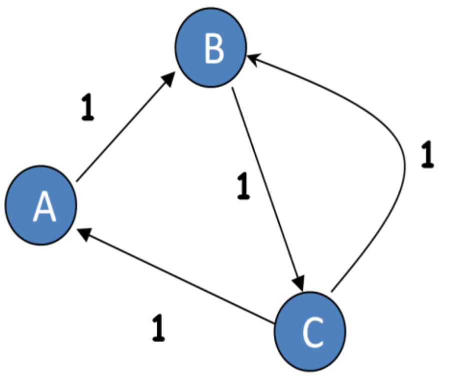
Random Walks on Graph

Piyush Grover(05CS1033)

A. Random Walks on Graph

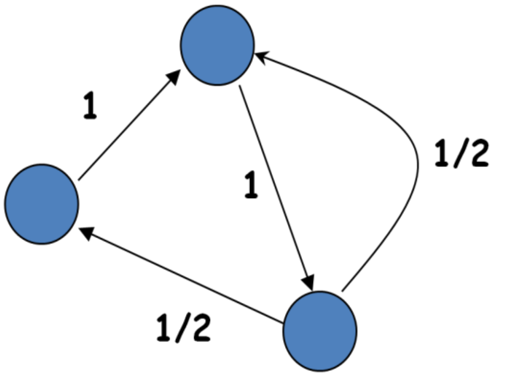
1. Definition and Examples

A random walk, sometimes denoted RW, is a mathematical formalization of a trajectory that consists of taking successive random steps.
Similarly, random walks on graph is a walk with randomly selected node at each step ie given a graph *G* and a starting point (node s), select a neighbor u of node s at random and move to this neighbor u. In the next step, select a neighbor of node u and move to it and so on. If the graph is undirected, u can be any neighbor of s but if the graph is directed then u can only be selected as a neighbor of s if there is an edge from s to u. The sequence of nodes selected in this way is a random walk on graph.



$$\begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{matrix}$$

Adjacency matrix A



$$\begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \end{matrix}$$

Transition matrix P

Given the adjacency matrix A of graph G, we can compute the transition matrix P. Since, for the given graph G, a(i, j) = 1 represents a forward edge from node i to node j whereas a(i, j) = 0 means there does not have any forward edge from i to j.

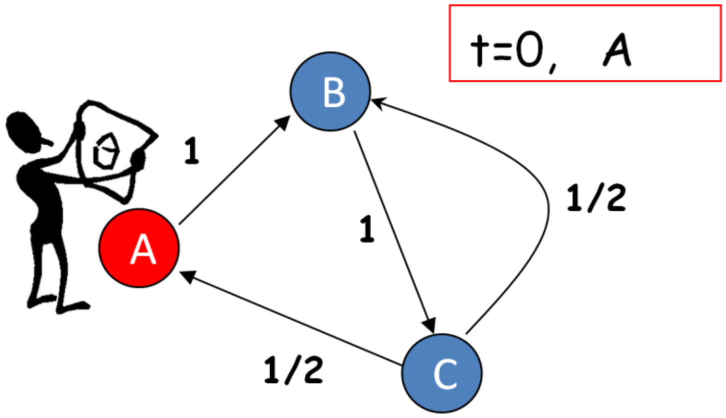
Now, transition matrix P represents the probability of reaching a node from a particular node ie p(i, j) is the probability of reaching node-j from node-i.

Therefore,

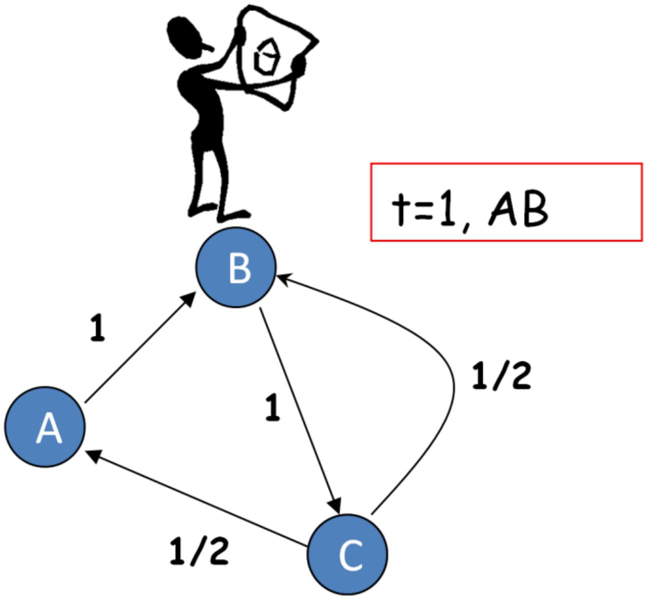
$$p(i,j) = \frac{a(i,j)}{\sum_k a(i,k)}$$

Following pictorial example clearly explains the random walks on graphs:

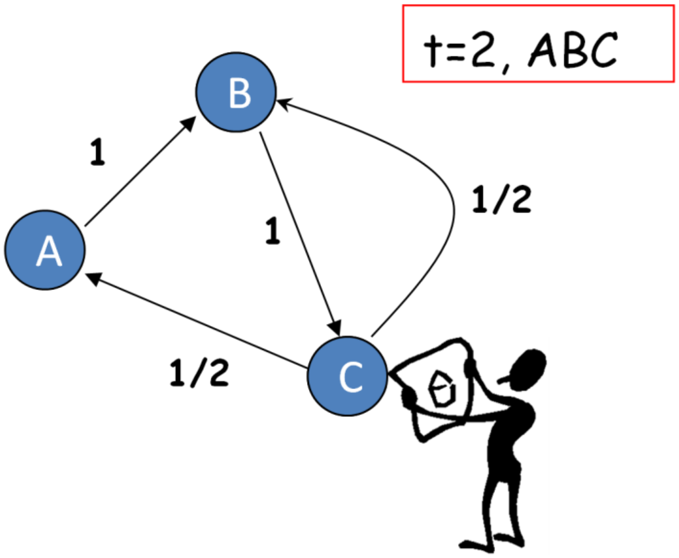
Given the above graph G with the starting node A, a random walker starts his walk from node-A. So, at t = 0 walker is at A.



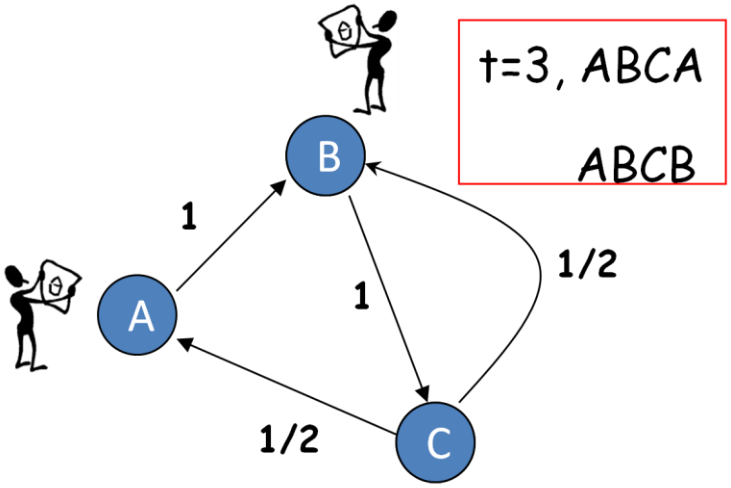
Now, in the next step at t = 1, walker randomly selects a neighbor of A. Since, there is only one edge going out of A (to B), random walker moves to node B with probability 1.



Similarly, in the next step random walker chooses node C as a next node to be visited because it has probability 1 of arriving from node B.



But now at node C random walker can jump to any of the two nodes because the probability of arriving at A and B is same so he randomly chooses one node and continue his walk. So, in this way random walks proceed.



Examples:

- The Page-Rank of a particular page is derived from the theoretical probability of visiting that page when clicking on links at random. This model (known as Random surfer model) is nothing but the random walks on the Web Graph.
- In social networks, flow of information is also a Random walk.
- The motion of dust particles in the atmosphere is also an example of Random walk.
- The shuffling of cards also a kind of random walk.

2. Relationship to Markov Chains

There are some probabilistic models which follow the Markov property are known as Markov chains or Markov process. Having the Markov property means that, given the present state, future states are independent of the past states. In other words, the description of the present state fully captures all the information that could influence the future evolution of the process. Future states will be reached through a probabilistic process instead of a deterministic one. These processes are also called Memory-less process.

A random walk on a directed graph is nothing but a Markov chain. Since, in a random walk, when a walker is at some particular node, the next node to be visited is independent of the previous nodes which have already been visited. It means the current node decides the next node on the basis of the transition matrix associated with the current node. The series of nodes in the random walk forms a Markov chain. Specifically, Markov chain is defined for directed graphs but we can generalize it for undirected graphs also, considering undirected graph as a directed graph with each edge in undirected graph as a bidirectional edge in the directed graph.

When we talk about graphs we deal with nodes and edges but when we use the term Markov process we are more concerned about the probability distributions. In the random graphs, we assume that initial node is given a priori but in the real situation the initial node is chosen on the basis of probabilities associated with each node. Here, we assume that the initial node is chosen from a probability distribution P_0.

Since, we have already stated what a transition matrix corresponds to now we will see some properties of transition matrix M.

The adjacency matrix A is given to us. So

$$M = D^{-1}A$$

where D^{-1} is derived from the diagonal matrix D.

$$d(i,j) = \begin{cases} \sum_k a(i,k) & if\ i = j \\ 0 & otherwise \end{cases}$$

$$d^{-1}(i,j) = \begin{cases} {}^{1}\!/_{d(i,j)} & if\ i = j \\ 0 & otherwise \end{cases}$$

Example:

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} \end{bmatrix}$$

$$\Rightarrow \quad \boldsymbol{D} = \begin{bmatrix} \mathbf{2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{2} \end{bmatrix}$$

$$\Rightarrow \quad \boldsymbol{D^{-1}} = \begin{bmatrix} {}^{\mathbf{1}}\!/_{\mathbf{2}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & {}^{\mathbf{1}}\!/_{\mathbf{2}} \end{bmatrix}$$

$$\Rightarrow \ \boldsymbol{M = D^{-1}A} = \begin{bmatrix} \mathbf{1/2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1/2} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} \end{bmatrix}$$

$$\Rightarrow \ \boldsymbol{M} = \begin{bmatrix} \mathbf{0} & \mathbf{1/2} & \mathbf{1/2} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{1/2} & \mathbf{1/2} & \mathbf{0} \end{bmatrix}$$

A random walk for a given graph can be represented in terms of the transition matrix *M* as follows.

$$P_{t+1} = M^T P_t$$

$$P_t = (M^T)^t P_0$$

$P_t(i)$ is the probability of reaching at node *i* after *t* steps.
$P_0(i)$ is the probability of node *i* to be selected as initial node.

Properties of transition matrix *M:*

- Transition matrix *M* exists if and only if all the diagonal elements of *D* are non zero which means there should be at least one edge going out from each node in the graph.
- If *A* is symmetric, it does not necessarily mean that *M* is also symmetric. *M* is symmetric when the graph is regular which means all the nodes in the graph have same out degree.
- *M* is also a right stochastic matrix which implies *M* is a matrix each of whose rows consists of nonnegative real numbers, with each row summing to 1. So the probability transition of going from any node to another node in *M* in *k* steps is given by M^k.

Properties of Random walks as Markov Chains:

a. ***Symmetric Random Walks:***

If a random walk (v_0,v_1,…,v_t) over a graph when reversed, has the same probability if v_0 = v_t then this kind of walk is called as Symmetric Random Walk. For an undirected graph, P(u →v) **=** P(v → u) holds.

b. Time Reversibility

A random walk is time reversible if the reversed walk is also a random walk with initial distribution P_t. Since, symmetric property of random walks holds for a restricted class of graphs, it's not very useful. All undirected graphs are time reversible. In addition to this, directed graphs can also be time reversible under certain conditions.

c. Stationary or Steady State distribution

*P** is stationary or steady state distribution if $P^* = M^T P^*$. Directed graphs could also be time reversible if the random walk follows a P* distribution which means probability of being at a particular node does not change with time, it remains forever. For every graph *G,* the stationary distribution *P** is:

$$P^*(v) = {}^{d(v)}/_{2m}$$

where *m* is the total number of edges in *G* and *d(v)* is the degree of node *v.*

Proof:

Instead of proving this we will verify the above statement.

Since, $\quad P^* = M^T P^*$

and $\quad M^T = (D^{-1}A)^T$

$$\Rightarrow \quad M^T = \left\{\begin{bmatrix} 1/d(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1/d(n) \end{bmatrix}\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}\right\}^T$$

$$\Rightarrow \quad M^T = \begin{bmatrix} a_{11}/d(1) & \cdots & a_{n1}/d(1) \\ \vdots & \ddots & \vdots \\ a_{1n}/d(n) & \cdots & a_{nn}/d(n) \end{bmatrix}^T$$

$$\Rightarrow \quad M^T = \begin{bmatrix} a_{11}/d(1) & \cdots & a_{1n}/d(n) \\ \vdots & \ddots & \vdots \\ a_{n1}/d(1) & \cdots & a_{nn}/d(n) \end{bmatrix}$$

$$\Rightarrow \quad P^* = \begin{bmatrix} a_{11}/d(1) & \cdots & a_{1n}/d(n) \\ \vdots & \ddots & \vdots \\ a_{n1}/d(1) & \cdots & a_{nn}/d(n) \end{bmatrix} P^*$$

$$\Rightarrow \quad P^* = \begin{bmatrix} a_{11}/d(1) & \cdots & a_{1n}/d(n) \\ \vdots & \ddots & \vdots \\ a_{n1}/d(1) & \cdots & a_{nn}/d(n) \end{bmatrix} \begin{bmatrix} {}^{d(1)}/_{2m} \\ \vdots \\ {}^{d(n)}/_{2m} \end{bmatrix}$$

$$\Rightarrow \quad P^* = \begin{bmatrix} {}^{a_{11}}/_{2m} + {}^{a_{12}}/_{2m} + \cdots + {}^{a_{1n}}/_{2m} \\ {}^{a_{21}}/_{2m} + {}^{a_{22}}/_{2m} + \cdots + {}^{a_{2n}}/_{2m} \\ \vdots \\ {}^{a_{n1}}/_{2m} + {}^{a_{n2}}/_{2m} + \cdots + {}^{a_{nn}}/_{2m} \end{bmatrix}$$

$$\Rightarrow \quad P^* = \begin{bmatrix} \sum_{k=1}^{n} {}^{a_{1k}}/_{2m} \\ \sum_{k=1}^{n} {}^{a_{2k}}/_{2m} \\ \vdots \\ \sum_{k=1}^{n} {}^{a_{nk}}/_{2m} \end{bmatrix}$$

$$\Rightarrow \quad P^* = \frac{1}{2m} \begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(n) \end{bmatrix}$$

$$\Rightarrow \quad P^*(v) = {}^{d(v)}/_{2m}$$

Since, here ${}^{1}/_{2m}$ is the normalization factor such that $\sum_k P^*(k) = 1$.

Properties:

- For a regular graph uniform distribution is a stationary distribution.
- Stationary distribution is unique for every connected graph but for a disconnected graph we will have more than one stationary distribution. This is so because, the initial distribution, for each of the disconnected components, will not be unique.

3. Parameters related to random walk

a. Access or hitting time

Access time or hitting time, denoted as H_{ij}, is the expected number of steps before the node *j* is visited, starting from node *i.*

b. Commute time

Commute time is the expected number of steps to go to node *j* starting from a node *i* and coming back to *i* from *j.*

Commute time $i \rightarrow j \rightarrow i := H_{ij} + H_{ji}$.

c. Cover time

Staring from a node/distribution, the expected number of steps to reach every node is the cover time.

. B. Applications of Random Walk

1. Ranking WebPages

Ranking WebPages is a classical problem in the information retrieval. The problem is, given a query word and a large number of WebPages consisting of the query word, to find out which of them are most relevant to the query. So, on the basis of the relevancy, rank the WebPages. But the basis of the relevancy can be hyperlink structure or can be the frequency of the query word occurring in the WebPages or can be the time when the page is last updated or it can be some other criteria.

a. Naïve and modified Page Ranks

Ranking WebPages is nothing but the random walk on the web graph, an assignment of Page Rank while walking around the WebPages. So, this can be done by stimulating the random surfer using the power iteration method which means, given the Web Link matrix *A* of the web graph, first compute the transition matrix *M* and compute power of M^T until it converges. The value of M^T at which it converges will help in deciding Eigen vector centrality to rank the web pages. So, the final result will consist of WebPages ordered by their rank.

$$P_t = (M^T)^t \, P_0$$

b. *Problems with Pagerank*

- There would be very less pages that link to a page which is new in the scenario hence this method is bias to the pages which came earlier.
- This method does not consider the content of the page ie it ignores the frequency of words occurring in the page which are relevant to the query.
- Since, we have discussed earlier that if the graph is disconnected then we will not have unique stochastic matrix *M*, therefore, the ranking of pages will not be unique.
- Consider the WebLink graph with webpages as the nodes of the graph, the node with no incoming links will get rank 0 or in the connected graph a node with no outgoing links will get nonzero rank and other nodes will get 0 ranks.
- Sinking of nodes is also the one of the major problem. Say if a node doesn't have any outgoing link and at some point of time random surfer arrives at this node then since due to absence of outgoing links, the random surfer can not jump to any other node, the random walk terminates and the nodes which may be of high importance can get 0 ranks. So, in the WebGraph, nodes with no outgoing links are called Sink nodes.
- PageRank is easy to be fooled in the sense that several nodes can be created that can be pointed to a particular page which in return will enhance the PageRank of that page.

2. Mixing rate

a. *Definition*

Since, WebLink graph consists of billion of nodes and power iteration method of computing webpages require multiplication of M^T various times until it converges, which is computationally intensive.

The mixing rate is how fast the random walk stationary distribution converges to its limiting value. It is of quite importance because power iteration method works if the Mixing rate is high. Mixing rates for some graphs can be very large such that the time complexity is so less even up to the O(*log n*).

b. *Relationship between Spectral gap and mixing rate*

Given a graph *G* and corresponding transition matrix *M* of the random walk. Compute the eigen values of M^T.

Suppose the eigen values are $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \lambda_n$.

Since *M* is a stochastic matrix, its highest eigen value is 1 and the lowest eigen value can not be less than -1. So the Spectral Gap for any graph is defined as

$$\begin{aligned}\text{Spectral Gap } &= \lambda_1 - \lambda_2 \\ &= 1 - \lambda_2.\end{aligned}$$

Mixing rate is governed by Spectral Gap such that higher the Spectral Gap faster the convergence that is higher the Mixing rate.

For a random walk starting at node *i*:

$$|P_t(j) - \pi(j)| \leq \sqrt{\frac{d(j)}{d(i)}}\lambda^t$$

where, $\pi(j)$ is the steady state distribution value of node *j* after t^th^ iteration.

λ is the second highest eigen value.

d(j) and *d(i)* are the degrees of node *j* and node *i* respectively.

The equation states that how far is the $P_t(j)$ from its steady state distribution value for the node *j*. If λ is very small then it converges very fast alternatively 1 – λ the Spectral Gap is high then it converges very fast. Since, for the web graph λ is very small which means for web it is not computationally intensive.

3. HITS: Hypertext Induced Text Search

Since, the problem with the PageRank is that the rank of a page is given in the order, in which the nodes appear in the random walk but this is not always so useful. Suppose, we search for a query say *Complex networks*, and since **amazon.com** has lots of books on complex networks and it is linked by many other links because it contains books on complex networks so it has a very high PageRank but it might not be informative in the sense that it doesn't contain any information about the *complex networks* but it links to many pages (books) which contain this information. So, there are two types of nodes in the Webgraph one is, which contains the information and another which points to these informative links. For this, Jon Kleinberg introduced two new terms for these types of nodes.

a. *Definition of hubness and authority*

As it has already mentioned that there are two types of nodes, one which contains the authoritative information which are called *authorities* and second, which points to these authoritative nodes, are called as *hubs*. Since, good hubs are those which link to good authorities and good authorities are those which are linked by the good hubs. Any node either can be hub or authority; it is decided by their hubness and authoritativeness score respectively.

For each *v ϵ* V in a subgraph of interest:

$a(v)$:- The authority score of node *v*.

$h(v)$:- The hubness of node *v*.

A site is very authoritative if it receives many citations. Citation from important sites weight more than citations from less-important sites.

Hubness shows the importance of a site. A good hub is a site that links to many authoritative sites.

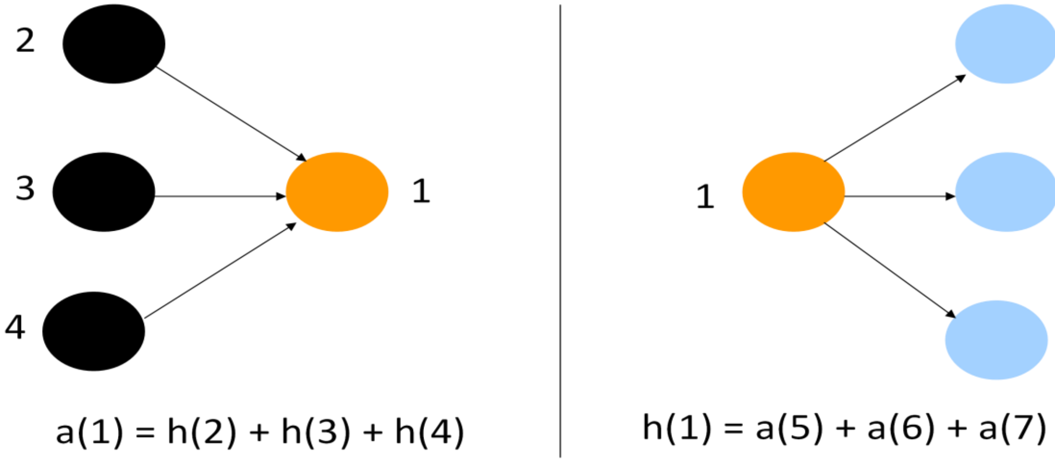
b. *Corresponding Markov Chain and the Convergence proof*

The following algorithm is used to create the query subgraph:

```
$Subgraph(\sigma, \varepsilon, t, d)$
    σ: a query string
    ε:a text-based search engine
    t, d: natural numbers
    Let $R_\sigma$ denote the top t results of ε on σ.
    Set $S_\sigma := R_\sigma$
    For each page $p \epsilon R_\sigma$
        Let $\Gamma^+(p)$ denotes the set of all pages p points to.
        Let $\Gamma$-(p) denotes the set of all pages pointing to p.
        Add all pages in $\Gamma^+(p)$ to $S_\sigma$..
        If | $\Gamma$-(p)| ≤ d then
            Add all pages in $\Gamma$-(p) to $S_\sigma$ .
        Else
            Add an arbitrary set of d pages in $\Gamma$-(p) to $S_\sigma$ .
        End
```

```
End
Return $S_\sigma$.
```

The hub score and authority score is computed after creating the subgraph. Hub score of a node is nothing but the sum of the authority scores of its children and authority score is the sum of the hub scores of its parents.



So, it has recursive dependency.

$$a(v) = \sum_{w \,\epsilon\, \mathrm{Pr}[v]} h(w)$$
$$h(v) = \sum_{w \,\epsilon\, ch[v]} a(w)$$

It's quite similar to PageRank but in HITS hub and authority are separate. One more thing is, similar to PageRank it also converges. Since given the adjacency matrix A of the WebGraph we will compute hubness and authoritativeness.

$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \sum_{w \,\epsilon\, \mathrm{Pr}[v]} h(w) = A\vec{h}$$

$$\vec{h} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix} = \sum_{w \,\epsilon\, ch[v]} a(w) = A^T\vec{a}$$

$$\Rightarrow \quad \vec{a} = (AA^T)\vec{a}$$
$$\Rightarrow \quad \vec{h} = (A^TA)\vec{h}$$

Since, it is similar to the equation that we got in the case of PageRank. So by multiplying (AA^T) repeatedly and normalizing it such that the total number of hubs remains the same as the initial value, we will get it converged.

So, $\vec{a}$ is nothing but the principle eigen vector of matrix AA^T and $\vec{h}$ is the principle eigen vector of matrix A^TA.

c. *Limitations of HITS*

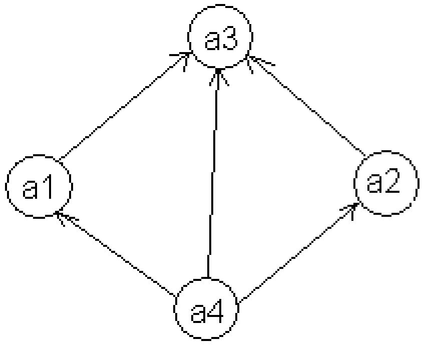
- The sinking problem of nodes is solved by introducing hubs and authorities separately. So, the sink nodes can have high authority scores but they definitely have zero hub score.
- Convergence is not an issue as we have seen above.
- HITS like PageRank can be fooled easily by using the concept of Tightly Knit Community (TKC) Effect.

d. *Tightly Knit Community Effect*

A tightly-knit community is a small but highly interconnected set of sites. Roughly speaking, the *TKC effect* occurs when such a community scores high in link-analyzing algorithms, even though the sites in the TKC are not authoritative on the topic, or pertain to just one aspect of the topic. As an example, consider a collection ***C*** which contains the following two communities: A community *y*, with a small number of hubs and authorities, in which every hub points to most of the authorities; and a much larger community *z*, in which each hub points to a smaller part of the authorities. The topic covered by *z* is the dominant topic of the collection, and is probably of wider interest on the WWW. Since there are many *z*-authoritative sites, the hubs do not link to all of them, whereas the smaller *y* community is densely interconnected. The TKC effect occurs when the sites of *y* are ranked higher than those of *z*.

4. Generalization of HITS: Co-citation networks and Bibliographic coupling

Since, WebGraph consists of WebPages as nodes in the similar way a Citation network consist of articles as nodes.



Suppose, a citation networks *G* is given with adjacency matrix *W*. for the given graph *G*, w(i, j) = 1 represents article a_i cites a_j.

$$a_{ij} = \sum_{k=1}^{n} w_{ik}^{T} w_{kj} = \sum_{k=1}^{n} w_{ki} w_{kj}$$

$$A = W^{T}W$$

Matrix *A* is called the *co-citation matrix*. a(i, j) = p means there are p nodes each for which w(i,k) = w(k, j) = 1 holds. It implies matrix *A* counts the number of articles in which both *i* and *j* are cited. *A* is a symmetric matrix which means it's an undirected network.

$$h_{ij} = \sum_{k=1}^{n} w_{ik} w_{kj}^{T} = \sum_{k=1}^{n} w_{ik} \, w_{jk}$$

$$H = WW^{T}$$

H is called the *bibliographic coupling matrix*. It gives the common number of citations, article *i* and article *j* has. It is also a symmetric matrix.

Since, *H* and *A* both are non stochastic matrices means they can not be viewed as random walk and also they are not the Markov chains.

5. SALSA: The Stochastic Approach to Link Structure Analysis

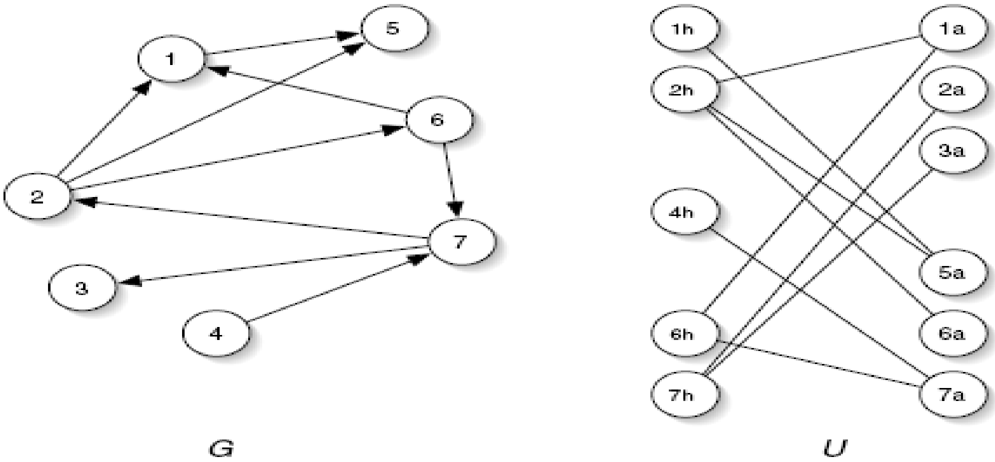
a. *Algorithm*

Since, HITS algorithm doesn't follow the random walk concept as hub and authority matrices are not stochastic in nature. Therefore, SALSA, which is the probabilistic extension of the HITS algorithm, is introduced. Random walk is carried out by following hyperlinks both in the forward and in the backward direction.

The basic idea of SALSA algorithm is, we have two types of walks, one is *Hub walk* and the second is *Authority walk*.

- *Hub walk:* Follow a Web link from a page u_h to a page w_a (a forward link) and then immediately traverse a backward link going from w_a to v_h where *(u,w)* Є *E* and *(v,w)* Є *E.*
- *Authority walk:* Follow a Web link from a page w_a to a page u_h (a backward link) and then immediately traverse a forward link going back from v_h to w_a where *(u,w) Є E* and *(v,w) Є E.*

To visualize the above two walks in a better way, we form a Bi-partite graph from the given WebGraph. Every node in the WebGraph can be seen both as a potentially hub and as an authority. For this, each node in the WebGraph is divided into two nodes, a hub and an authority as in the following figure.



G

U

Figure 5.6 Forming a bipartite graph in SALSA.

Node *3h* is not there in the figure because node *3* has no outgoing links hence its hub score is zero. If there is an outgoing link in *G* from a node *i* to a node *j* then there will be an undirected edge from node i*h to* node j*a* in *U*. Similarly, if there is an incoming link to a node *i* from a node *j* then there will be an edge in *U* connecting i*a* and j*h*. Thus, *U* is formed.

Now, we define two matrices on which the random walk will be performed.

- The *hub matrix* $\widetilde{H}$ is defined as follows:

$$\tilde{h}ij = \sum_{k|(i_h,k_a),(j_h,k_a)\epsilon\tilde{G}} \frac{1}{\deg(i_h)} \cdot \frac{1}{\deg(k_a)}$$

$\tilde{h}ij$ is the probability of going from node i*h* to node j*h* which means probability of going to node *j* from node *i* using a forward link and then a backward link.

- The *authority matrix* $\tilde{A}$

$$\tilde{a}ij = \sum_{k|(k_h,i_a),(k_h,j_a)\epsilon\tilde{G}} \frac{1}{\deg(i_a)} \cdot \frac{1}{\deg(k_h)}$$

$\tilde{a}ij$ is the probability of going from node i*a* to node j*a* which means probability of going to node *j* from node *i* using a backward link and then a forward link.

b. *Markov chain formulation and convergence*

Hub and authority matrices can be defined in terms of matrix form as follows:

$$Hub\ Matrix:\ \widetilde{H} = W_r W_c^T$$

$$Authority\ Matrix:\ \tilde{A} = W_c^T W_r$$

where W_r is the row stochastic matrix which is normalized by dividing each element by the sum of the row in W. Similarly, W_c is the column stochastic matrix which is normalized by dividing each element with the sum of the column in W.

Example:

$$W = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \Rightarrow W_r = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \\ 1/3 & 1/3 & 1/3 & 0 \end{bmatrix}$$

Similarly, $W_c = \begin{bmatrix} 0 & 1/2 & 1/3 & 0 \\ 1/2 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1 \\ 1/2 & 1/2 & 1/3 & 0 \end{bmatrix}$

Now, we look the Marcov chain formulation of this but before that we define some terms.

$$d_{in}(i) \triangleq \sum_{k \epsilon H | k \to i} w(k \to i)$$

$$d_{out}(k) \triangleq \sum_{i \epsilon A | k \to i} w(k \to i)$$

$$\mathcal{W} = \sum_{i \epsilon A} d_{in}(i) = \sum_{k \epsilon H} d_{out}(k)$$

Preposition:

Whenever M_A is an irreducible chain (has a single irreducible component), it has a unique stationary distribution $\pi = (\pi_1, \ldots, \pi_{|H|})$ satisfies:

$$\pi_i = \frac{d_{in}(i)}{\mathcal{W}} \; for\ all\ i \in A.$$

Similarly, whenever M_H is an irreducible chain, its unique stationary distribution $\pi = (\pi_1, \ldots, \pi_{|H|})$ satisfies:

$$\pi_k = \frac{d_{out}(k)}{\mathcal{W}} \; for\ all\ k \in H.$$

6. Link Analysis

a. Definition

We have seen some of the algorithms of information retrieval from the web, which are basically related to the links associated with the nodes of the WebGraph. Since, all the outgoing and incoming links are not of equal importance so which of the nodes are of relatively higher importance than the others, on the basis of importance how the weights should be given to the links during random walk, this is what we deal in the link analysis.

b. Limitations of link analysis

- *Meta tags/invisible text:* Meta elements provide information about a given Web page, most often to help search engines categorize them correctly. These meta tags/elements are hidden from the user. Search engines relying on meta tags in documents are often intentionally misled by web developers to enhance the PageRank.

- *Pay-for-place:* Search engines are biased in the sense that they are paid by the organizations to enhance there PagRank so if the page we are getting for our query may not be informative.
- *Stability:* Since, the WebGraph is dynamic, means there are new nodes/edges are added up by the time, but adding even a small number of nodes/edges to the graph has a significant impact in these algorithms.
- *Topic drift:* A top authority may be a hub of pages on a different topic resulting in increased rank of the authority page.
- *Content evolution:* Adding/removing links/content can affect the intuitive authority rank of a page requiring recalculation of page ranks

c. *Advanced Techniques: Static Ranking, Netrank*

Static Ranking: We have seen link analysis algorithms which do not deal mainly with the query part of a page. So, static ranking is basically related to how we can enhance the query part in the search engine. In the static ranking, PageRank is taken as a feature. In addition to PageRank, hub score, authority score, the frequency of words occurring in the webpage related to the query etc. are also used as the features. So, there are thousands of features for a particular page and by using these features a net score is computed for that page and the rank is given to it.

7. Clustering Based on message passing

a. *Chinese Whispers Algorithm*

b. *Affinity Propagation*