

## SOCIAL NETWORK ANALYSIS

### BAD714D

#### MODULE-3

Link Analysis - Applications of Link Analysis, Signed Networks, Strong and Weak Ties, Link Analysis Algorithms, PageRank, Personalised PageRank, DivRank, SimRank, PathSIM.

#### T1 – Chapter 4 (4.1 – 4.8)

### 3.1 APPLICATIONS OF LINK ANALYSIS

**Entity ranking:** Firstly, link-based ranking is used to find relevant/similar matches for a given input pattern against a large corpus of information. This varies from simple hyperlink-based ranking to more nuanced similarity constructs including text, images, etc. This idea can also be extended to the areas of computer networking, route balancing, and caching.

**Anomaly detection:** On the other end of the spectrum, link analysis can also be used to determine anomalous behaviour when known patterns are violated or missing (discussed in Chapter 8). The most fundamental application of this is in criminal investigations and the proactive prevention of crimes. Another area of application of link analysis is the prediction of an early outbreak of diseases.

**Mining new patterns:** A by-product of link analysis is the task of mining newer patterns of interest. This can be useful in determining a new set of web pages that are gaining hyperlinked traction but are not indexed yet, or new relations that are about to form in the network.

**Adversarial attack:** A notorious application of link analysis, or the determination of important nodes in a network, is enabling adversarial attacks on selected nodes to disrupt services (online as well as offline).

### 3.2 SIGNED NETWORKS

So far, the networks we have studied have edges that are directed/undirected in addition to being weighted/unweighted. While the direction and weight of an edge  $w(v, u)$  capture the direction of flow of information and the strength of influence of that

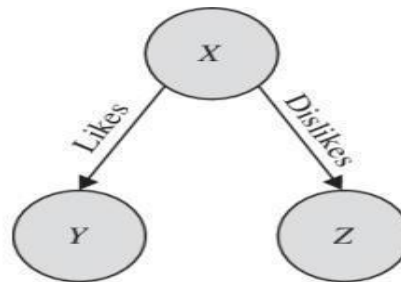
information respectively, they do not capture **how the information is perceived by the receiving node**.

While most examples of user networks capture friends/co-author networks — where the absence of an edge is an indicator of the lack of friendship/co-authorship between nodes — the so-called friendship dynamics are more complex, both online and offline. In most exchanges, there is an element of like/dislike (YouTube), agree/disagree (Reddit), positive/negative review (Amazon), etc.

In order to capture this reaction or opinion, we make use of **signed networks**. A signed network can be thought of as a network where each edge is labelled with a **positive or negative sign** (along with direction and weight, if applicable). Signed networks are better at capturing not only the presence of a relation, but also the **nature of the relationship** between entities.

For example, while users on Twitter follow political leaders of different ideologies, they may not necessarily agree with all the ideologies — this preference can be captured by the positive/negative sign along the follower–followee edges, as shown in Figure 4.1.

Figure 4.1: A signed network diagram showing user X following users Y and Z. The edge from X to Y is labeled 'Likes' (positive), and the edge from X to Z is labeled 'Dislikes' (negative).



**FIGURE. 4.1.** A Twitter network can be modeled as a signed network. User X follows both users Y and Z. While X supports the views of Y, it dislikes the views of Z, yet continues to receive updates from both

In addition to capturing the dynamic of the relationship, signed networks can also be used to capture the structural balance and status of the nodes in the network. In fact, the concept of signed network in complex systems has been adopted from fields of sociology and psychology, studying the power dynamics among people in a network.

### 3.2.1 Balance Theory of Undirected Signed Networks

Consider two nodes A and B which form a signed edge. In solidarity,  $(A,B,+)$  indicates that A and B are friends, and  $(A,B,-)$  captures enmity. Left on its own, either

friendship/enmity is a balanced system. However, upon introduction of a third node C forming a signed triadic closure with A and B, balance may no longer prevail. The combination of signs supported by the triad (A,B,C) is shown in Figure 4.2. Based on Figure 4.2 and Table 4.1, we can infer the following:

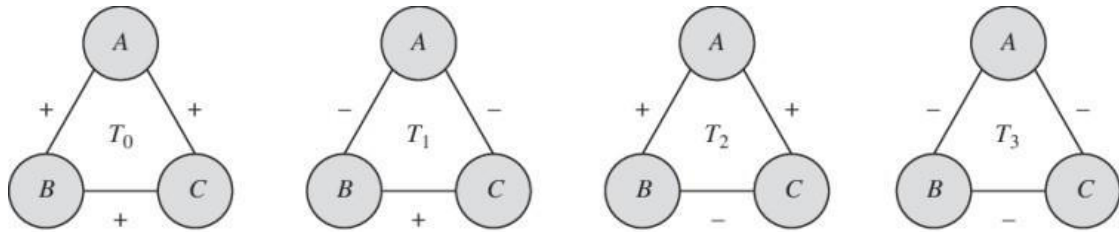


FIGURE 4.2. The combinations of signed triads in an undirected network.

Table 4.1. Various dynamics possible in a signed undirected triad where  $\langle A, B \rangle$ ,  $\langle A, C \rangle$  and  $\langle B, C \rangle$  are edges among the three nodes A, B, and C. The triads are shown in Figure 4.2.

| Triad | AB | AC | BC | Dynamics                          | Product of signs | Stable |
|-------|----|----|----|-----------------------------------|------------------|--------|
| $T_0$ | +  | +  | +  | Everyone is trustworthy friends   | +                | Yes    |
| $T_1$ | -  | -  | +  | Two friends share a mutual enemy  | +                | Yes    |
| $T_2$ | +  | +  | -  | Two enemies share a mutual friend | -                | No     |
| $T_3$ | -  | -  | -  | Everyone is a distrusted enemy    | -                | No     |

A triad consisting of three positive signs is **balanced (T0)**. When everyone is a friend of others, there is mutual trust and respect, and the structure is stable.

A triad consisting of two negative signs and one positive sign is also **balanced (T1)**. The mutual enmity towards a third party between two friends provides a common ground of interaction, keeping the structure stable, with trust between nodes established based on distrust towards a common entity.

A triad consisting of two positive signs and one negative sign is **unbalanced (T2)**. The mutual friends would be under stress to take sides and support one friend over the other, while the two friends maintain animosity. The structure remains balanced only if the common friend is able to influence the enemy nodes to become friends as well, or one of the enemy nodes exerts greater pressure on the common friend and changes its opinion about the third person — leading to a scenario of mutual enmity towards a third party.

A triad consisting of three negative signs is also **unbalanced (T3)**. When there is no mutual trust among the nodes, the structure is unstable and will disintegrate — unless two enemies come together and conspire against the third, becoming friends in the process, leading to a balanced case of friends sharing a common enemy.

A triad is **balanced** if the product of the signs of individual cycles (starting with one node at a time) is always **positive** (T0, T1).

Balance theory is more likely to **overestimate** the presence of balanced/stable triads and **underestimate** the presence of unbalanced/unstable triads, both in comparison to chance. Thus, according to balance theory, stable triads will be over-represented in a real-world scenario. In other words, as unbalanced triads can be a point of contention, causing stress and mistrust, people will try to minimise them in their interpersonal relationships. Hence, unbalanced triads (T2, T3) would be under-represented in the network.

### 3.2.2 Status Theory of Signed Networks

While balance theory views the edge-sign between two nodes as an indication of friend or foe, the **status theory** views the edge-sign as an indicator of the **relative status/prestige** of a node with respect to the other.

- If **A** initiates a positive link to **B**, status theory assumes that A considers B to have a **higher status** than itself.
- If **A** initiates a negative link to **B**, then A considers B to have a **lower status** than itself.

#### Example 4.1

Based on Figure 4.1, by initiating a positive link from **X** to **Y**, user X thinks that Y is higher in status than itself. Similarly, by initiating a negative link from **X** to **Z**, user X thinks that it is higher in status than Z. Thus, the relative status of the nodes with respect to X is:

$$Y > X > Z$$

It should also be noted that while balance theory is by default **undirected**, status theory is underpinned by **both sign and direction**, as it captures relative status.

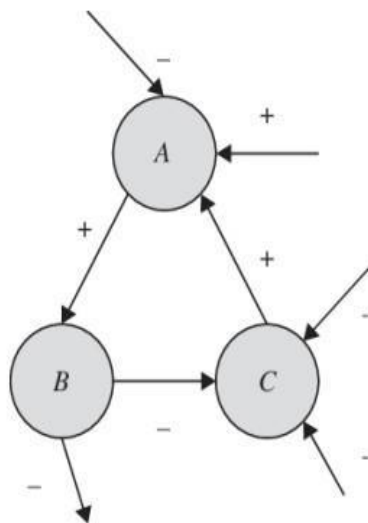
On the basis of the sign initiated by a node, two node-level metrics can be generated:

- **Generative baseline:** The fraction of positive signs generated by a node — i.e., out of all the signs initiated by a node, how many are positive.
- **Receptive baseline:** The fraction of positive signs received by a node — i.e., out of all the signs received by a node, how many are positive.

**Example 4.2**

Based on the snapshot in Figure 4.3:

- Edges generated by A = 1 (+), and edges received by A = 3 (+, +, -)
- Edges generated by B = 2 (-, -), and edges received by B = 1 (+)
- Edges generated by C = 1 (+), and edges received by C = 3 (-, -, -)



**FIGURE 4.3.** A snapshot of positive and negative links received by three nodes A, B, and C.

Thus, the scores for generative baselines ( $g$ ) are

$$A_g = \frac{1}{1} = 1, B_g = \frac{0}{2} = 0, C_g = \frac{1}{1} = 1$$

Similarly, the scores for receptive baselines ( $r$ ) are

$$A_r = \frac{2}{3} = 0.667, B_r = \frac{1}{1} = 1, C_r = \frac{0}{3} = 0$$

**FIGURE 4.3.** A snapshot of positive and negative links received by three nodes A, B, and C.

Thus, the scores for **generative baselines (g)** are:

$$A_g = \frac{1}{1} = 1, B_g = \frac{0}{2} = 0, C_g = \frac{1}{1} = 1$$

Similarly, the scores for **receptive baselines (r)** are:

$$A_r = \frac{2}{3} = 0.667, B_r = \frac{1}{1} = 1, C_r = \frac{0}{3} = 0$$

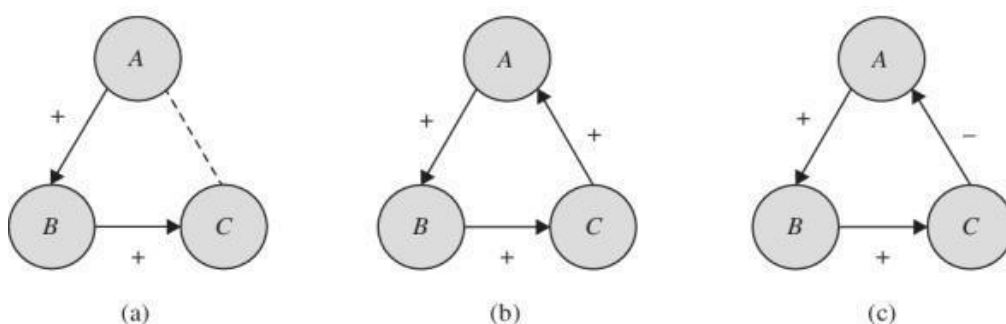
#### Notation key:

- $A_g, B_g, C_g$  = generative baseline scores (fraction of outgoing signs that are positive)
- $A_r, B_r, C_r$  = receptive baseline scores (fraction of incoming signs that are positive)
- T0, T1, T2, T3 = the four triad sign-configurations (balanced: T0, T1; unbalanced: T2, T3)

### 3.2.3 Triad Balance vs. Status

In a directed cyclic triad, if A initiates a positive link towards B, and B initiates a positive link towards C, then based on status theory we get  $\text{status}(B) > \text{status}(A)$  and  $\text{status}(C) > \text{status}(B)$ . By transitivity,  $\text{status}(C) > \text{status}(A)$ ; and when C initiates a link towards A then it will be negatively signed. Under an undirected graph balance theory, where (A,B,+) and (B,C,+), it will predict the edge between C and A to be positive instead. Thus, with respect to the above example, two theories seem to be at conflict with each other. It also indicates that balance theory will overestimate the occurrence of positive triplets, as shown in Figure 4.4.

The study conducted by Leskovec et al. (2010) provides a comprehensive real-world overview of the links generated as a result of both the theories and the insights obtained from them.



(4.1) FIGURE 4.4. (a) A system where A is positively linked to B, and B is positively linked to C. (b) The resultant link between A and C based on the balance theory. (c) The resultant link between A and C based on status theory.

### 3.3 STRONG AND WEAK TIES

From among various relationships that a person forms, some of these relationships over time develop into close inter-personal bonds with shared interest and beliefs, while some remain only acquaintances. The close bonds can be said to possess **stronger ties**, while the acquaintances be **weaker ties**. Strong ties can be thought of as a source of confidence and emotional dependency, whereas weak ties are primarily acquaintances.

This dynamics of the strength of ties and how it influences the process of information diffusion in a network was first described in a famous study by Granovetter (1973). In his work, Granovetter empirically indicated that if A and B share a strong tie and A and C share a strong tie, then it is highly likely that owing to the overlapping similarity of interest, C and B will eventually develop stronger ties as well. This phenomenon is also known as **triadic closure**.

#### 3.3.1 Strength of a Tie

Figuratively, the strength of ties captures a sense of closeness among people. Empirically, the strength of ties between two nodes can be determined by how closely knit their circle of mutual friends are. In the simplest form, this can be captured via the **Jaccard score**.

The **neighbourhood overlap (NO)** of edges of nodes x and y is given by:

$$NO(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

where  $\Gamma(x)$  represents the set of neighbours of node x.

The numerator captures the number of common neighbours between x and y, whereas the denominator captures the total number of neighbours present in the neighbourhood of at least x or y. The higher the overlap, the higher the chance of x and y forming a link (if not formed yet).

#### Example 4.3

Based on Figure 4.5, let us calculate the neighbourhood overlap of (A,M) and (A,E).

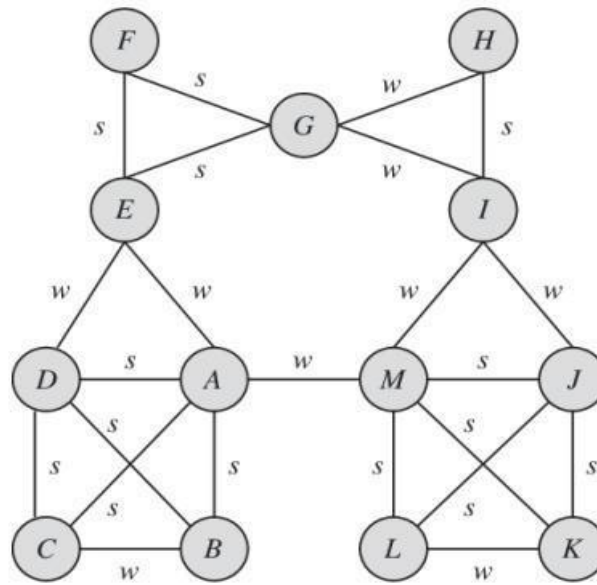


FIGURE 4.5. A network with 13 nodes and 23 edges, depicting strong (s) and weak (w) ties between the nodes.

$$\begin{aligned}
 \Gamma(A) &= \{B, C, D, E, M\}, \Gamma(M) = \{A, I, J, K, L\}, \Gamma(E) = \{A, D, F, G\} \\
 |\Gamma(A) \cap \Gamma(M)| &= |\{\phi\}| = 0 \text{ and } |\Gamma(A) \cap \Gamma(E)| = |\{D\}| = 1 \\
 |\Gamma(A) \cup \Gamma(M)| &= |\{A, B, C, D, E, M, I, J, K, L\} - \{A, M\}| = 8 \\
 |\Gamma(A) \cup \Gamma(E)| &= |\{A, B, C, D, E, F, G, M\} - \{A, E\}| = 6 \\
 \text{Thus, } NO(A, M) &= \frac{0}{8} = 0 \text{ and } NO(A, E) = \frac{1}{6}.
 \end{aligned}$$

FIGURE 4.5. A network with 13 nodes and 23 edges, depicting strong (s) and weak (w) ties between the nodes.

### 3.3.2 Triadic Closure

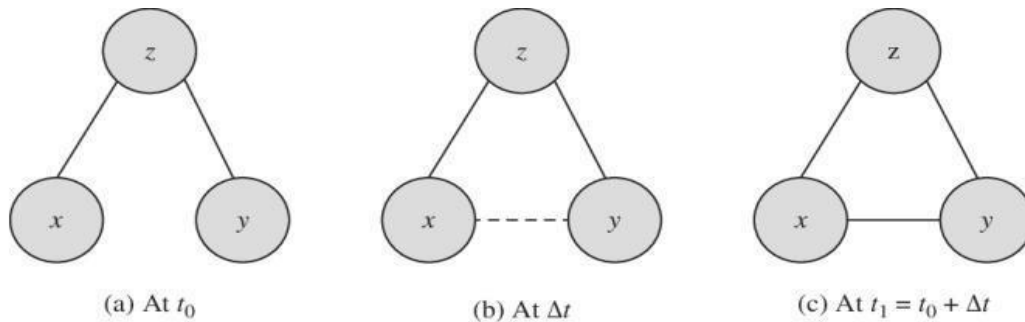
In layman terms, triadic closure states that a friend of a friend is also a friend. Within a network  $G(V, E_{t_0})$ , at timestamp  $t_0$ , if two people B and C are not friends with each other, but they both share a mutual friend A, then there is an increased likelihood of B and C becoming friends in the near future,  $t_i > t_0$ .

In other words, if  $\langle x, y \rangle \notin E_{t_0}$ ,  $\langle x, z \rangle \in E_{t_0}$ ,  $\langle y, z \rangle \in E_{t_0}$ , then it is highly likely that at time  $t_i > t_0$ ,  $\langle x, y \rangle \in E_{t_i}$ . This phenomenon is termed **triadic closure**. Figure 4.6 shows the phenomenon of triadic closure.

$$\Gamma(A) = \{B, C, D, E, M\}, \Gamma(M) = \{A, I, J, K, L\}, \Gamma(E) = \{A, D, F, G\}$$

$$\begin{aligned}
|\Gamma(A) \cap \Gamma(M)| &= |\{\phi\}| = 0 \text{ and } |\Gamma(A) \cap \Gamma(E)| = |\{D\}| = 1 \\
|\Gamma(A) \cup \Gamma(M)| &= |\{A, B, C, D, E, M, I, J, K, L\} - \{A, M\}| = 8 \\
|\Gamma(A) \cup \Gamma(E)| &= |\{A, B, C, D, E, F, G, M\} - \{A, E\}| = 6
\end{aligned}$$

$$\text{Thus, } NO(A, M) = \frac{0}{8} = 0 \text{ and } NO(A, E) = \frac{1}{6}$$



**FIGURE 4.6.** The phenomenon of triadic closure. (a) Nodes x and y are mutual friends with z at  $t_0$ . (b) Owing to a mutual friend, x and y get introduced to each other (dotted line) at  $\Delta t$ . (c) At  $t_1 = t_0 + \Delta t$ , x and y have formed a connection leading to a triadic closure.\*

There are many reasons behind the formation of a triadic closure:

1. **Opportunity:** Every mutual friend like z between x and y gives them an opportunity to meet and become acquainted with each other, thereby increasing the chances of developing a connection.
2. **Trust:** In a network, if connections are formed based on mutual trust, then x is more likely to trust y, as y is already trusted by z, provided x trusts z.
3. **Incentive:** z has an incentive to bring its mutual friends together in order to increase interpersonal trust and cooperation among the peers, thereby increasing the likelihood of x and y forming a connection.

We can quantify the strength of a triadic closure with respect to node z and two mutual friends x and y based on the **clustering coefficient** of z. The clustering coefficient of a node (as discussed in Chapter 2), in turn, measures the probability of two mutual friends of the given node being friends of each other.

$$CC_z = \frac{\sum_{i,j \in \Gamma(z)} \mathbb{I}[(i,j) \in E]}{k_z(k_z - 1)/2}$$

where  $k_z$  is the degree of node  $z$ , the denominator indicates the total possible links that can be formed among the neighbours of  $z$ , and  $\Gamma(z)$  represents the neighbourhood of  $z$ . Here, we assume that  $\langle i, j \rangle = \langle j, i \rangle$  (the undirected network). For every node pair  $(i, j)$  that belongs to the neighbourhood of  $z$ , we determine whether a link between them already exists or not, i.e.,  $(i, j) \in E$ . The Indicator function  $\mathbb{I}[\cdot]$  returns 1 if the condition specified is true, and 0 otherwise.

Overall,  $CC_z$  gives us the fraction of pairs of  $z$ 's neighbours/friends that are already friends. Being a probability,  $CC_z$  gives us a range of likelihood of two mutual friends  $(x, y)$  of  $z$  becoming friends in the future — 0 (none of  $z$ 's friends are friends with each other) to 1 (all of  $z$ 's friends are friends with each other).

#### Example 4.4

Based on the network in Figure 4.5 and Equation 4.2, let us determine how likely nodes B and M are to form a link, leading to a triadic closure among A, B, and M. With  $\Gamma(A) = \{B, C, D, E, M\}$  and  $k_a = 5$ , we have  $\langle B, C \rangle$ ,  $\langle B, D \rangle$ ,  $\langle C, D \rangle$ , and  $\langle D, E \rangle$  as the valid edges. Thus,  $CC_A = \frac{4}{(5 \times 4)/2} = 0.4$ ; i.e., there is a 40% chance of B and M becoming friends in the future.

In addition, the **Strong Triadic Closure Property** states that if node  $z$  has strong ties with two of its neighbours  $x$  and  $y$ , then  $x$  and  $y$  must develop at least a weak tie between them. In Figure 4.5, we see that node G has strong ties with both nodes E and F, and we would expect  $\langle E, F \rangle$  to be at least a weak tie (it is a strong tie in our example).

#### 3.3.3 Dunbar Number

Another cognitive study (Dunbar 1992) indirectly supports strong and weak ties in the real world. It proposes the idea that depending on the short-term memory of the human brain, people are constrained in terms of the strong ties they can maintain at any given point in time, formalised as the **Dunbar Number**. This number, which empirically ranges from 100 to 250, represents the set of people one can be in close contact with. For general purpose, the Dunbar number is set at 150.

The study further highlighted that the overall set of social interactions of a person can be way more than 250, but there is a limit to close ties. This is quite evident online as well, where a person can have millions of followers but only count a few of them as close associates.

### 3.3.4 Local Bridges and Importance of Weak Ties

An edge is considered a **bridge**, if its removal increases the number of disconnected components in a network. A **local bridge** has a more lenient definition.

#### Local Bridge

A local bridge is an edge  $\langle i, j \rangle$  between nodes  $i$  and  $j$  such that upon its removal, there will be no edge between  $i$ 's neighbour and  $j$ 's, even though there could be indirect connections among them.

An edge can be considered a local bridge if its NO score is 0; i.e., the endpoints of the edge have no mutual friends. In addition, an edge that is considered a local bridge is not a part of any triad.

Based on this definition and Example 3.1 where  $NO(A, M) = 0$ , we can say that in the network of Figure 4.5, edge  $\langle A, M \rangle$  should be a weak tie, which it is.

As local bridges serve as points of distributing information from one section of the network to the other, weak ties can act as local bridges, or local bridges can be considered weak ties between the endpoint nodes of the edge.

#### Importance of Local Bridges/Weak Ties

1. Granovetter (1973) highlighted the importance of weak ties for information diffusion. Imagine a scenario where you want to switch jobs and join another organisation. While the closest associates have your best interest in mind, they will not help you score diverse offers due to similar or even the same information circulating in the close groups. Here, a weaker tie, an acquaintance working in another organisation or group, will bring newer and more diverse referrals. Thus, within the strong ties, a person is more likely to receive information on same topics/beliefs, whereas the weaker ties bring in diversity of opinion and information. Such weaker ties act as a bridge and help a person gain access to newer and wider information. The importance of weaker ties is not just limited to the access of information but also access to opportunities (as in the example of new job referrals). This phenomenon is known as the **strength of weak ties**.
2. In case of stress/conflict between the two groups, weak ties act as mediators and help the other side understand a newer perspective.
3. In an adversarial setting, removing local bridges can lead to the formation of echo chambers, and their absence can increase the disinformation gap.

Unfortunately, acting as a bridge, in case of disease outbreaks, local bridges may cause the disease to transmit from one group to another.

Another way of looking at local bridges and weak ties is to use the concept of **embeddedness** of an edge.

**Edge embeddedness.** For an edge  $\langle x, y \rangle$ , its embeddedness can be defined as the number of mutual friends that the endpoints of the edge possess; i.e.,

$$\text{Embeddedness}(\langle x, y \rangle) = |\Gamma(x) \cap \Gamma(y)|$$

where  $\Gamma(x)$  is the set of neighbours of node  $x$ .

Thus, by definition, a local bridge is an edge with embeddedness of zero. A node with lesser embeddedness as compared to its neighbours is more likely a bridge and less likely to be safeguarded by the presence of mutual neighbours in the network. On the other hand, this risk is a trade-off for having access to novel information owing to residing in multiple groups.

### 3.4 LINK ANALYSIS ALGORITHMS

While a majority of studies in link analysis is related to ranking web pages (Kleinberg 1999; Page et al. 1999; Mei et al. 2010), some are related to determining the similarity between objects (Jeh and Widom 2002) (in both homogeneous and bipartite heterogeneous network settings). Other studies exclusively focus on heterogeneous systems (Lee et al. 2012). In the following sections, we shall discuss in detail various algorithms for link analysis with varying aims and applications.

On a side note, we shall see that the general-purpose algorithms (PageRank or DivRank) are not incremental; any modification in the input network would require a complete re-computation of ranks. To alleviate this problem with respect to new edges being added to the system, a fast incremental PageRank approximation method was proposed by Bahmani et al. (2010). Since then, variants of PageRank have been proposed to deal with different shortcomings and enhancements with respect to the original solution.

### 3.5 PAGERANK

Invented by the founders of Google, PageRank (Page et al. 1999) was initially proposed as a method of producing a global ranking of all web pages on the web. The method has been subsequently used to rank authors and papers in a citation network, proteins in a

metabolic network (Jiang et al. 2017), and even sentences in documents for text summarisation (Mihalcea and Tarau 2004).

The basic idea behind PageRank is that a node's importance is determined by the incoming edges it receives from other important nodes in the system.

### 3.5.1 Intuition

Imagine a series of blogs. The first blog ends with a line that says "Go to the second blog to continue reading." While you move to the second page, you again notice that the second blog starts with a line that says "Go to the first blog to read the introduction."

This outgoing hyperlink from the first blog to the second is termed an **out-edge** or **forward link**. Similarly, the incoming hyperlink to the first blog from the second blog is termed an **in-edge** or **backward link**.

In the World Wide Web, with every forward link a page establishes, it transfers some of its importance/rank influence to the forward page. Thus, if a highly important node points to a lesser important one, there is an enhancement in the status of the latter node, with the importance of each node being determined by its in-edges.

### Simple Ranking

For a node  $w$  with rank  $R(w)$ , let  $F_w$  be the set of nodes that  $w$  points to (Forward links) and  $B_w$  be the set of nodes that point to  $w$  (Backward links). Let  $N_w = |F_w|$  be the number of forward links. Then, the simple ranking of  $w$ , denoted by  $R(w)$ , is given as:

$$R(w) = \sum_{b \in B_w} \frac{R(b)}{N_b} \quad (4.3)$$

The aforementioned definition assumes that the collection of web pages is a connected component. However, there could be pages that do not refer to any other page or are not referred by any other page. Imagine a scenario where no hyperlinks exist. In this case, each page will be equally (un)important with a uniform rank of:

$$R(p) = \frac{1}{\text{Total number of pages}}$$

### Example 4.5

Using the network in Figure 4.7, let us try to determine the PageRank scores of nodes using Equation 4.3. We see that for a node,  $R(v) = \sum_{b \in B_v} R(b)$ , where  $R(v)$  is the simple PageRank score for  $v$ . Initially, we assume that each node has a PageRank score of 1.

#### Example 4.5

Using the network in Figure 4.7, let us try to determine the PageRank scores of nodes using Equation 4.3. We see that for a node  $R(v) = \sum_{b \in B_v} R(b)$ , where  $R(v)$  is the simple PageRank score for  $v$ . Initially, we assume that each node has PageRank score of 1.

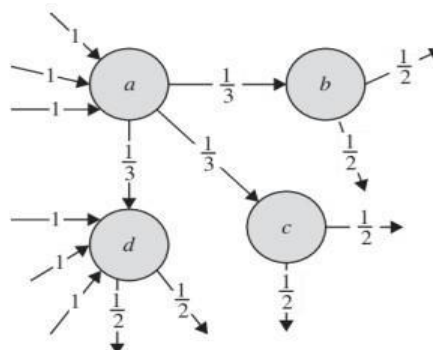


FIGURE 4.7. A network with the PageRank of each node initialised as 1, where the weight of each out-link from node  $v$  is normalised by  $|F_v|$ .

FIGURE 4.7. A network with the PageRank of each node initialised as 1, where the weight of each out-link from node  $v$  is normalised by  $|F_v|$ .\*

$$R(a) = 1 + 1 + 1 = 3, R(b) = \frac{1}{3}, R(c) = \frac{1}{3}, R(d) = \frac{1}{3} + 1 + 1 + 1 = \frac{10}{3}$$

$$PageRank_{raw} = \left[ 3, \frac{1}{3}, \frac{1}{3}, \frac{10}{3} \right], Sum(PageRank_{raw}) = 3 + \frac{1}{3} + \frac{1}{3} + \frac{10}{3} = 7$$

$$PageRank_{normalised} = \frac{PageRank_{raw}}{Sum(PageRank_{raw})},$$

$$PageRank_{normalised} = [0.4, 0.05, 0.05, 0.5]$$

$$Sum(PageRank_{normalised}) = 0.4 + 0.05 + 0.05 + 0.5 = 1$$

$$R(a) = 1 + 1 + 1 = 3, R(b) = \frac{1}{3}, R(c) = \frac{1}{3}, R(d) = \frac{1}{3} + 1 + 1 + 1 = \frac{10}{3}$$

$$PageRank_{raw} = \left[ 3, \frac{1}{3}, \frac{1}{3}, \frac{10}{3} \right], Sum(PageRank_{raw}) = 3 + \frac{1}{3} + \frac{1}{3} + \frac{10}{3} = 7$$

$$PageRank_{normalised} = \frac{PageRank_{raw}}{Sum(PageRank_{raw})}, PageRank_{normalised} = [0.4, 0.05, 0.05, 0.5]$$

$$Sum(PageRank_{normalised}) = 0.4 + 0.05 + 0.05 + 0.5 = 1$$

### 3.5.2 Random Walk

Consider a random surfer browsing the web, who starts at a random page  $P_1$ . The surfer follows a hyperlink of  $P_1$  and moves to subsequent pages  $P_2, P_3, \dots, P_m$ . Upon reaching  $P_m$ , the surfer finds no hyperlinks to jump to (dead-end). The surfer then decides to randomly leap to any one of the web pages, each having an equal probability of the surfer landing on that page. Once on a new page, the surfer continues its journey.

In fact, upon landing on a page  $P_i$ , the surfer can choose from either of the following two actions:

- With a probability  $\alpha$ , it can choose to randomly jump to another page and forego the hyperlinks. This action of random jumping can be denoted as  $E = \frac{1}{N}$ , where  $N$  is the total number of web pages in the network.
- With a probability  $1 - \alpha$ , it can stay the course of following subsequent hyperlinks.

Translating random walks to node prestige, one can say that the more number of times a node is visited during random walking, the higher the importance of the node. Imagine a junction on the road that needs to be crossed to enter different parts of the city. Of all roads in the city, this junction might be the most important.

Combining this analogy with the probabilities discussed here, the PageRank formulation can be represented in terms of a probability score of web page  $w$  as:

$$R(w) = (1 - \alpha) \sum_{b \in B_w} \frac{R(b)}{N_b} + \alpha E \Rightarrow R(w) = (1 - \alpha) \sum_{b \in B_w} \frac{R(b)}{N_b} + \alpha \frac{1}{N} \quad (4.4)$$

We shall come to the definition of  $\alpha$  shortly. As of now, let us consider this as a parameter which controls the balance between the importance of the two components of the formulation.

$$\sum_{i=1}^N R(i) = 1$$

Thus, the PageRank of a page is the probability of that page being visited during a random walk.

The updated random walk ranking formula in Equation 4.4 overcomes two drawbacks of the simple ranking mentioned in Equation 4.3.

At first, as discussed earlier, when the surfer reaches a dead-end, it has the ability to randomly jump to other pages. If this functionality is not provided, the surfer will remain stuck even when there are pages left to explore.

Next, consider a system where page A hyperlinks to B, and B hyperlinks to A or B itself. If A and B do not hyperlink to another page, then without the ability of random jumps,

the surfer can remain stuck in the loop, and will not be able to explore other pages on the system. This is called a **spider trap**.

As evident from Equation 4.4, the overall rank of page  $w$  is updated if the ranking of one of the pages  $b \in B_w$  is updated. Subsequently, a rank update of  $w$  causes an update in the ranking of all the pages  $f \in F_w$ , making PageRank a **recursive method**. In the following sections, we shall study how to formalise the convergence of this recursive method.

### 3.5.3 Matrix Manipulation and Convergence of PageRank

Let us consider a system of  $N$  web pages where the PageRank vector is of the form  $R = \langle r_1, r_2, \dots, r_n \rangle$ , and  $0 \leq r_i \leq 1$ , with the initial set  $R$  being  $r_1 = r_2 = \dots = r_n = \frac{1}{N}$ .

The hyperlinked web pages can be represented as nodes of a directed network with edges representing the direction of the link. With  $A$  initiated as an adjacency matrix, we have  $A_{ij} = 1$  if there is an out-edge from  $i$  to  $j$ , and  $A_{ij} = 0$  otherwise.

Since, for the PageRank calculation, the out-edge is weighted by the total links from that node,  $A$  is normalised such that:

$$A_{ij} = \frac{1}{N_i}, \text{ where } N_i = |F_i| \quad (4.5)$$

The normalised  $A$  is thus a **stochastic matrix**.

For the first iteration, we have PageRank scores as  $R_0$ . Then at the next iteration, we have  $R_1 = R_0 A$ , and so on. This can be written as:

$$R_{i+1} = R_i A \quad (4.6)$$

Again, Equation 4.6 only accounts for jumps where  $A_{ij} \neq 0$ . Let  $E = \langle e_1, e_2, \dots, e_n \rangle$  be a vector of length  $N$ , which represents the probability of random jumps and is formalised as  $e_1 = e_2 = \dots = e_n = \frac{1}{N}$ .

Again, with  $\alpha$  representing the uniform probability of performing a random jump, our rank iteration can be modified as:

$$R_{i+1} = (1 - \alpha)R_i A + \alpha E \quad (4.7)$$

The parameter  $\alpha$  is commonly expressed using the **damping factor**,  $d = 1 - \alpha$ , where a damping factor of 0.70 would mean that the random surfer continues exploring hyperlinks with probability 0.70, and makes a random jump with probability 0.30. Various studies have tested different damping factors, but it is generally assumed that the damping factor will be set around **0.85**.

Also, note that the condition in Equation 4.5 (i.e.,  $0 \leq r_i \leq 1$  and the normalisation) holds for each iteration. Thus, at each iteration, the PageRank vector  $R_i$  represents the probability distribution of prestige/importance of a system of web pages.

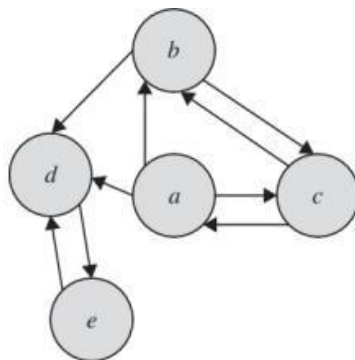
Ideally, the aforementioned system should converge when  $R_{i+1} = R_i$ . Accounting for the system's error and precision, with a tolerance of  $\epsilon$ , we say that the aforementioned system of equations converges if:

$$|R_{i+1} - R_i| \leq \epsilon$$

### Example 4.6

Using the toy network in Figure 4.8, and the iterative PageRank in Equation 4.7, let us determine the first iteration of ranks, assuming the damping factor,  $d = 0.8$ .

In the absence of any prior knowledge, let us assume that the initial ranking of each node is the same. Thus,  $R_0 = [1/5, 1/5, 1/5, 1/5, 1/5]$ . Also, let us assume that each node has an equal likelihood of being randomly visited, with  $E = [1/5, 1/5, 1/5, 1/5, 1/5]$ .



**FIGURE 4.8.** A sample network with the backward and forward edges marking the hyperlinks among the nodes.

**FIGURE 4.8.** A sample network with the backward and forward edges marking the hyperlinks among the nodes.

We get the adjacency matrix as:

$$A = \begin{array}{c|ccccc} & a & b & c & d & e \\ \hline a & 0 & 1 & 1 & 1 & 0 \\ b & 0 & 0 & 1 & 1 & 0 \\ c & 1 & 1 & 0 & 0 & 0 \\ d & 0 & 0 & 0 & 0 & 1 \\ e & 0 & 0 & 0 & 1 & 0 \end{array}$$

The normalised matrix is:

$$A = \begin{array}{c|ccccc} & a & b & c & d & e \\ \hline a & 0 & 0.33 & 0.33 & 0.33 & 0 \\ b & 0 & 0 & 0.5 & 0.5 & 0 \\ c & 0.5 & 0.5 & 0 & 0 & 0 \\ d & 0 & 0 & 0 & 0 & 1 \\ e & 0 & 0 & 0 & 1 & 0 \end{array} \quad (4.8)$$

### 3.6 PERSONALISED PAGERANK

In Equation 4.7, we noticed that vector  $\mathbf{E}$  is used as a mechanism to provide the randomness in the system, and avoid dead-ends and looping. While for a generalised system of ranking,  $\mathbf{E}$  is set as a uniform distribution, it can be modified to focus on one or more relevant pages, leading to a system of **Personalised PageRank**.

Consider the case where after surfing through a website, the random surfer ultimately returns to the index page of the website, or after performing some search queries the user returns to the landing page of the search engine. Under both scenarios, the chance of returning to the landing page is higher than that of returning to a random web page. If this landing page is represented as  $\mathbf{l}$ , then we can have  $e_l = 1$  and  $e_{i \neq l} = 0$ , or  $e_l \gg e_{i \neq l}$ , and our random distribution vector  $\mathbf{E}$  will be skewed towards the page  $\mathbf{l}$ .

Now, let us consider another scenario where instead of returning to a single page, the surfer can return to one of the pages from among a set of pages. This set, denoted by  $\mathbf{S}$ , can be obtained as a list of the recent pages or most clicked pages in the user's history. One can further extend this idea to make use of the textual content of the pages to obtain a list of pages, denoted by  $S_w$ , which are most similar (may or may not be directly connected) to the current page under consideration.

In any of the aforementioned scenarios, we obtain a set  $\mathbf{S}$  or  $S_w$  of pages which skews the distribution of  $\mathbf{E}$ . Equation 4.4 can be modified to capture this as:

$$R(w) = (1 - \alpha) \sum_{b \in B_w} \frac{R(b)}{N_b} + \alpha E(S_w) \quad (4.9)$$

Making use of Equation 4.9, we can obtain customised ranking for a user (a page), leading to the formation of **personalised ranks** (or **topic-sensitive ranks**). In the NetworkX library, such personalisation is supported as an optional functional parameter in the main PageRank algorithm.

### 3.6.1 Time Homogeneous Random Walks and Stationary Distribution

As evident from Equations 4.4 and 4.8, the value of matrix **A** remains constant throughout the iterations. Such a family of random walks where the transition probabilities remain constant over time is termed **time-homogeneous random walks**.

In addition, random walks can be represented as **Markov chains** of the given network. Each page/vertex represents a state, and the random walk is a series of transitions from one state to another. The edges of the transition are marked by the pre-computed transition probability of moving from one state to another.

For a network of **N** web pages (as used in PageRank),  $p_0: N \times N \rightarrow [0, 1]$  denotes the pre-computed transition probabilities. These probabilities can be estimated as:

$$p_0(u, v) = (1 - d) \cdot p^*(v) + d \cdot \frac{w(u, v)}{\sum_{b \in F_u} w(u, b)} \quad (4.10)$$

where **d** is the damping factor, and  $w(u, v)$  is the weight of the outgoing edge  $\langle u, v \rangle$  present in the out-link set  $F_u$ . If no edge exists between **u** and **v**, then  $w(u, v) = 0$ . Here,  $p^*(v)$  represents the prior distribution of the vector **E**, inclusive of the case of personalised ranking.

$$\sum_{v \in V} p^*(v) = 1 \text{ and, for uniform distribution, } p^*(v) = \frac{1}{N} \quad \forall v \in V$$

If  $p_T(u)$  denotes the probability of the random surfer being at state (page) **u** at time (iteration) **T**, then using Equation 4.10, the probability of reaching state **v** from one of its inward neighbours  $u \in B_v$  at time  $T + 1$ , can be given as:

$$p_{T+1}(v) = \sum_{u \in B_v} (1 - d) \cdot p^*(v) + d \cdot p_T(u) \cdot \frac{w(u, v)}{\sum_{b \in F_u} w(u, b)} \quad (4.11)$$

Since for time-homogeneous random walks,  $p_t(u, v) = p_0(u, v)$ , Equation 4.11 can also be represented as:

$$p_{T+1}(v) = \sum_{u \in B_v} p_0(u, v) p_T(u) \quad (4.12)$$

Thus, the Markov chain converges when  $\forall v \in V$ ,  $p_{T+1}(v) \approx p_T(v)$ , with a score  $\pi(v)$  such that  $\sum_{v \in V} \pi(v) = 1$ .  $\pi(v)$  measures the **prestige** of vertex  $v$ .

The final unique solution  $\pi$  to which the distribution of ranks converges is termed the **stationary distribution** of the Markov chain, and is the desired output of PageRank. In other words, over a large number of iterations, the stationary distribution of a node is related to the amount of time (or fraction of time) a random surfer spends at the node.

It can be observed that this interpretation of time-homogeneous random walks and convergence to a stationary distribution can be computed via the matrix manipulation technique

#### Example 4.6 (continued)

$$\begin{aligned} R_0 \times A &= [0.1, 0.1666, 0.1666, 0.3666, 0.2] \\ d(R_0 \times A) &= 0.8 \times [0.1, 0.167, 0.167, 0.367, 0.2] \\ &= [0.08, 0.1336, 0.1336, 0.2936, 0.16] \\ (1 - d)E &= 0.2 \times [0.2, 0.2, 0.2, 0.2, 0.2] = [0.04, 0.04, 0.04, 0.04, 0.04] \end{aligned}$$

Since  $R_1 = d(R_0 \times A) + (1 - d)E$ :

$$\begin{aligned} R_1 &= [0.08, 0.1336, 0.1336, 0.2936, 0.16] + [0.04, 0.04, 0.04, 0.04, 0.04] \\ R_1 &= [0.12, 0.1736, 0.1736, 0.3336, 0.2] \\ \text{Sum}(R_1) &= 1.0008 \approx 1 \end{aligned}$$

### 3.6.2 Advantages and Disadvantages of PageRank

Some advantages of PageRank are as follows:

1. The vectorised system of equations is fast to compute.

2. Equation 4.7 is guaranteed to converge to a unique solution.
3. As the ranking is independent of the query (or any node), the ranks can be pre-computed during indexing and re-used during query time.
4. As the in-edges with respect to a page are harder to manipulate than out-edges, the ranks are robust and stable.
5. The intuition of a lesser-known entity gaining importance by interacting with a known entity is evident in various social interactions, such as the ranking of players in Chess or Tennis.

**Some disadvantages of PageRank are as follows:**

1. The PageRank score takes into account the connections of a node, and not the content of a page. Thus, it is prone to spamming.
2. A page can get a high rank by either connecting to a small number of important pages or a lot of trivial pages to achieve the same effect. This again can lead to spamming.
3. While a ranking system built on the basis of in-edges is more robust than that of out-edges, it is still prone to manipulation in the form of buying its way into hyperlinking with important pages.

The intuitive system of ranking opened avenues for more enhanced ranking metrics for the web as well as for other inter-connected systems such as traffic management, biology, and knowledge graphs. While it is not the first system to exploit network structure and state for a ranking metric (refer to Franceschet (2010) for the background used in PageRank), it has grown to become one of the most commonly used metrics.

It is important to note that the ranking system discussed earlier is a **static system**, with the assumption that no modification in the adjacency matrix occurs for the duration of indexing as well as querying. Any modification requires all ranks to be re-computed again.

**Notation key:**

- $p_0(u, v)$  = pre-computed transition probability from  $u$  to  $v$
- $p_T(u)$  = probability of being at state  $u$  at iteration  $T$
- $\pi(v)$  = stationary distribution / prestige score of vertex  $v$

- $w(u, v)$  = weight of edge from  $u$  to  $v$
- $p^*(v)$  = prior distribution (for personalised ranking)

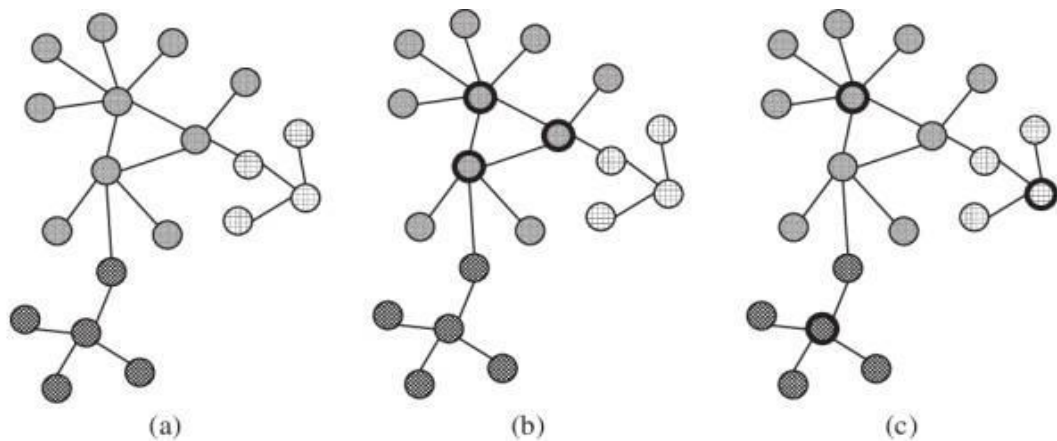
### 3.7 DIVRANK

As discussed in the previous section, PageRank focuses solely on the prestige/connectivity of a page/node to determine its importance. Owing to this very factor of connectivity, it is highly likely that the top-ranking nodes are all related to the same themes and carry similar (or even redundant) information. In fact, if one vertex is visited frequently by the random surfer, then all its neighbours are also likely to be visited frequently on the path, thereby becoming more important/prestigious. This is termed **smoothing** or **regularisation** of ranking scores. But owing to this smoothing, the top-ranked results are not diverse.

As birds of a feather flock together, a few high-degree prestigious nodes contribute to boosting the scores of their neighbours, and all the top-ranked nodes tend to be occupied by neighbouring vertices (likely to carry similar information).

Take an example of a user requesting a list of famous eateries in the city. If all the places in the top rank are occupied by chicken eateries but the user is a vegetarian, then the top-ranked results are not of much relevance to the user. A result consisting of places that are famous for different cuisines (even if they are not as famous as the top restaurants) will be more relevant to the user (provided that the user has a more diverse choice rather than restricting the world-view!).

As the time-homogeneous random walk system, depending on the connectivity measure, does not promote diversity, in order to shift the focus from smoothing/cooperation among neighbours, one needs to add an element of "**competition**." By competing with its neighbours, the rich gets richer over time and absorbs the scores of its neighbours. This idea is explored in **DivRank (Diverse Rank)** (Mei et al. 2010), which combines prestige and diversity (as opposed to only prestige in PageRank), leading to more diverse and relevant results.



**FIGURE 4.9.** Intuition behind DivRank. (a) The input network consists of three groups of nodes (marked by three different patterns). (b) PageRank scores (larger the size of the node, higher the rank) based on time-homogeneous random walks weighted by degree distribution. Nodes (with the highest degree) and their neighbours forming a triadic closure, are the top three nodes (marked by thicker borders). This is due to higher degree and smoothing. The top three nodes all belong to the same group. (c) DivRank scores based on vertex-reinforced random walks weighted by the degree distribution. Nodes with the highest degree in each group end up as top three ranked nodes (nodes with thicker boarder). Owing to the competition and self-looping, the central node in each group absorbs the rank from its neighbourhoods and reduces their ranks (shown as a change in size of the nodes), in turn leading to a competitive diversified ranking. The figure is adopted from [Mei et al. \(2010\)](#).

### 3.7.1 Vertex-Reinforced Random Walks

While the time-homogeneous random walk assumes that the transition probabilities remain static over time, it does not fully capture real-world interactions. For example, a person is more likely to comment on a post made by a friend than by a stranger; in turn, the user's timeline will receive more information about her friends than a random update. A museum is more likely to be visited if it has been visited and recommended by a previous visitor. A product is more likely to be bought if it is trending, and more purchases keep the trend active.

Thus, over time, the probability of coming across certain entities (people, places, products, etc.) increases. Therefore, transition probabilities are updated with time, and the family of random walks so obtained is termed a **time-variant random walk process**.

One way of achieving time-varying transition probabilities is via **vertex-reinforced random walks (VRRW)**.

#### VRRW

When the transition probability from one state to the next,  $p_T(u \rightarrow v) \rightarrow p_{T+1}(u \rightarrow v)$ , is reinforced by the number of previous visits to the state  $N_T(v)$ , i.e.,

$$p_T(u \rightarrow v) \propto p_0(u \rightarrow v) \cdot N_T(v)$$

then the process is known as the **VRRW**.

### 3.7.2 Random Walk Formulation in DivRank

As discussed in Section 4.6.1,  $p^*(v)$  refers to the prior distribution of visiting vertex  $v$ ;  $p_0(u, v)$  is the organic and pre-computed transition probability obtained from the network topology before the random walk starts.

In DivRank, the organic probability is captured via:

$$p_0(u, v) = \begin{cases} \alpha \frac{w(u, v)}{\sum_{b \in F_u} w(u, b)}, & u \neq v \\ 1 - \alpha, & u = v \end{cases} \quad (4.11)$$

Here,  $\alpha$  captures the probability that the random walk will follow one of the neighbours, or choose to stay at the current state/node. At a given timestamp, there is a chance that the surfer stays at the node, the probability of which is reinforced by the number of previous visits at the current node.

The overall transition probability,  $p_T(u, v)$ , employing the damping factor  $d$ , is thus defined as:

$$p_T(u, v) = (1 - d)p^*(v) + d \cdot \frac{p_0(u, v) \cdot N_T(v)}{\sum_{b \in F_u} p_0(u, b) \cdot N_T(b)} \quad (4.12)$$

The overall probability of the random surfer moving to node  $v$  at time  $T + 1$ , from one of its neighbours  $B_v$ , can be obtained as a summation over Equation 4.12, subject to the probability of the surfer being in the neighbourhood at time  $T$ :

$$p_{T+1}(v) = (1 - d)p^*(v) + \sum_{u \in B_v} d \cdot p_T(u) \cdot \frac{p_0(u, v) \cdot N_T(v)}{\sum_{b \in F_u} p_0(u, b) \cdot N_T(b)}$$

Initially,  $N_0(1) = N_0(2) = \dots = N_0(n)$ . Thus, each vertex is equally favoured. Since  $p_0(u, v)$  is expressed in terms of the degree of a node, the transition probability starts favouring nodes with higher degree. Over time, due to the reinforcement factor and self-

looping, the transition probability of higher-degree nodes gets favoured, skewing the ranking in favour of "rich gets richer."

Empirically, these high-degree nodes are spread across different communities/clusters, and thus the result of DivRank ends up being more diverse.

#### Notation key added in this section:

- $N_T(v)$  = number of previous visits to vertex  $v$  up to time  $T$
- $\propto$  = "proportional to"
- $\alpha$  = probability of following a neighbour vs. self-looping (staying at current node)
- $p_0(u, v)$  = organic/pre-computed transition probability (piecewise: self-loop vs. neighbour case)
- Piecewise ("cases") notation = different formulas depending on condition ( $u \neq v$  vs.  $u = v$ )

### 3.8 SIMRANK

While various metadata/features of an entity can be used to determine similarity, these features are domain-specific and hard to obtain in practice. In contrast, one can always derive contextual similarity from the network topology, which is more readily available.

Making use of this structural similarity, Jeh and Widom (2002) proposed **SimRank**, a general-purpose, domain-agnostic method for determining the level of similarity between two entities based on their shared inter-object relations. The SimRank score underpins the fact that two entities/nodes are similar if they are related to or referenced by similar entities, obtaining a similarity score based solely on the topological similarity of entities.

For a network with node set  $V$  consisting of  $n$  nodes, the similarity score is calculated for each pair of nodes  $(u, v)$ ,  $\forall (u, v) \in V \times V$ , generating  $n^2$  similarity scores per network. In contrast, measures such as PageRank or DivRank are computed for every node, generating a ranked list of  $n$  scores per network.

#### Example 4.7

Based on the citation network of Figure 4.10, we can say that papers  $P_3$  and  $P_4$  appear to be similar due to the fact that they are directly cited by the same paper  $P_5$ . Similarly, we

can also say that papers  $P_2$  and  $P_7$  could be similar based on the fact that they are directly cited by the same source  $P_8$ .

However, drawing such conclusions for the similarity between  $P_1$  and  $P_2$  is not easy, as they do not share a common neighbour:

$$\Gamma(P_1) = \{P_3, P_7, P_6\}, \Gamma(P_2) = \{P_8, P_4\}, \Gamma(P_1) \cap \Gamma(P_2) = \phi$$

However,  $P_8$  and  $P_5$  seem to be indirect sources of similarity, based on the fact that citations are transitive in nature — if, say,  $P_4$  cites  $P_3$ , and  $P_3$  cites  $P_1$ , then  $P_4$  indirectly cites  $P_1$ . While a neighbourhood-based citation similarity will not be able to capture this transitivity, with the help of SimRank, such indirect topological influences can be accounted for in calculating similarity between  $P_1$  and  $P_2$ .

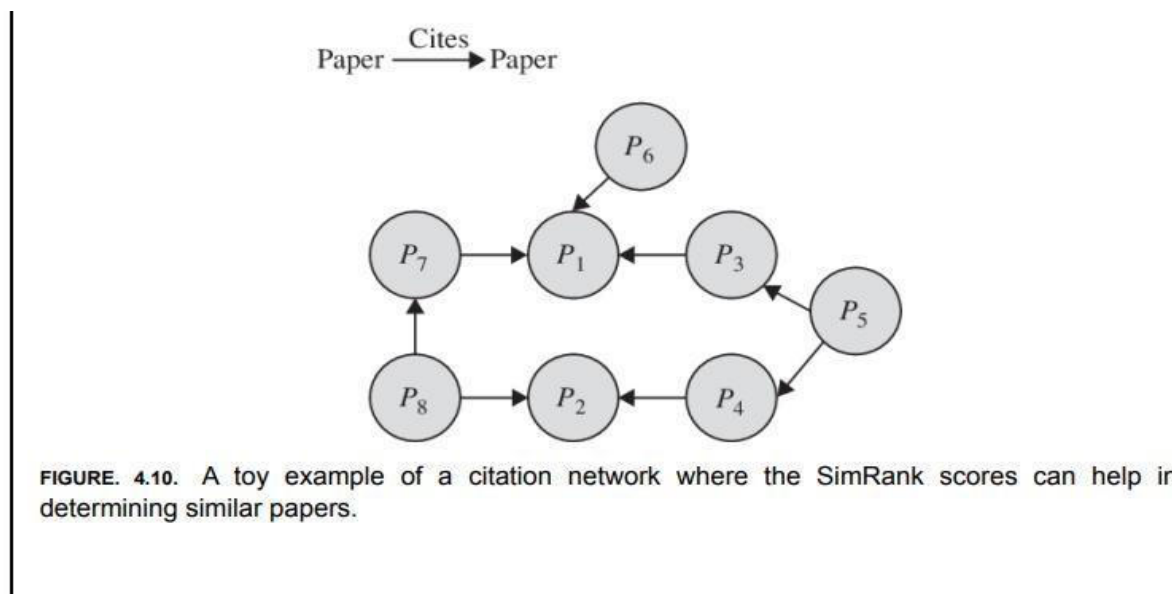


FIGURE. 4.10. A toy example of a citation network where the SimRank scores can help in determining similar papers.

### 3.8.1 Basic Formulation of SimRank

For a node  $\mathbf{v}$  in the network, we have  $I(\mathbf{v})$  and  $O(\mathbf{v})$  as the set of the in-degree and out-degree neighbours of  $\mathbf{v}$ , respectively. An individual in-degree neighbour can be represented as  $I_i(\mathbf{v})$  for  $1 \leq i \leq |I(\mathbf{v})|$ . Similarly, an individual out-degree neighbour can be represented as  $O_i(\mathbf{v})$ .

We consider  $s(\mathbf{u}, \mathbf{v})$  as the similarity score, ranging over  $[0, 1]$ , between two randomly selected nodes  $\mathbf{u}$  and  $\mathbf{v}$  of the network. Consequently, the similarity score can also be formulated as:

$$s(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } I(a) = \phi \text{ or } I(b) = \phi \\ \frac{C}{|I(a) \cup I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) & \text{otherwise} \end{cases} \quad (4)$$

Here,  $s(w, w) = 1$  becomes the base case similarity score, implying that the similarity of a singleton node with itself is maximum, i.e., 1 — a node is maximally similar to itself.

Since the similarity score depends on the neighbourhood of the nodes, there is no way of determining the score for a neighbourhood that does not exist. Hence, for the case when either  $I(a) = \phi$  or  $I(b) = \phi$ , we set the score to 0.

For the general case, we can say that the similarity between two randomly selected nodes, **a** and **b**, is proportional to the average similarity between their neighbours, while the denominator helps in normalisation.

In Equation 4.13, the parameter **C** acts as a **decaying factor**. Consider a scenario where two nodes **u** and **v** are referred to by node **w**. We then know that  $s(w, w) = 1$ , and by virtue of related similarity, nodes **u** and **v** are also similar with a score  $s(u, v) \propto s(w, w)$ , or  $s(u, v) = Cs(w, w)$ . As  $0 \leq s \leq 1$  and  $s(w, w) = 1$ , we know that for  $0 < C < 1$ , the similarity score of the 1-hop neighbours of **w** is somewhat lesser than 1.

Now, if both **u** and **v** refer to nodes  $p_1$  and  $p_2$ , then by transitivity of similarity among 2-hop neighbours,  $s(p_1, p_2) = Cs(u, v) = C^2s(w, w)$ .

Thus, as we move further away from **w**, the similarity among the  $n^{th}$ -hop neighbours decreases, with **C** providing the decaying factor. Figure 4.11 illustrates the same. This is intuitive to understand — the farther apart two users are in a network, the less likely they are to share similar interests.

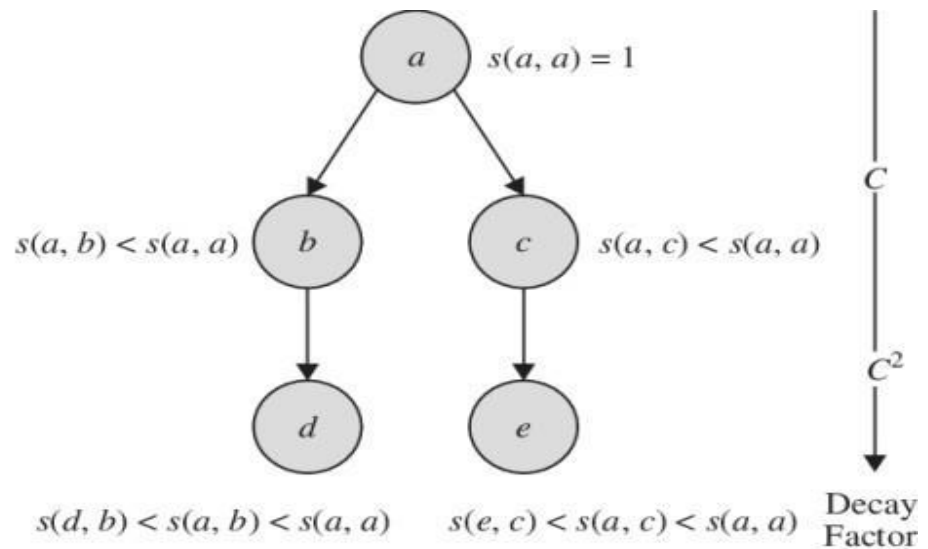


FIGURE 4.11. Impact of decay factor  $C$  on propagation of similarity scores along the edges.

### 3.8.2 Naive Solution to SimRank

For a network with a vertex set  $V$  of  $n$  nodes, at a given iteration  $k$ , we keep  $n^2$  scores  $R_k(a, b)$ , where the scores at iteration  $k + 1$  are computed from the scores of iteration  $k$ , with  $R_0(a, b)$  providing a lower bound:

$$R_0(a, b) = \begin{cases} 0 & \text{if } a \neq b \\ 1 & \text{if } a = b \end{cases} \quad (4.14)$$

and

$$R_{k+1}(a, b) = \begin{cases} 1 & \text{if } a = b \\ \frac{C}{|I(a)| |I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} R_k(I_i(a), I_j(b)) & \text{if } a \neq b \end{cases} \quad (4.15)$$

with  $\lim_{k \rightarrow \infty} R_k(a, b) = s(a, b)$ .

The rank metric is a **monotonically increasing** function, with  $0 \leq R_k(a, b) \leq R_{k+1}(a, b) \leq 1, \forall (a, b) \in V \times V, k \geq 0$ . At each iteration  $k$ , the scores are propagated forward along the direction of the edge, until the system converges.

### 3.8.3 SimRank in Heterogeneous Bipartite Network

Equations 4.13 and 4.15 can easily be extended to a bipartite setting consisting of two sets of objects. Consider a heterogeneous network of users and products, with the only set of bipartite edges directed from users to products if a user buys a product. Now, two users  $u_1$  and  $u_2$  can be considered similar if they buy similar products. Consequently, two products  $p_1$  and  $p_2$  can be considered similar if they are bought by similar users. Hence, the similarity of products and users are **mutually reinforced**.

Let  $s(u_1, u_2)$  denote the similarity between two users  $u_1$  and  $u_2$ , and let  $s(p_1, p_2)$  denote the similarity between two products  $p_1$  and  $p_2$ .

As the directed edges go from users to products, for two users  $u_1 \neq u_2$ , we can rewrite Equation 4.13 as:

$$s(u_1, u_2) = \frac{C_1}{|O(u_1) \cup O(u_2)|} \sum_{i=1}^{|O(u_1)|} \sum_{j=1}^{|O(u_2)|} s(O_i(u_1), O_j(u_2)) \quad (4.16)$$

Similarly, for  $p_1 \neq p_2$ , we can rewrite the similarity between products as:

$$s(p_1, p_2) = \frac{C_2}{|I(p_1) \cup I(p_2)|} \sum_{i=1}^{|I(p_1)|} \sum_{j=1}^{|I(p_2)|} s(I_i(p_1), I_j(p_2))$$

From Equation 4.16, one can say that the similarity between two users in the bipartite heterogeneous network is the average similarity between the products they purchased. Consequently, one can say that the similarity between two products is the average similarity between the users who bought those products.

### Example 4.8

Let us calculate the similarity between users **A** and **B** in Figure 4.12.

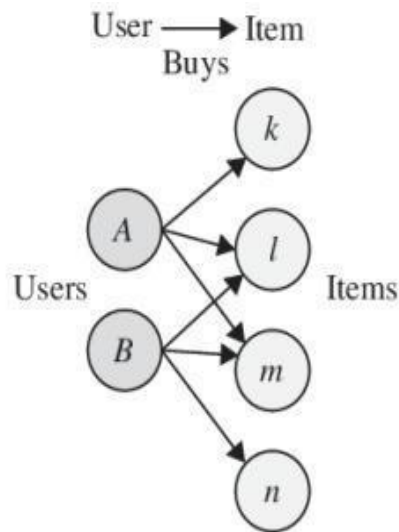


FIGURE 4.12. A toy example of a bipartite heterogeneous network of users (left-side nodes) and items (right-side nodes).

$$O(A) = \{k, l, m\} \text{ and } O(B) = \{l, m, n\}$$

$$I(k) = \{A\}, I(l) = \{A, B\}, I(m) = \{A, B\}, I(n) = \{B\}$$

Based on Equation 4.16, there are nine combinations to consider from the set  $\{O(A) \times O(B)\}$ :

$$s(A, B) = \frac{C_1}{3 \times 3} (s(k, l) + s(k, m) + s(k, n) + s(l, l) + s(l, m) + s(l, n) + s(m, l) + s(m, m) + s(m, n))$$

Using the base conditions  $s(X, X) = 1$  and  $s(X, Y) = s(Y, X)$ :

$$s(k, l) = \frac{C_2}{1 \times 2} [s(A, A) + s(A, B)] = \frac{C_2}{2} [1 + s(A, B)] = \frac{C_2}{2} + \frac{C_2 s(A, B)}{2}$$

$$s(k, m) = \frac{C_2}{1 \times 2} [s(A, A) + s(A, B)] = \frac{C_2}{2} [1 + s(A, B)] = \frac{C_2}{2} + \frac{C_2 s(A, B)}{2}$$

$$s(k, n) = \frac{C_2}{1 \times 1} [s(A, B)] = C_2 s(A, B)$$

$$s(l, l) = 1$$

$$\begin{aligned}
s(l, m) &= \frac{C_2}{2 \times 2} [s(A, A) + s(A, B) + s(B, A) + s(B, B)] = \frac{C_2}{4} [1 + 2s(A, B) + 1] \\
&= \frac{C_2}{2} + \frac{C_2 s(A, B)}{2} \\
s(l', n) &= \frac{C_2}{2 \times 1} [s(A', B) + s(B', B)] = \frac{C_2}{2} [1 + s(A, B)] = \frac{C_2}{2} + \frac{C_2 s(A, B)}{2} \\
s(m, l) &= s(l, m) = \frac{C_2}{2} + \frac{C_2 s(A, B)}{2} \\
s(m, m) &= 1 \\
s(m, n) &= \frac{C_2}{2} [s(A, B) + s(B, B)] = \frac{C_2}{2} [1 + s(A, B)] = \frac{C_2}{2} + \frac{C_2 s(A, B)}{2}
\end{aligned}$$

Placing the above values into the main equation, we get:

$$\begin{aligned}
s(A, B) &= \frac{C_1}{9} \left( 6 \cdot \frac{C_2}{2} + 6 \cdot \frac{C_2 s(A, B)}{2} + C_2 s(A, B) + 2 \right) \\
s(A, B) &= \frac{C_1}{9} (3C_2 + 4C_2 s(A, B) + 2) \\
9s(A, B) &= 3C_1 C_2 + 4C_1 C_2 s(A, B) + 2C_1 \\
s(A, B) &= \frac{3C_1 C_2 + 2C_1}{9 - 4C_1 C_2}
\end{aligned}$$

Solving the last equation and substituting  $C_1 = C_2 = 0.8$ , we get:

$$s(A, B) = 0.5465838509316772 \approx 0.547$$

### Notation key for this section:

$I(v), O(v)$  = set of in-degree and out-degree neighbours of node  $v$

$I_i(v), O_i(v)$  = the  $i^{th}$  individual in-/out-degree neighbour

$s(a, b)$  = SimRank similarity score between nodes  $a$  and  $b$ ,  $\in [0, 1]$

$C, C_1, C_2$  = decay factors ( $C_1$  for user-side,  $C_2$  for product-side in bipartite case)

$\phi$  = empty set

$\lim_{k \rightarrow \infty}$  = limit as iterations  $k$  go to infinity

Here's this section cleaned up, in order:

### 3.9 PATHSIM

#### 3.9.1 Introduction to Heterogeneous Network

$$\begin{aligned} s(l, m) &= \frac{C_2}{2 \times 2} [s(A, A) + s(A, B) + s(B, A) + s(B, B)] = \frac{C_2}{4} [1 + 2s(A, B) + 1] \\ &= \frac{C_2}{2} + \frac{C_2 s(A, B)}{2} \end{aligned}$$

In the last section, we discussed how a simple extension of SimRank is possible for a heterogeneous bipartite network. However, in general, a heterogeneous network can be made up of different types of entities, and even different types of links among the entities, denoting the variety of relations supported by the system. Most of the node ranking algorithms discussed so far do not leverage the type of relationship. Since entities and links of different types can carry different semantics and represent different latent similarities between entities of the network, it is important to discuss metrics that take semantic similarity into account.

In this section, we shall discuss a new metric for similarity score, called **PathSim**, proposed by Sun et al. (2011) as a measure of similarity search scoring and ranking in heterogeneous information networks.

Formally, a network  $G(V, E)$  consists of nodes and edges of different types. We can specify a set  $\mathcal{A}$  capturing different node types supported by the network, and a function  $\phi(v): V \rightarrow \mathcal{A}$ , mapping each node  $v \in V$  of the network to its object type. Similarly, we can specify a set  $\mathcal{R}$  capturing different types of links supported by the network, and a function  $\psi(e): E \rightarrow \mathcal{R}$ , mapping each edge  $e \in E$  of the network to its link type. Any information system can thus be represented as a tuple of the form  $(V, E, \mathcal{A}, \mathcal{R}, \phi, \psi)$ . Based on this tuple, we can differentiate between heterogeneous and homogeneous networks.

#### Heterogeneous Network

If  $|\mathcal{A}| > 1$  or  $|\mathcal{R}| > 1$  or both, then the network is considered a **heterogeneous network**. On the contrary, if  $|\mathcal{A}| = 1$  and  $|\mathcal{R}| = 1$ , then it is termed a **homogeneous network**.

**network.** Thus, a network consisting of either different types of nodes or different types of edges, or both, is a heterogeneous network. Some examples of heterogeneous networks are as follows:

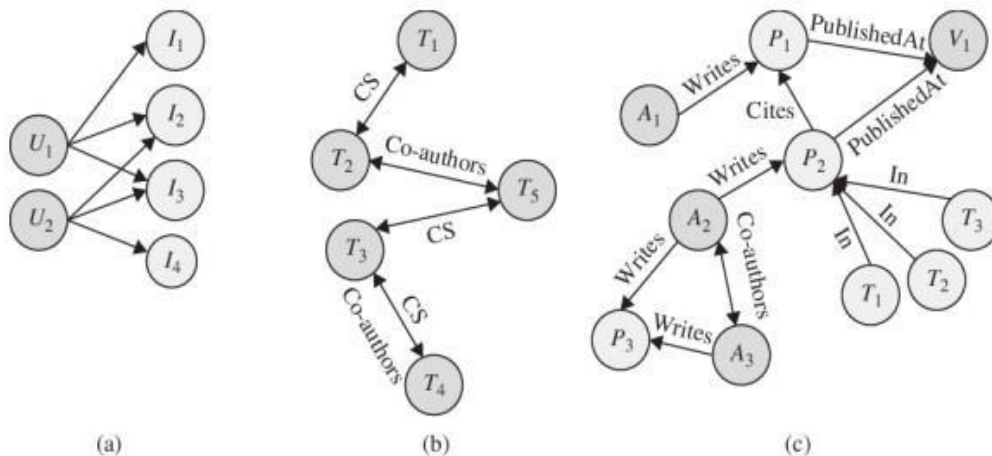
**Case 1:**  $|\mathcal{A}| > 1$  and  $|\mathcal{R}| = 1$ . Let us reconsider the bipartite network of users and products, with only a set of valid edges present between users and products. This is an example of a heterogeneous network with two types of objects  $\{\mathcal{A}:\text{users, products}\}$ , and only one type of link  $\{\mathcal{R}:\text{user} \xrightarrow{\text{buys}} \text{product}\}$ .

#### Example 4.9

As shown in Figure 4.13(a), nodes on the left are the users  $U_1, U_2, \dots$ , and nodes on the right are items  $I_1, I_2, \dots$ , bought by the users, like  $U_1 \xrightarrow{\text{buys}} \{I_1, I_2, I_3\}$ .

#### Example 4.9

As shown in Figure 4.13(a), nodes on the left are the users  $U_1, U_2, \dots$ , and nodes on the right are items  $I_1, I_2, \dots$ , bought by the users, like  $U_1 \xrightarrow{\text{buys}} \{I_1, I_2, I_3\}$ .



**FIGURE 4.13.** Some examples of heterogeneous networks, with different node types marked with different colours and different nodes of the same type annotated with subscripts. Double-sided arrows indicate undirected relations. (a) Case 1:  $|\mathcal{A}| > 1$  and  $|\mathcal{R}| = 1$ . (b) Case 2:  $|\mathcal{A}| = 1$  and  $|\mathcal{R}| > 1$ . (c) Case 3:  $|\mathcal{A}| > 1$  and  $|\mathcal{R}| > 1$ .

**Case 2:**  $|\mathcal{A}| = 1$  and  $|\mathcal{R}| > 1$ . Let us consider a network consisting of professors at a university, with different links capturing whether they work in the same department, or

have co-authored a paper together (even if from different departments). Here, there is only one type of node  $\{\mathcal{A}:\text{professors}\}$ , whereas different types of edges can exist between them, specifically  $\{\mathcal{R}:\text{professor} \xrightarrow{\text{co-authors}} \text{professor} \text{professor} \xrightarrow{\text{department}} \text{professor}\}$ .

### Example 4.10

As shown in Figure 4.13(b), teachers ( $T_1, T_2, \dots$ ) are connected with each other bi-directionally, as the edge relations are symmetric. While  $T_2$  and  $T_5$  are co-authors, they are not from the same department. Meanwhile,  $T_2$  and  $T_1$  are from the same department, but have not collaborated together. Nodes  $T_3$  and  $T_4$  are both from the same department, and have been co-authors as well. Thus, nodes of the same type are known to connect with their peers based on different edge types.

**Case 3:  $|\mathcal{A}| > 1$  and  $|\mathcal{R}| > 1$ .** Let us consider the example of a bibliography network consisting of different types of objects such as  $\{\mathcal{A}:\text{papers, conference, keywords, authors}\}$ , and different links between them, such as  $\{\mathcal{R}:\text{'paper' } \xrightarrow{\text{cites}} \text{paper} \text{term} \xrightarrow{\text{in}} \text{paper} \text{author} \xrightarrow{\text{writes}} \text{paper} \text{author} \xrightarrow{\text{co-authors}} \text{author}\}$ . A paper can be written by multiple authors, and two papers, irrespective of the authors, can be published at the same venue. At the same time, different authors can co-author a paper together, and different keywords can be used to represent the paper.

### Example 4.11

As shown in Figure 4.13(c), different node types include papers ( $P_i$ ), authors ( $A_i$ ), venues ( $V_i$ ), and keywords ( $T_i$ ). By collaborating on the same paper  $P_3$ , nodes  $A_2$  and  $A_3$  are co-authors (as shown by the bi-directional edge between them). Authors  $A_1$  and  $A_2$  are not directly connected, but one can assume that they work in the same field, as both have their respective papers  $P_1$  and  $P_2$  published at the same venue  $V_1$ , as marked by the "PublishedAt" edge. In addition, we see various terms  $T_1, T_2, T_3$  associated with a paper; these terms can appear in other papers as well, and based on similarity of terms we can obtain similar papers.

One should note that for directed edges, the reverse relations are also true. For example, take  $\text{Author} \xrightarrow{\text{writes}} \text{Paper}$  — if this edge exists in the network, then its reverse  $\text{Paper} \xrightarrow{\text{writtenBy}} \text{Author}$  is automatically true, even if not explicitly shown.

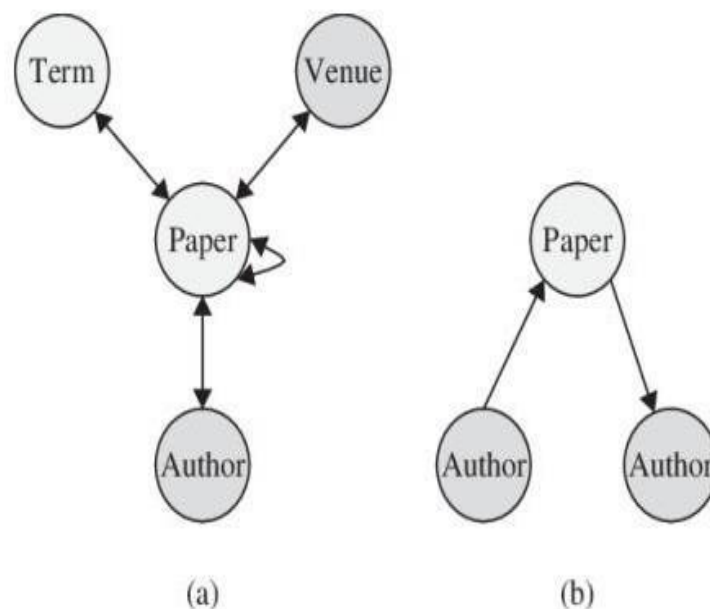
### 3.9.2 Network Schema and Meta-Paths

In our last example, we have seen with a bibliographic network, how complex the system can become with different types of interactions taking place between different entities. To model the system in an easier manner, one can provide a meta/schema level overview of the system. This is also known as a **network schema**.

#### Network Schema

The network schema  $T_G$  is a metadata-level outline for a heterogeneous directed network  $G(V, E)$  and the information tuple  $(V, E, \mathcal{A}, \mathcal{R}, \phi, \psi)$ , where  $\phi: V \rightarrow \mathcal{A}$  is the object-type mapping and  $\psi: E \rightarrow \mathcal{R}$  is the link-type mapping. With the directed graph being defined over the object types  $\mathcal{A}$  and relations  $\mathcal{R}$ , we represent the network schema as  $T_G = (\mathcal{A}, \mathcal{R})$ .

One can think of the network schema as an analogous form of the entity-relationship diagram of a relational database, with the schema telling us how many types of objects are present in the network, and their possible relations. An example of a network schema is shown in Figure 4.14(a).



**FIGURE 4.14.** Examples of (a) network schema and (b) meta-path for a toy bibliographic network. The network has bi-directional links between the entities — Terms ( $T$ ) and Papers ( $P$ ), Venue ( $C$ ) and Papers ( $P$ ), Authors ( $A$ ) and Papers ( $P$ ). A link from one paper to another (self-loop on  $P$ ) refers to the citation network among the papers. The meta-path shows the co-authorship in form of the composite relation — ‘author-paper-author’.

**FIGURE 4.14.** Examples of (a) network schema and (b) meta-path for a toy bibliographic network. The network has bi-directional links between the entities — Terms (T) and Papers (P), Venue (C) and Papers (P), Authors (A) and Papers (P). A link from one paper to another (self-loop on P) refers to the citation network among the papers. The meta-path shows the co-authorship in the form of the composite relation — 'author-paper-author'.

Based on the network schema, we can see that two entities can be connected via different paths. For example, two authors can be connected via an "author-paper-author" path of co-authoring, an "author-paper-paper-author" path of citation, etc. Different paths deliver varying semantic similarity/differences, or measure different topological connectivity. These paths, as discussed earlier, can be represented as **meta-paths**.

In general, a meta-path is a meta-level description of the structural connectivity between entities (which can be of the same type or different types). A meta-path  $\mathbf{P}$  over a graph  $G(V, E)$  and network schema  $T_G = (\mathcal{A}, \mathcal{R})$  can be defined as follows.

### Meta-Path

A meta-path is a path  $\mathbf{P}$  of length  $\mathbf{l}$  defining a composite relation over the  $\mathbf{l}$  links  $R = R_1 \circ R_2 \circ R_3 \cdots \circ R_l$  and  $l + 1$  objects  $A_1, A_2, \dots, A_{l+1}$ , denoted in the form of:

$$\overset{R_1}{A_1} \rightarrow \overset{R_2}{A_2} \rightarrow \cdots \overset{R_l}{A_l} \rightarrow A_{l+1}$$

The length  $\mathbf{l}$  of the path  $\mathbf{P}$  is the number of relations in  $\mathbf{P}$ . If there is no ambiguity of the relationship between entities  $A_i$  and  $A_{i+1}$ , then the meta-path can be represented as  $P = (A_1, A_2, \dots, A_{l+1})$ . For example, in our bibliography network, the co-author relationship of length  $l = 2$  can be described as Author  $\rightarrow$  Paper  $\rightarrow$  Author, or simply **APA**.

Two paths  $P_1 = (A_1, A_2, \dots, A_{l+1})$  and  $P_2 = (\overset{1}{A'_1}, \overset{2}{A'_2}, \dots, \overset{k+1}{A'_{k+1}})$  of lengths  $l$  and  $k$ , respectively, can be concatenated if and only if  $A_{l+1} = A'_1$ . The concatenated path  $\tilde{P} = (P_1, P_2)$  will be  $(A_1, A_2, \dots, A_{l+1}, \overset{1}{A'_1}, \overset{2}{A'_2}, \dots, \overset{k+1}{A'_{k+1}})$ . Again, the co-author relation is a meta-path that is a simple concatenation of paths AP and PA to form APA. An example of the APA meta-path is shown in Figure 4.14(b).

A **path instance**  $p$  of  $\mathbf{P}$  is defined for a set of entities  $(a_1, a_2, \dots, a_{l+1})$ , where  $\forall i, \phi(a_i) = A_i$ , and each subsequent link  $(a_i, a_{i+1})$  following the path occurs in the composite relation  $\mathbf{R}$  of  $\mathbf{P}$ .

In addition,  $P^-$  is the inverse/reverse of path  $\mathbf{P}$ , provided  $\mathbf{P}$  is symmetric, and the inverse path is valid in the network scheme. For example, saying that Author  $\xrightarrow{\text{writes}}$  Paper can be reverse-mapped as Paper  $\xrightarrow{\text{writtenBy}}$  Author.  $p^-$  then becomes the reverse path instance.

### Example 4.12

Given a bibliography network containing the following object types, what are some of the possible meta-paths to determine author similarity? Valid object types include {Author, (A), Paper, (P), Venue, (V), Terms, (T)}.

Some of the meta-paths could include:

1. **APA:** Author  $\xrightarrow{\text{Writes}}$  Paper  $\xrightarrow{\text{WrittenBy}}$  Author
2. **AVA:** Author  $\xrightarrow{\text{SpeaksAt}}$  Venue  $\xrightarrow{\text{Invites}}$  Author
3. **APPA:** Author  $\xrightarrow{\text{Writes}}$  Paper  $\xrightarrow{\text{Cites}}$  Paper  $\xrightarrow{\text{WrittenBy}}$  Author
4. **APTPA:** Author  $\xrightarrow{\text{Writes}}$  Paper  $\xrightarrow{\text{Contains}}$  Terms  $\xrightarrow{\text{In}}$  Paper  $\xrightarrow{\text{WrittenBy}}$  Author
5. **APVPA:** Author  $\xrightarrow{\text{Writes}}$  Paper  $\xrightarrow{\text{PublishedAt}}$  Venue  $\xrightarrow{\text{Published}}$  Paper  $\xrightarrow{\text{WrittenBy}}$  Author

### 3.9.3 Heuristic Measures of Object Similarity via Meta-Path

Given a meta-path  $P = (A_1, A_2, \dots, A_{l+1})$ , and the end-point object instances  $x \in A_1$  and  $y \in A_{l+1}$ , we can use the meta-path semantics to formulate simple heuristic measures of finding the similarity between objects  $\mathbf{x}$  and  $\mathbf{y}$ . While these heuristics can work for objects of different types on an individual-case basis, in order to perform ranking, homogeneous end points are needed for comparison. Some of these measures include:

1. **Path count:** It indicates the number of path instances  $p$  of  $P_l$ , which begin at  $\mathbf{x}$  and end at  $\mathbf{y}$ . The similarity score is:

$$s(x, y) = |\{p \in P_l \mid \forall (x \in A_1 \text{ and } y \in A_{l+1})\}|$$

2. **Random walk:** For a random surfer starting at  $\mathbf{x}$  and following the path  $P_l$ , what is the probability of it ending at  $\mathbf{y}$ ? This score is the sum of the probability of each

path instance  $p$  whose end points are  $\mathbf{x}$  and  $\mathbf{y}$ , where the path-wise probability is denoted by  $\text{Prob}(p)$ . The similarity score is:

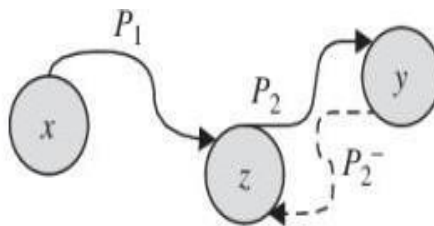
$$s(\mathbf{x}, \mathbf{y}) = \sum_{p \in P_l} \text{Prob}(p)$$

Here, we only consider those instances as valid if the condition  $(\mathbf{x} \in A_1 \text{ and } \mathbf{y} \in A_{l+1})$  is met; otherwise they are ignored for summation.

**3. Pairwise random walk:** For a concatenated meta-path  $P = (P_1, P_2)$  with instances starting at  $\mathbf{x}$  and  $\mathbf{y}$ , if we reverse the second sub-path to have two sets of random walkers starting at  $\mathbf{x}$  and  $\mathbf{y}$  and reaching a mid-point  $\mathbf{z}$ , it forms a valid instance as  $(\mathbf{x} \rightarrow \mathbf{z} \leftarrow \mathbf{y})$ . Here,  $\mathbf{z}$  is the instance of the object at the mid-point of both sub-paths, where the first sub-path ends and the second begins. The similarity score between  $\mathbf{x}$  and  $\mathbf{y}$ , given  $\mathbf{z}$ , can again be obtained as the sum of probabilities of starting at the end points and reaching a middle ground:

$$s(\mathbf{x}, \mathbf{y} \mid \mathbf{z}) = \sum_{(p_1, p_2^-) \in (P_1, P_2^-)} \text{Prob}(p_1) \cdot \text{Prob}(p_2^-)$$

Here, we only consider those instances as valid if the condition  $(\mathbf{x} \in A_1 \text{ and } \mathbf{y} \in A_{l+1})$  is met; otherwise they are ignored for summation. Also note that we make use of  $P_2^-$  for probability computation, as we want to modify the original path form of  $(\mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{y})$  to  $(\mathbf{x} \rightarrow \mathbf{z} \leftarrow \mathbf{y})$ . A sample meta-path is shown in Figure 4.15.



**FIGURE 4.15.** An example of pair-wise random walk heuristic with original meta-path as  $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2)$ . Here,  $x$  and  $y$  are the desired end points of the meta-path, and  $z$  is the concatenation mid-point. The dotted lines mark the inverse path.

**FIGURE 4.15.** An example of a pairwise random walk heuristic with original meta-path  $P = (P_1, P_2)$ . Here,  $x$  and  $y$  are the desired end points of the meta-path, and  $z$  is the concatenation mid-point. The dotted lines mark the inverse path.

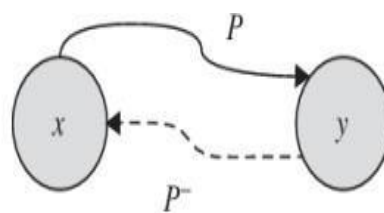
### 3.9.4 Formulation of PathSim

Let us start by defining one of the most basic symmetric meta-paths.

#### Round-trip Meta-Path

A meta-path of the form  $P = (P_l, P_l^-)$ , where the starting and ending objects are the same, i.e.,  $x \in A_1 \wedge x \in A_{l+1}$ , is termed a **round-trip meta-path**. By default, it is always symmetric. Figure 4.16 presents an example of a round-trip meta-path.

Simpler meta-path heuristics like path count or random walk are biased towards nodes with higher degree. Meanwhile, the pairwise random walk method works best if the overall path length is short, covering a small portion which increases the likelihood of meeting at a midpoint. To improve upon these shortcomings, a new metric for measuring similarity between two objects of the same type was introduced in the form of **PathSim** (Sun et al. 2011). It is a symmetric metric that captures both objects/peers with similar topology as well as visibility. One can define the visibility of a node in terms of how likely a random surfer ends up in a round-trip with respect to the given node and the meta-path under consideration. By making use of these two parameters, PathSim is able to capture the subtle similarity between strongly connected and visible peers.



**FIGURE 4.16.** An example of a round-trip meta-path with start and end object  $x$  and forward-path  $P$  visiting  $y$  intermediately.

**FIGURE 4.16.** An example of a round-trip meta-path with start and end object  $x$  and forward-path  $P$  visiting  $y$  intermediately.

## PathSim

Let us consider  $p_{x \rightsquigarrow y}$  as a path instance between the end points  $x$  and  $y$ , and  $p_{x \rightsquigarrow x}$ ,  $p_{y \rightsquigarrow y}$  as the round-trip paths for  $x$  and  $y$ , respectively. A meta-path-based symmetric similarity measure between two objects  $x$  and  $y$  of the same type can be given as:

$$s(x, y) = \frac{2 \times |\{p_{x \rightsquigarrow y}: p_{x \rightsquigarrow y} \in P\}|}{|\{p_{x \rightsquigarrow x}: p_{x \rightsquigarrow x} \in P\}| + |\{p_{y \rightsquigarrow y}: p_{y \rightsquigarrow y} \in P\}|} \quad (4.17)$$

The numerator of Equation 4.17 captures the instances of valid occurrence. Similar to the idea of path counts, it captures connectivity. However, this connectivity is balanced/normalised by a factor of **visibility**, captured by the denominator.

Based on the aforementioned definition, we can state the following properties of PathSim:

1. **Symmetric:**  $s(x, y) = s(y, x)$
2. **Normalised:**  $s(x, y) \in [0, 1]$
3. **Self-maximised:**  $s(x, x) = 1$

Let us consider an example (adapted from Lee et al. (2012)) to better understand PathSim.

### Example 4.13

Consider a meta-path AVA, i.e., Author  $\xrightarrow{\text{PublishesAt}}$  Venue  $\xrightarrow{\text{Publishes}}$  Author, and the venue-based publishing frequency captured in Table 4.2. The aim is to find the colleague most similar to Mike.

**Table 4.2. Table Capturing the Author–Venue Publishing Frequency**

| Author | MOD | VLDB | ICDE | KDD |
|--------|-----|------|------|-----|
| Mike   | 2   | 1    | 0    | 0   |
| Jim    | 50  | 20   | 0    | 0   |
| Mary   | 2   | 0    | 1    | 0   |
| Bob    | 2   | 1    | 0    | 0   |

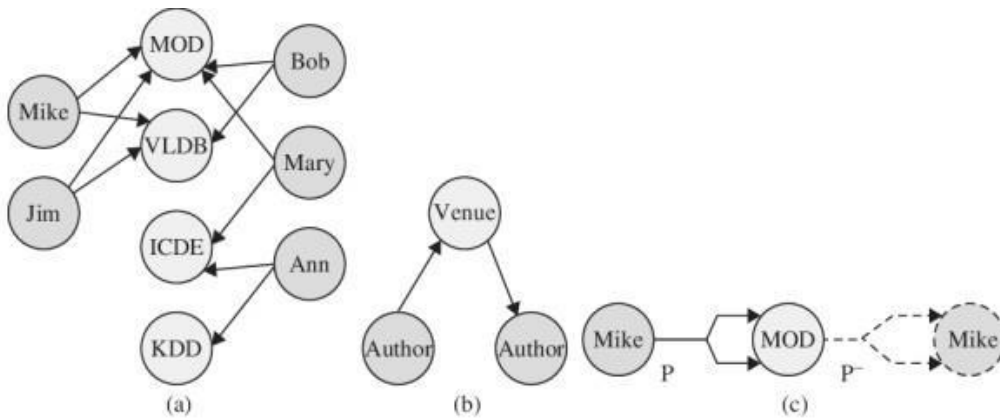
| Author | MOD | VLDB | ICDE | KDD |
|--------|-----|------|------|-----|
| Ann    | 0   | 0    | 1    | 1   |

Table 4.2 actually represents the adjacency matrix  $W_{AV}$  capturing the number of times an author has published at a venue, with an edge for each publication existing in the overall heterogeneous network. Let us capture the relation between authors and venues in the form of a bipartite subsystem, as shown in Figure 4.17(a).

Now, for the meta-path AVA as shown in Figure 4.17(b), let us consider the round-trip case, where both author nodes represent the same object. For the author Mike and the venue MOD, this round-trip looks like the path shown in Figure 4.17(c), where we have Mike as the starting and ending author. The information flows in the form of Mike *PublishesAt* MOD *Publishes* Mike.

From Table 4.2,  $W_{AV}$  for Mike–MOD is 2, i.e., starting at node Mike, we can reach the node MOD via either of the two published paths. Once at MOD, we can travel back to Mike using a reverse path of either of the two published paths. The path count can be represented as a product of the number of forward and backward paths. Owing to the symmetric relation of the round-trip paths, these two values will be the same and equal to the publishing frequency of the respective pair. Thus, for Mike–MOD, the round-trip path count score is  $2 \times 2 = 4$ .

The overall round-trip count score for Mike on the AVA meta-path is a summation over the round-trip paths with respect to all the venues. Similarly, we can compute the round-trip path count for each of the authors. This provides us with the **visibility**  $V_p$  of individual authors, capturing  $|\{p_{x \rightsquigarrow x}: p_{x \rightsquigarrow x} \in P\}|$ :



**FIGURE 4.17.** Graphical references for Example 4.12. (a) Publishing relationship between authors and venues based on Table 4.2. Note that the edges capture the presence/absence of a published relation and not the frequency of publication. (b) Meta-path under consideration,  $\mathcal{P} = \text{AVA}$ . (c) A path-instance example of round-trip path  $(\mathcal{P}, \mathcal{P}^-)$  between the author, Mike and the venue, MOD. The two forking edges represent the frequency of publication, where the dotted lines are the reverse path.

$$V_p(\text{Mike}) = 2 \times 2 + 1 \times 1 + 0 \times 0 + 0 \times 0 = 5$$

$$V_p(\text{Jim}) = 50 \times 50 + 20 \times 20 + 0 \times 0 + 0 \times 0 = 2900$$

$$V_p(\text{Mary}) = 2 \times 2 + 0 \times 0 + 1 \times 1 + 0 \times 0 = 5$$

$$V_p(\text{Bob}) = 2 \times 2 + 1 \times 1 + 0 \times 0 + 0 \times 0 = 5$$

$$V_p(\text{Ann}) = 0 \times 0 + 0 \times 0 + 1 \times 1 + 1 \times 1 = 2$$

Now, on similar lines, let us find the connectivity between Mike and Jim via MOD. Starting at Mike, we can reach MOD via any one of the two publishing edges. Once at MOD, we can move to Jim via one of the 50 publishing edges ( $W_{AV}$  for Jim–MOD = 50, based on Table 4.2). Here, the path count is again a product of the number of forward paths of Mike–MOD and the number of backward paths of MOD–Jim. In this case, it is  $2 \times 50$ . The overall **connectivity**  $C_p$  between Mike and Jim can be captured as a summation over the path counts obtained from individual venues, capturing  $\{p_{x \rightsquigarrow y}: p_{x \rightsquigarrow y} \in P\}$ :

$$C_p(\text{Mike}, \text{Jim}) = 2 \times 50 + 1 \times 20 + 0 \times 0 + 0 \times 0 = 120$$

$$C_p(\text{Mike}, \text{Mary}) = 2 \times 2 + 1 \times 0 + 0 \times 1 + 0 \times 0 = 4$$

$$C_p(\text{Mike}, \text{Bob}) = 2 \times 2 + 1 \times 1 + 0 \times 0 + 0 \times 0 = 5$$

$$C_p(\text{Mike}, \text{Ann}) = 2 \times 0 + 1 \times 0 + 0 \times 1 + 0 \times 1 = 0$$

Now, based on the connectivity ( $C_p$ ) and the visibility ( $V_p$ ), we can determine the PathSim-based similarity between the authors. We can see, based on Equation 4.17, that  $C_p$  and  $V_p$  capture the numerator and the denominator, respectively. Thus, the similarity scores in terms of  $C_p$  and  $V_p$  can be written as:

$$s(\text{Mike}, \text{Peer}) = \frac{2 \times C_p(\text{Mike}, \text{Peer})}{V_p(\text{Mike}) + V_p(\text{Peer})}$$

Based on this updated formula, we can determine the PathSim-based similarity scores as:

$$\begin{aligned} s(\text{Mike}, \text{Jim}) &= \frac{2 \times 120}{5 + \frac{2900}{2 \times 4}} = 0.0826 \\ s(\text{Mike}, \text{Mary}) &= \frac{2 \times 5}{5 + 5} = 0.8 \\ s(\text{Mike}, \text{Bob}) &= \frac{2 \times 5}{5 + 5} = 1.0 \\ s(\text{Mike}, \text{Ann}) &= \frac{2 \times 0}{5 + 2} = 0.0 \end{aligned}$$

From the PathSim calculations, we see that **Bob** is Mike's closest colleague — similar in terms of both preferred venue and frequency. While Jim appears to be more similar to Mike than Mary (owing to Jim's higher publishing frequency), he falls short of matching the publishing pattern of Mike. In addition, we see that while Ann is closer to Mike in terms of publishing frequency, there is no overlap in terms of venue, leading to a similarity score of zero. Thus, we say that if the preferred venue matches — i.e., valid paths in AVA exist for a given author pair — then those counts are normalised by the respective visibility of the authors. The closer the combined visibility matches the connectivity, the higher the similarity scores.

#### Notation key for this section:

- $\mathcal{A}$  = set of object/node types;  $\mathcal{R}$  = set of link/edge types
- $\phi(v)$  = object-type mapping function;  $\psi(e)$  = link-type mapping function
- $T_G = (\mathcal{A}, \mathcal{R})$  = network schema

- $A_1 \xrightarrow{R_1} A_2$  = meta-path notation (object type  $\rightarrow$  relation  $\rightarrow$  object type)
- $p_{x \rightsquigarrow y}$  = a path instance from x to y
- $P^-$  = reverse/inverse of path P
- $V_p$  = visibility (round-trip path count) of a node under meta-path P
- $C_p$  = connectivity (path count) between two nodes under meta-path P
- $W_{AV}$  = adjacency matrix between authors (A) and venues (V)

