# Link Analysis

Hung Le

University of Victoria

February 17, 2019

## So many types of data with link structures

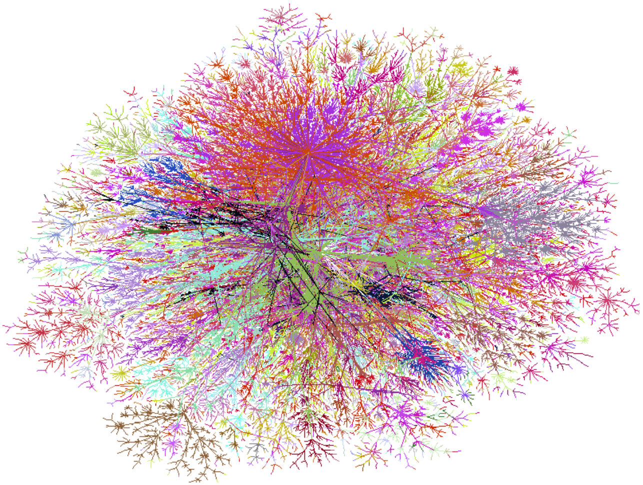


Figure: The Internet[1] network

[1] http://people.seas.harvard.edu/~babis/amazing.html

## So many types of data with link structures

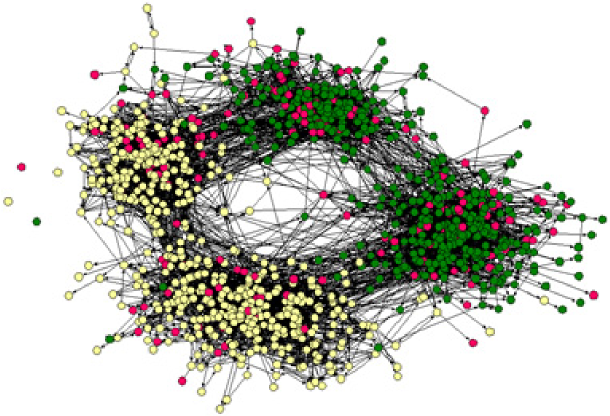


Figure: The Friendship[2] network

[2]http://people.seas.harvard.edu/~babis/amazing.html

## So many types of data with link structures

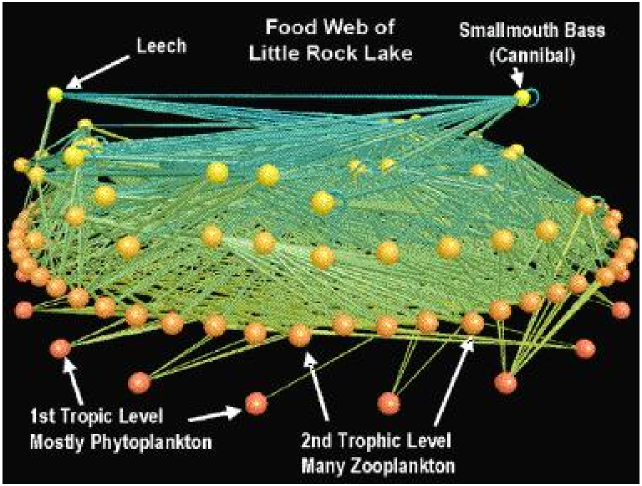

Leech
Food Web of Little Rock Lake
Smallmouth Bass (Cannibal)
1st Tropic Level Mostly Phytoplankton
2nd Trophic Level Many Zooplankton

Figure: The Food[3] network

[3] http://people.seas.harvard.edu/~babis/amazing.html

## PageRank[4]

**PageRank**

Given a network of nodes with (directional) links, find a robust way to assign importance scores to nodes based on the link structure.

[4] "The PageRank citation ranking: Bringing order to the web" by L. Page, S. Brin, R. Motwani, T. Winograd

## PageRank[4]

**PageRank**

Given a network of nodes with (directional) links, find a robust way to assign importance scores to nodes based on the link structure.

The scoring scheme should be robust w.r.t :

- Small changes.
- Local changes.

[4] "The PageRank citation ranking: Bringing order to the web" by L. Page, S. Brin, R. Motwani, T. Winograd

## PageRank[4]

**PageRank**

Given a network of nodes with (directional) links, find a robust way to assign importance scores to nodes based on the link structure.

The scoring scheme should be robust w.r.t :

- Small changes.
- Local changes.

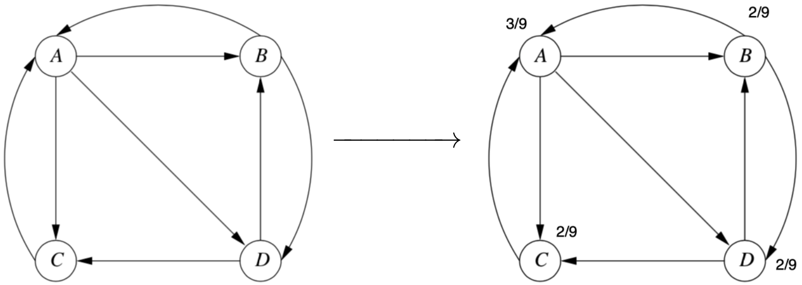
Pages with many (trustworthy) in-links should have higher scores.

[4] "The PageRank citation ranking: Bringing order to the web" by L. Page, S. Brin, R. Motwani, T. Winograd

# PageRank

## PageRank

Given a network of nodes with (directional) links, find a robust way to assign importance scores to nodes based on the link structure.



## Search Engine vs Spammers

Early search engines (Yahoo, AltaVista):

- Searches are mostly based on keyword matching: given a keyword, find web pages that contain the keyword.

## Search Engine vs Spammers

Early search engines (Yahoo, AltaVista):

- Searches are mostly based on keyword matching: given a keyword, find web pages that contain the keyword.

Search engine spammers are very *creative*:

- Add hot keywords to the page even if the page has nothing to do with the keywords.
- Change the keywords font to match with the background color, so that users do not realize it.

## Search Engine vs Spammers

The importance of links:

- Pages are linked to by many other pages should be more important $\Rightarrow$ define the number of in-links of a page to be its important scores.

## Search Engine vs Spammers

The importance of links:

- Pages are linked to by many other pages should be more important ⇒ define the number of in-links of a page to be its important scores.

Search engine spammers are very *creative*:

- Link farm: create many ghost web pages and put links to the target page to boost the target page's score.
- Make comments on many trustworthy pages and put links to the target page in their comments.

## PageRank

**PageRank**

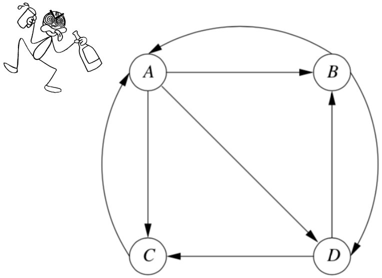
Given a network of nodes with (directional) links, find a robust way to assign importance scores to nodes based on the link structure.

## The Random Surfer Model

A random surfer, starting from a node of the graph, visit other nodes of the graph randomly by the following rule:

- From the standing node $A$, the surfer would visit a (uniformly) random node $B$ from the set all nodes that $A$ links to.

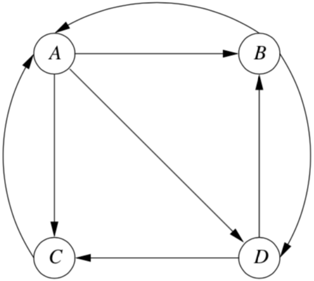
Frequency of a node visited by the random surfer after a sufficiently long walk is its importance score.



(Image from [5])

[5] http://clipart-library.com/clipart/LcdRjbnc4.htm

## The Random Surfer Model - An Example



A random walk of 81 steps by the random surfer:
ACADCADBACACACADCADCABDCADCADCACADBDBADBADBDCABABA-CACACADBABDBDCABDBACABABDCABABA
Frequency: $F[A] = 29, F[B] = 28, F[C] = 27, F[D] = 27$.

## The Random Surfer Model - Why

Why the random surfer model gives good important scores:

- Pages with more in-links are visited more often.
- Small changes or local changes of the network does not affect the behavior of the random surfer by much.

## The Random Surfer Model

Simulating a long random walk of the random surfer is very time consuming. Let's look at a slightly different view.

## The Random Surfer Model

Simulating a long random walk of the random surfer is very time consuming. Let's look at a slightly different view.

- Represent the distribution of location of the random surfer by a vector $\mathbf{v}$.
    - $\mathbf{v}[i]$ is the probability of the random surfer at $i$-th vertex.
    - $\sum_{i=1}^{n} \mathbf{v}[i] = 1$
- Suppose that the random surfer starts at any vertex of equal probability. That is, the initial vector $\mathbf{v}_0$ has $\mathbf{v}_0[i] = \frac{1}{n}$ for all $i$.

## The Random Surfer Model

Simulating a long random walk of the random surfer is very time consuming. Let's look at a slightly different view.

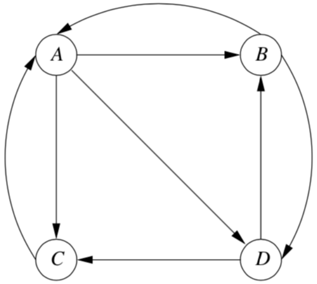
- Represent the distribution of location of the random surfer by a vector $\mathbf{v}$.
  - $\mathbf{v}[i]$ is the probability of the random surfer at $i$-th vertex.
  - $\sum_{i=1}^{n} \mathbf{v}[i] = 1$
- Suppose that the random surfer starts at any vertex of equal probability. That is, the initial vector $\mathbf{v}_0$ has $\mathbf{v}_0[i] = \frac{1}{n}$ for all $i$.

Represent the graph of a *transition matrix* $M$ where $M[i,j]$ is the probability that the surfer at vertex $j$ will move to vertex $i$ in the next step.

$$M[i,j] = \begin{cases} \frac{1}{\deg^-(j)} & \text{if there is an arc } j \to i \text{ in } G \\ 0 & \text{otherwise} \end{cases}$$

where $\deg^-(j)$ is the number of arcs going out from $j$ in the graph.

## An example of transition matrix



$$M = \begin{bmatrix} 0 & \frac{1}{2} & 1 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \qquad \mathbf{v}_0 = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix} \tag{1}$$

## Markov Process

The distribution of the random surfer in the next step, given the distribution at the current step $\mathbf{v}_{t-1}$ is:

$$\mathbf{v}_t = M\mathbf{v}_{t-1} \tag{2}$$

This is called a *Markov process*. The limiting distribution (if any) $\mathbf{v}$ would satisfied:

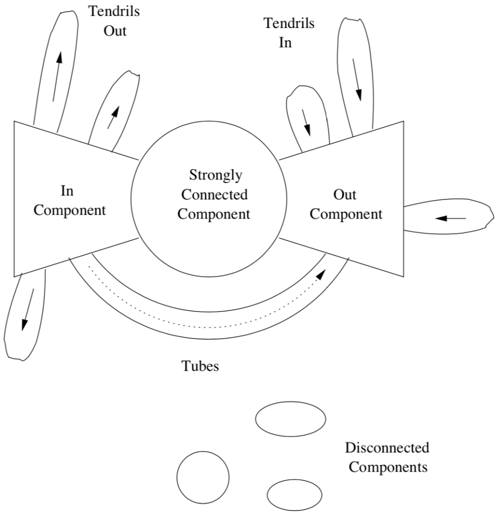
$$\mathbf{v} = M\mathbf{v} \tag{3}$$

In practice, we often obtain a limiting distribution (if any) after 50-75 steps.

## Existence of the limiting distribution

The limiting distribution exists if and only if the graph is strongly connected. That is, there is a directed path $s \rightsquigarrow t$ between any two nodes $s, t \in G$.

## Structure of the Web



## Dead ends and spider traps

Dead ends and spider traps are particularly problematic because once the random surfer gets into them, the surfer cannot get out. Thus, in this case, there is no limiting distribution of the process:

$$\mathbf{v}_t = M\mathbf{v}_{t-1} \tag{4}$$

## Avoiding dead ends and spider traps

We can avoid spider traps using *taxation*:

- Assume that the random surfer, from a node, can be *teleported* to a random node (which possibly is the node itself) with proability $\frac{1}{n}$.

## Avoiding dead ends and spider traps

We can avoid spider traps using *taxation*:

- Assume that the random surfer, from a node, can be *teleported* to a random node (which possibly is the node itself) with proability $\frac{1}{n}$.

The transition matrix becomes:

$$\beta M + (1-\beta)\frac{1}{n}\mathbf{1}_{n\times n} \tag{5}$$

where $\mathbf{1}_{n\times n}$ is the $n \times n$ matrix of all 1s.

## Avoiding dead ends and spider traps

We can avoid spider traps using *taxation*:

- Assume that the random surfer, from a node, can be *teleported* to a random node (which possibly is the node itself) with proability $\frac{1}{n}$.

The transition matrix becomes:

$$\beta M + (1-\beta)\frac{1}{n}\mathbf{1}_{n\times n} \tag{5}$$

where $\mathbf{1}_{n\times n}$ is the $n \times n$ matrix of all 1s.
The distribution of the random surfer after one step become:

$$\begin{aligned} \mathbf{v}_t &= (\beta M + (1-\beta)\frac{1}{n}\mathbf{1}_{n\times n})\mathbf{v}_{t-1} \\ &= \beta M\mathbf{v}_{t-1} + (1-\beta)\frac{1}{n}\mathbf{1} \end{aligned} \tag{6}$$

where $\mathbf{1}$ is the vector of all 1s. $\beta \in [0.8, 0.9]$ in practice.

## Avoiding dead ends and spider traps

Dead ends has the corresponding column in the transition matrix $M$ of all 0s. Thus, the corresponding column of the matrix:

$$\beta M + (1-\beta)\frac{1}{n}\mathbf{1}_{n\times n} \tag{7}$$

is sum to $(1-\beta)$, not 1.0. That implies $(\beta M + (1-\beta)\frac{1}{n}\mathbf{1}_{n\times n})\mathbf{v}$ is not a probability distribution. To resolve this situation:

- We remove all dead ends until the graph has no dead end left. Note that removing a dead ends can give rise to other dead ends.
- Update the PageRank of dead ends by the reverse removing orders following the equation:

$$\mathbf{v}[i] = \sum_{j\in N^{+}(i)} \frac{\mathbf{v}[j]}{\deg^{-}[j]} \tag{8}$$

## Computing PageRank - No dead end

```
PAGERANK($G(V, E)$)
    initialize $\mathbf{v}_0 \leftarrow \frac{1}{n}\mathbf{1}$.
    choose $\beta \in [0.8, 0.9]$
    **repeat**
        **for** $i \leftarrow 1$ to $n$
            $\mathbf{v}_t[i] = \beta \sum_{j \in N^+(i)} \frac{\mathbf{v}_{t-1}[j]}{\deg^-(j)} + (1-\beta)\frac{1}{n}$
    **until** converged
    return $\mathbf{v}_T$
```

## Computing PageRank - No dead end

```
PAGERANK($G(V, E)$)
    initialize $\mathbf{v}_0 \leftarrow \frac{1}{n}\mathbf{1}$.
    choose $\beta \in [0.8, 0.9]$
    **repeat**
        **for** $i \leftarrow 1$ to $n$
            $\mathbf{v}_t[i] = \beta \sum_{j \in N^+(i)} \frac{\mathbf{v}_{t-1}[j]}{\deg^-(j)} + (1-\beta)\frac{1}{n}$
    **until** converged
    return $\mathbf{v}_T$
```

Time complexity: $O(T(|V| + |E|))$ where $T$ is the number of iterations. In practice, $T \in [50, 75]$ suffices.

## Computing PageRank - with dead ends

```
$\textsc{PageRankWithDeadEnds}(G(V,E))$
    Remove dead ends from $G$.
    $\mathbf{v}_0 \leftarrow \textsc{PageRank}(G(V,E))$
    **for each** dead end $i$ in the reverse removing order
        $\mathbf{v}_t[i] \leftarrow \sum_{j \in N^+(i)} \frac{\mathbf{v}_{t-1}[j]}{\deg^-[j]}$
    return $\mathbf{v}_T$
```

Time complexity: $O(T(|V|+|E|))$ where $T$ is the number of iterations. In practice, $T \in [50,75]$ suffices.

## Using PageRank in Search

PageRank is still a very important signal for ranking today, but it is just one among many.

## Topic Sensitive PageRank - Motivation

Different people interested in different topics given the same query. For example, results for the query “Jaguar” could be:

Mac OS X
Version 10.2
Jaguar

## Topic Sensitive PageRank

Idea: We only teleport the random surfer to a page belonged to a particular topic.
Intuition: A page likely is linked to another page of the same topic.
The transition matrix becomes

$$\beta M + (1-\beta)\frac{1}{|S|}\mathbf{1}^S_{n\times n} \tag{9}$$

where $\mathbf{1}^S_{n\times n}$ is the $n \times n$ matrix such that rows belong to pages in the same topic set $S$ are all 1s. Hence,

$$\begin{aligned}\mathbf{v}_t &= (\beta M + (1-\beta)\frac{1}{|S|}\mathbf{1}^S_{n\times n})\mathbf{v}_{t-1} \\ &= \beta M \mathbf{v}_{t-1} + (1-\beta)\frac{1}{|S|}\mathbf{1}_S\end{aligned} \tag{10}$$

where $\mathbf{1}_S$ is the vector of all 1s for pages in $S$; pages not in $S$ have 0 in $\mathbf{1}_S$.

## Finding Topics of Web pages

This problem has been studied for decades. A possible simple way to find topics is:

- Determine a set of keywords $S_1, S_2, \ldots, S_k$ for each of $k$ topics.

## Finding Topics of Web pages

This problem has been studied for decades. A possible simple way to find topics is:

- Determine a set of keywords $S_1, S_2, \ldots, S_k$ for each of $k$ topics.
- Compute the Jaccard similarity between the page $P$ and each of the $S_i$, pick the topic with highest similarity as the topic for $P$.
  - Recall that both $P$ and $S_i$ are set of words, so calculating Jaccard similarity makes sense.

## Link Spam

Spammers are very creative. Here is one of the possible way they used to spam links against PageRank:

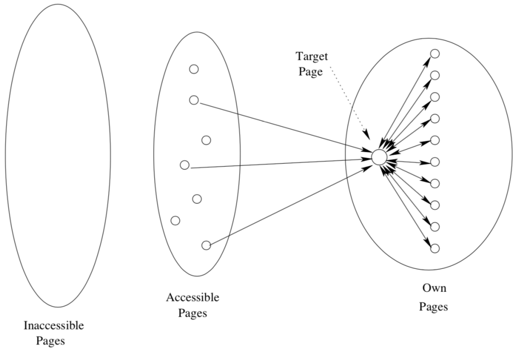


Figure: The Web from a link spammer's point of view.

## The Math of Spamming

Set up:

- Taxation parameter $\beta$, say $\beta = 0.85$.
- Target page $t$ and the number of supporting pages is $m$.
- Let $x$ be the amount of Page Rank score contributed by accessible pages.

Let $y$ be the final PageRank of $t$. Obviously $y \geq x$.

## The Math of Spamming

Set up:

- Taxation parameter $\beta$, say $\beta = 0.85$.
- Target page $t$ and the number of supporting pages is $m$.
- Let $x$ be the amount of Page Rank score contributed by accessible pages.

Let $y$ be the final PageRank of $t$. Obviously $y \geq x$.

**Observation**

*PageRank of each supporting page is:*

$$\beta \frac{y}{m} + \frac{1-\beta}{n} \tag{11}$$

## The Math of Spamming (Cont.)

- Contribution of PageRank of accessible pages is $x$.
- Contribution of PageRank of each supporting page back to $y$ is $\beta(\beta\frac{y}{m} + \frac{1-\beta}{n})$.
- Contribution of teleporting is $\frac{1-\beta}{n}$. Typically, $n$ is about billions, so this contribution is negligible. We will drop this in our calculation.

## The Math of Spamming (Cont.)

- Contribution of PageRank of accessible pages is $x$.
- Contribution of PageRank of each supporting page back to $y$ is $\beta(\beta\frac{y}{m} + \frac{1-\beta}{n})$.
- Contribution of teleporting is $\frac{1-\beta}{n}$. Typically, $n$ is about billions, so this contribution is negligible. We will drop this in our calculation.

Putting everything together, we have:

$$y = x + \beta m(\beta\frac{y}{m} + \frac{1-\beta}{n}). \tag{12}$$

## The Math of Spamming (Cont.)

- Contribution of PageRank of accessible pages is $x$.
- Contribution of PageRank of each supporting page back to $y$ is $\beta(\beta\frac{y}{m} + \frac{1-\beta}{n})$.
- Contribution of teleporting is $\frac{1-\beta}{n}$. Typically, $n$ is about billions, so this contribution is negligible. We will drop this in our calculation.

Putting everything together, we have:

$$y = x + \beta m(\beta\frac{y}{m} + \frac{1-\beta}{n}). \tag{12}$$

Solving the equation, we get:

$$y = \frac{x}{1-\beta^2} + \frac{\beta}{\beta+1}\frac{m}{n} \tag{13}$$

With $\beta = 0.85$, $y = 3.6x + 0.46\frac{m}{n}$. Note that, without supporting pages, $y \sim x$.

## Combating Link Spam - TrustRank

TrustRank is topic-sensitive PageRank, where the topic is *trustworthy*. A page $p$ is trustworthy if the spammer cannot easily add a link from $p$ to its target page.

## Combating Link Spam - TrustRank

TrustRank is topic-sensitive PageRank, where the topic is *trustworthy*. A page $p$ is trustworthy if the spammer cannot easily add a link from $p$ to its target page.

$$\mathbf{v}_t = \beta M \mathbf{v}_{t-1} + (1-\beta)\frac{1}{|S|}\mathbf{1}_S \tag{14}$$

where $S$ is the set of trustworthy pages, that is determined beforehand.

## Combating Link Spam - Spam Mass

The spam mass of a page $p$ is:

$$\frac{r(p) - t(p)}{r(p)} \tag{15}$$

where $r(p)$ is the (normal) PageRank of $p$ and $t(p)$ is the TrustRank of $p$.

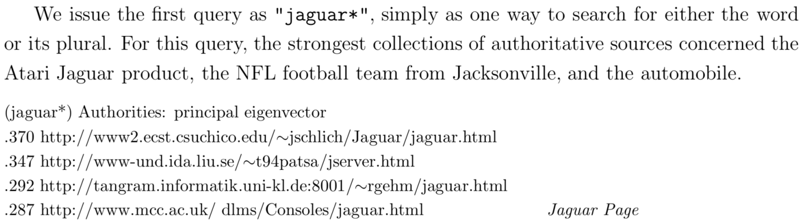
A page $p$ that has big (close to 1) spam mass is likely a spam page. We can eliminate $p$ out of the index of the search engine.

## Hubs and Authorities[6]

- Pages have out-links to many other pages called *hubs*. They are portals to access other pages.
- Pages have in-links from many other pages called *authorities*. They provide valuable information about a topic.

[6] "Authoritative Sources in a Hyperlinked Environment" by Jon Kleinberg

## Hubs and Authorities - An example



We issue the first query as `"jaguar*"`, simply as one way to search for either the word or its plural. For this query, the strongest collections of authoritative sources concerned the Atari Jaguar product, the NFL football team from Jacksonville, and the automobile.

(jaguar*) Authorities: principal eigenvector

.370 http://www2.ecst.csuchico.edu/∼jschlich/Jaguar/jaguar.html

.347 http://www-und.ida.liu.se/∼t94patsa/jserver.html

.292 http://tangram.informatik.uni-kl.de:8001/∼rgehm/jaguar.html

.287 http://www.mcc.ac.uk/ dlms/Consoles/jaguar.html *Jaguar Page*

Figure: Authorities example by Kleinberg.

## HITS - Hyperlink- Induced Topic Search

Each page $p_i$ has two features: hubiness $\mathbf{h}[i]$ and authority $\mathbf{a}[i]$, representing the degrees to which page $p_i$ is a hub an/or an authority.

- The vectors **h** and **a** do not need to be probability distributions, i.e, the sum of elements is not 1.

[7] This is the original formulation of HITS by Jon Kleinberg and is slightly different from the MMDS book.

## HITS - Hyperlink- Induced Topic Search

Each page $p_i$ has two features: hubiness $\mathbf{h}[i]$ and authority $\mathbf{a}[i]$, representing the degrees to which page $p_i$ is a hub an/or an authority.

- The vectors $\mathbf{h}$ and $\mathbf{a}$ do not need to be probability distributions, i.e, the sum of elements is not 1.

Hubiness and authority are closely related to each other by following recursive relation:

$$\begin{aligned} \mathbf{h}[i] &= \lambda \sum_{j \in N^-(i)} \mathbf{a}[j] \\ \mathbf{a}[i] &= \mu \sum_{j \in N^+(i)} \mathbf{h}[j] \end{aligned} \tag{16}$$

where $\lambda$ and $\mu$ are chosen so that[7] $||\mathbf{h}||_2 = ||\mathbf{a}||_2 = 1$. Here, $N^-(i)$ and $N^+(i)$ are the set of nodes that have links from and to $i$, respectively.

[7] This is the original formulation of HITS by Jon Kleinberg and is slightly different from the MMDS book.

## HITS- Hyperlink- Induced Topic Search (Cont)

In matrix form:

$$\begin{aligned}\mathbf{h} &= \lambda A\mathbf{a}\\ \mathbf{a} &= \mu A^T\mathbf{h}\end{aligned} \tag{17}$$

where $A$ is the adjacency matrix of $G$, i.e, $A[i,j] = 1$ if there is a link from $i$ to $j$, and $A[i,j] = 0$ otherwise.

$$\begin{aligned}\mathbf{h} &= \lambda\mu AA^T\mathbf{h}\\ \mathbf{a} &= \mu\lambda A^TA\mathbf{a}\end{aligned} \tag{18}$$

A conclusion from linear algebra: there always exists **h** and **a** satisfying Equation 18, regardless of the structure of the web. Specifically, **h** and **a** are eigenvectors of $AA^T$ and $A^TA$, respectively.

## Computing Hubbiness and Authority

```
$\mathrm{HITS}(G(V,E))$
    initialize unit vectors $\mathbf{a} = \mathbf{h} \leftarrow \frac{1}{\sqrt{n}}\mathbf{1}$.
    **repeat**
        **for** $i \leftarrow 1$ to $n$
            $\mathbf{h}[i] = \sum_{j \in N^{-}(i)} \mathbf{a}[j]$
            $\mathbf{a}[i] = \sum_{j \in N^{+}} \mathbf{h}[j]$
        $\mathbf{h} \leftarrow \frac{\mathbf{h}}{||\mathbf{h}||_2}$
        $\mathbf{a} \leftarrow \frac{\mathbf{a}}{||\mathbf{a}||_2}$
    **until** converged
    return $\mathbf{a}, \mathbf{h}$
```

Time complexity: $O(T(|V| + |E|))$ where $T$ is the number of iterations. In practice $T \in [20, 100]$ suffices.