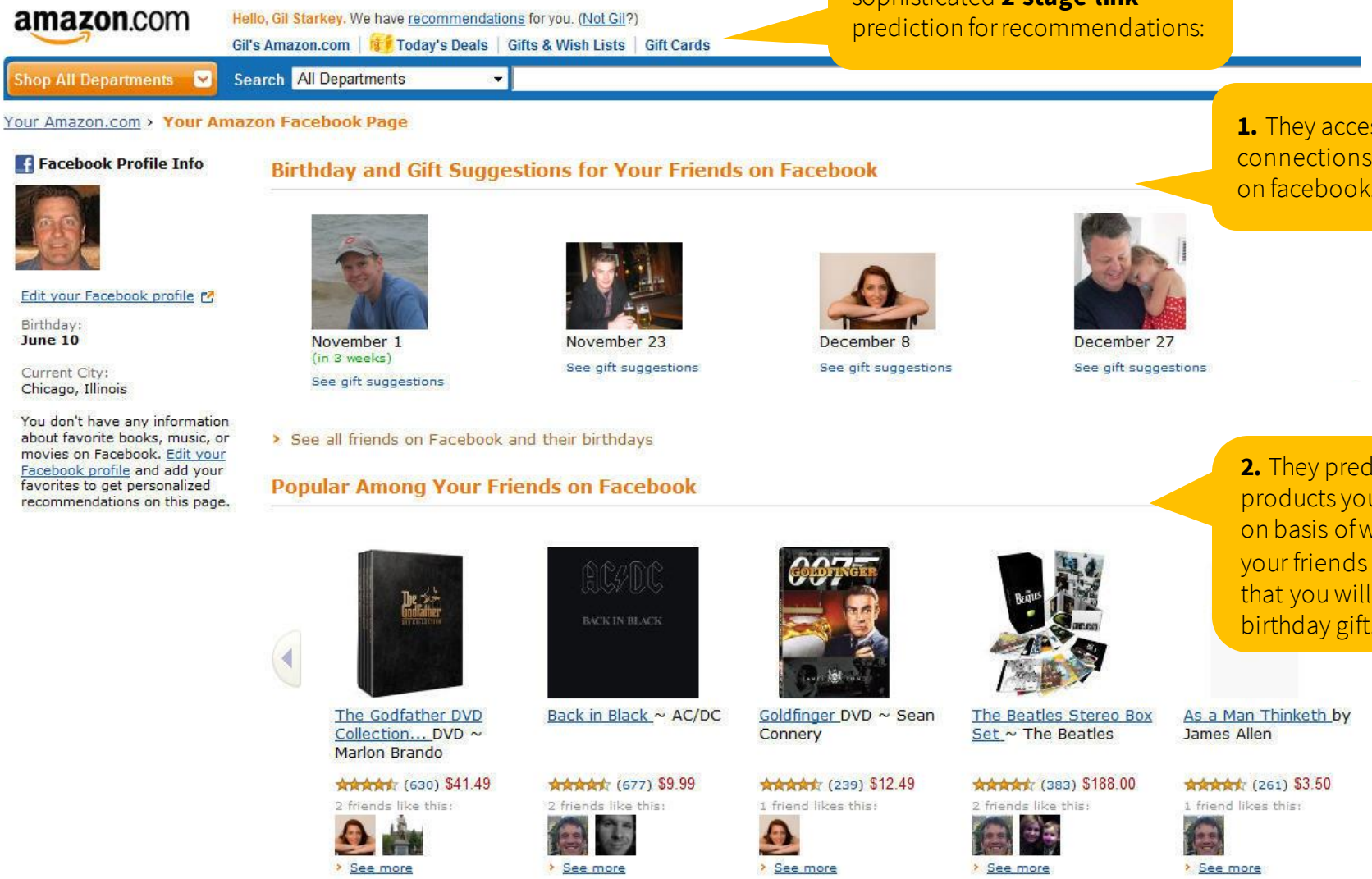
Katz AUC	SRW AUC
0.9761	200

The SRW's performance is superior, however, it comes at the **cost of longer computation time** compared to the Katz algorithm.

Link Prediction
Agenda

Commercial applications (especially in online retail) are abundant

Amazon seems to use a sophisticated **2-stage-link** prediction for recommendations:



1. They access your connections (existing links) on facebook.

2. They predict the products you will likely buy on basis of what products your friends like, assuming that you will buy them a birthday gift.

Public security application

1. Networks of terrorist cells are often described as “amorphous, invisible, resilient, dispersed”.
2. Since criminal networks make substantial efforts to operate in secrecy, **incompleteness** of the observed networks is a dominant problem in criminal network analysis.
3. Lü and Zhou (2011) applied several link prediction algorithms on the terrorist network – potentially a promising field of application in the future.

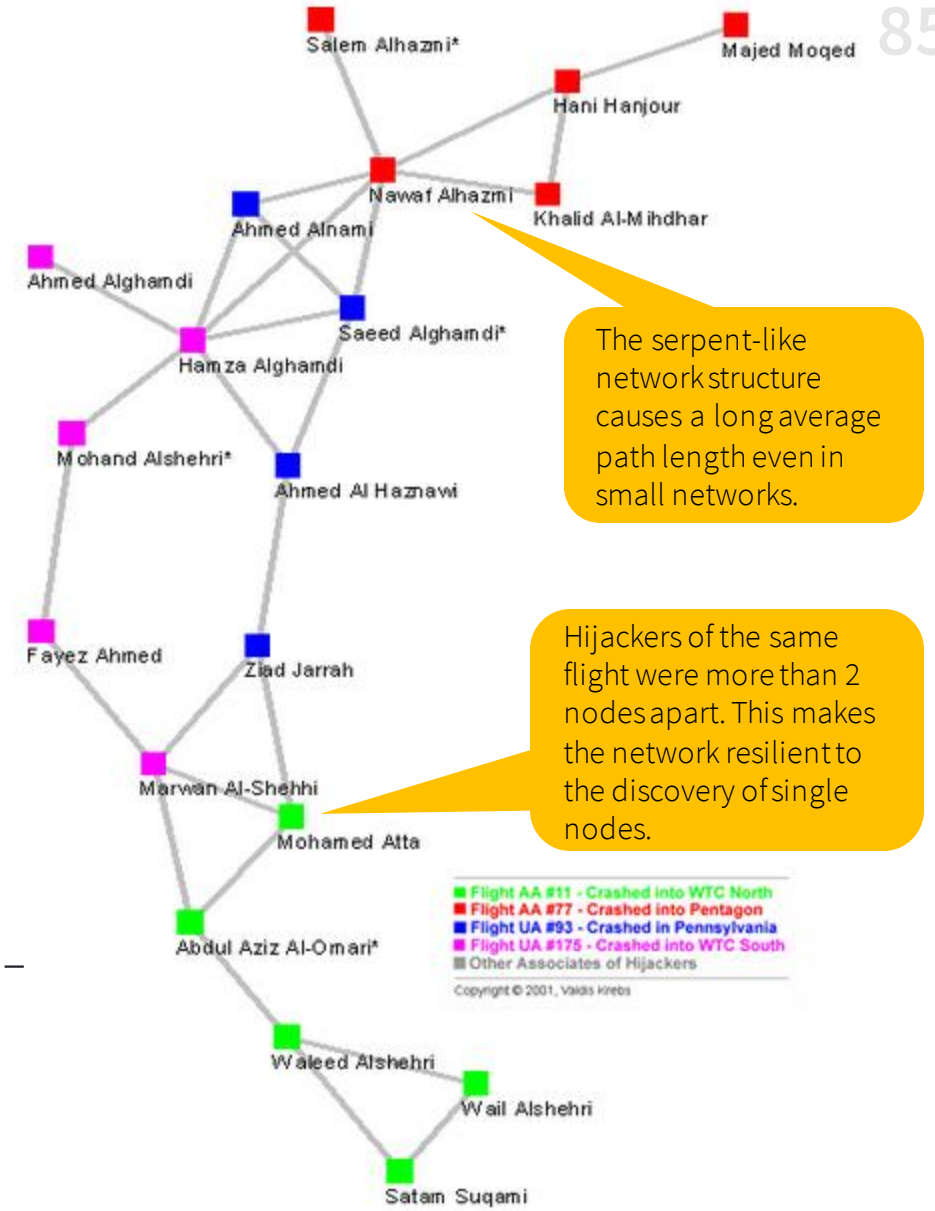


Figure 2

Krebs (2002)

Link Prediction
Agenda

Summary

1. Various link prediction algorithms from the literature have been presented and discussed.
2. The performance of the different algorithms depends on how well their underlying assumptions match the characteristics of the specific network to which they are applied.
3. Therefore, it is worthwhile to invest some effort in getting to know the nature and specifics of the particular network before applying the link prediction algorithms.
4. Link prediction is already widely used in online retailing. Many further areas of application in the future are imaginable.

Link Prediction
Agenda

References

1. Getoor, Lise, and Christopher P. Diehl. "Link mining: a survey." *ACM SIGKDD Explorations Newsletter* 7, no. 2 (2005): 3-12.
2. Krebs, Valdis E. "Mapping networks of terrorist cells." *Connections* 24, no. 3 (2002): 43-52.
3. Lü, Linyuan, and Tao Zhou. "Link prediction in complex networks: A survey." *Physica A: Statistical Mechanics and its Applications* 390, no. 6 (2011): 1150-1170.
4. Paarsch, Harry J., and Konstantin Golyaev. (*A Gentle Introduction to Effective Computing in Quantitative Research.* The MIT Press: Cambridge, Massachusetts, 2015.
5. Samatova, Nagiza F., William Hendrix, John Jenkins, Kanchana Padmanabhan, and Arpan Chakraborty, eds. *Practical Graph Mining with R*. CRC Press, 2013.

Networks Analytics
Group Assignment

Stefan Bublitz stefan_bublitz@access.uzh.ch

Luca Gaegauf luca.gaegauf@access.uzh.ch

Pascal Sutter pascal.sutter@uzh.ch

Salome Lang salome.lang@uzh.ch

University of Zurich[UZH]

Appendix: Comparing the methods on several real-world networks

Dataset/ Method	Jaccard	Katz	LRW	SRW	HSM
Slashdot 0811	0.8383	0.9815	0.9214	0.9781	0.8314
Slashdot 0902	0.8193	0.9795	0.9231	0.9779	0.8214
Pokec	0.6648	0.9828	0.9828	0.9859	0.7748
Average	0.7740	0.9813	0.9424	0.9806	0.8092
Variance	0.00903636	2.76333E-06	0.001222823	2.08133E-05	0.00091252
Duration (s)	15	40	766	728	900

The Jaccard predicting method is the fastest one, Katz is the second.

The Katz predicting model delivers the best AUC value in average and for the most datasets.