Link Analysis Ranking
Random walks
GRAPHS AND LINK ANALYSIS RANKING
Link Analysis Ranking

• Use the **graph structure** in order to determine the **relative importance** of the nodes
  • Applications: Ranking on graphs (Web, Twitter, FB, etc)

• **Intuition**: An edge from node \( p \) to node \( q \) denotes endorsement
  • Node \( p \) **endorses/recommends/confirms** the **authority/centrality/importance** of node \( q \)
  • Use the graph of recommendations to assign an **authority value** to every node
Rank by Popularity

• Rank pages according to the number of incoming edges (in-degree, degree centrality)

1. Red Page
2. Yellow Page
3. Blue Page
4. Purple Page
5. Green Page
Popularity

• It is not important only how many link to you, but how important are the people that link to you.

• Good authorities are pointed by good authorities
  • Recursive definition of importance
PageRank

• Good authorities should be pointed by good authorities
  • The value of a node is the value of the nodes that point to it.

• How do we implement that?
  • Assume that we have a unit of authority to distribute to all nodes.
  • Each node distributes the authority value they have to their neighbors
  • The authority value of each node is the sum of the authority fractions it collects from its neighbors.
  • Solving the system of equations we get the authority values for the nodes
    • \( w = \frac{1}{2} , \ w = \frac{1}{4} , \ w = \frac{1}{4} \)
A more complex example

\[ w_1 = \frac{1}{3} w_4 + \frac{1}{2} w_5 \]
\[ w_2 = \frac{1}{2} w_1 + w_3 + \frac{1}{3} w_4 \]
\[ w_3 = \frac{1}{2} w_1 + \frac{1}{3} w_4 \]
\[ w_4 = \frac{1}{2} w_5 \]
\[ w_5 = w_2 \]

\[ w_v = \sum_{u \to v} \frac{1}{d_{out}(u)} w_u \]
Random Walks on Graphs

• What we described is equivalent to a random walk on the graph

• Random walk:
  • Start from a node uniformly at random
  • Pick one of the outgoing edges uniformly at random
  • Move to the destination of the edge
  • Repeat.
Example

- Step 0
Example

- Step 0
Example

- Step 1
Example

• Step 1
Example

• Step 2
Example

• Step 2
Example

- Step 3
Example

• Step 3
Example

- Step 4…
Memorylessness

• Question: what is the probability $p_i^t$ of being at node $i$ after $t$ steps?

\[
p_1^t = \frac{1}{3} p_4^{t-1} + \frac{1}{2} p_5^{t-1}
\]
\[
p_2^t = \frac{1}{2} p_1^{t-1} + p_3^{t-1} + \frac{1}{3} p_4^{t-1}
\]
\[
p_3^t = \frac{1}{2} p_1^{t-1} + \frac{1}{3} p_4^{t-1}
\]
\[
p_4^t = \frac{1}{2} p_5^{t-1}
\]
\[
p_5^t = p_2^{t-1}
\]

• **Memorylessness property**: The next node on the walk depends only at the current node and not on the past of the process
Transition probability matrix

- Since the random walk process is memoryless we can describe it with the transition probability matrix.
- Transition probability matrix: A matrix $P$, where $P[i, j]$ is the probability of transitioning from node $i$ to node $j$
  \[ P[i, j] = 1 / \deg_{out}(i) \]

- Matrix $P$ has the property that the entries of all rows sum to 1
  \[ \sum_j P[i, j] = 1 \]

- A matrix with this property is called stochastic.
An example

\[
A = \begin{bmatrix}
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

\[
P = \begin{bmatrix}
0 & 1/2 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1/3 & 1/3 & 1/3 & 0 & 0 \\
1/2 & 0 & 0 & 1/2 & 0 \\
\end{bmatrix}
\]
Node Probability vector

• The vector $p^t = (p^t_1, p^t_2, \ldots, p^t_n)$ that stores the probability of being at node $v_i$ at step $t$

• $p^0_i$ = the probability of starting from state $i$ (usually set to uniform)

• We can compute the vector $p^t$ at step $t$ using a vector-matrix multiplication

$$p^t = p^{t-1} P$$
An example

\[
P = \begin{bmatrix}
0 & 1/2 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 \\
1/3 & 1/3 & 1/3 & 0 & 0 \\
1/2 & 0 & 0 & 1/2 & 0 \\
\end{bmatrix}
\]

\[
p_1^t = \frac{1}{3} p_{4}^{t-1} + \frac{1}{2} p_{5}^{t-1}
\]

\[
p_2^t = \frac{1}{2} p_1^{t-1} + p_3^{t-1} + \frac{1}{3} p_4^{t-1}
\]

\[
p_3^t = \frac{1}{2} p_1^{t-1} + \frac{1}{3} p_4^{t-1}
\]

\[
p_4^t = \frac{1}{2} p_5^{t-1}
\]

\[
p_5^t = p_2^{t-1}
\]
Stationary distribution

- The **stationary distribution** of a random walk with transition matrix $P$, is a probability distribution $\pi$, such that $\pi = \pi P$.

- The stationary distribution is an **eigenvector** of matrix $P$.
  - the **principal left eigenvector** of $P$ – stochastic matrices have maximum eigenvalue 1.

- The probability $\pi_i$ is the fraction of times that we visited state $i$ as $t \to \infty$. 
Computing the stationary distribution

• The **Power Method**
  • **Initialize** to some distribution $q^0$
  • **Iteratively** compute $q^t = q^{t-1}P$
  • After many iterations $q^t \approx \pi$ regardless of the initial vector $q^0$
  • Power method because it computes $q^t = q^0P^t$

• Rate of convergence
  • determined by the second eigenvalue $\lambda_2^t$
The stationary distribution

• What is the meaning of the stationary distribution \( \pi \) of a random walk?

• \( \pi(i) \): the probability of being at node \( i \) after very large (infinite) number of steps

• \( \pi = p_0 P^\infty \), where \( P \) is the transition matrix, \( p_0 \) the original vector
  • \( P(i, j) \): probability of going from \( i \) to \( j \) in one step
  • \( P^2(i, j) \): probability of going from \( i \) to \( j \) in two steps (probability of all paths of length 2)
  • \( P^\infty(i, j) = \pi(j) \): probability of going from \( i \) to \( j \) in infinite steps – starting point does not matter.
The PageRank random walk

- Vanilla random walk
  - make the adjacency matrix stochastic and run a random walk

\[
P = \begin{bmatrix}
  0 & 1/2 & 1/2 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 1 \\
  0 & 1 & 0 & 0 & 0 & 0 \\
  1/3 & 1/3 & 1/3 & 0 & 0 & 0 \\
  1/2 & 0 & 0 & 1/2 & 0 & 0
\end{bmatrix}
\]
The PageRank random walk

• What about sink nodes?
  • what happens when the random walk moves to a node without any outgoing links?

\[
P = \begin{bmatrix}
  0 & 1/2 & 1/2 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 & 0 \\
  1/3 & 1/3 & 1/3 & 0 & 0 \\
  1/2 & 0 & 0 & 1/2 & 0
\end{bmatrix}
\]
The PageRank random walk

- Replace these row vectors with a vector $v$
  - typically, the uniform vector

$$P' = P + dv^T$$

$$d = \begin{cases} 
1 & \text{if } i \text{ is sink} \\
0 & \text{otherwise}
\end{cases}$$
The PageRank random walk

- What about loops?
  - Spider traps
The PageRank random walk

- Add a random jump to vector \( v \) with prob \( 1-\alpha \)
  - typically, to a uniform vector
- Restarts after \( 1/(1-\alpha) \) steps in expectation
  - Guarantees irreducibility, convergence

\[
P'' = \alpha \begin{bmatrix}
0 & 1/2 & 1/2 & 0 & 0 \\
1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\
1/3 & 1/3 & 1/3 & 0 & 0 \\
1/2 & 0 & 0 & 0 & 1/2 \\
\end{bmatrix}
+ (1 - \alpha) \begin{bmatrix}
1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\
1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\
1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\
1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\
\end{bmatrix}
\]

\( P'' = \alpha P' + (1-\alpha)uv^T \), where \( u \) is the vector of all 1s

Random walk with restarts
PageRank algorithm [BP98]

- The Random Surfer model
  - pick a page at random
  - with probability $1 - \alpha$ jump to a random page
  - with probability $\alpha$ follow a random outgoing link
- Rank according to the stationary distribution

$$PR(p) = \alpha \sum_{q \rightarrow p} \frac{PR(q)}{|Out(q)|} + (1 - \alpha) \frac{1}{n}$$

$\alpha = 0.85$ in most cases
Stationary distribution with random jump

• If \( v \) is the jump vector

\[
\begin{align*}
p^0 &= v \\
p^1 &= \alpha p^0 P + (1 - \alpha)v = \alpha v P + (1 - \alpha)v \\
p^2 &= \alpha p^1 P + (1 - \alpha)v = \alpha^2 v P^2 + (1 - \alpha)v \alpha P + (1 - \alpha)v \\
&\vdots \\
p^\infty &= (1 - \alpha)v + (1 - \alpha)v \alpha P + (1 - \alpha)v \alpha^2 P^2 + \cdots \\
&= (1 - \alpha)(I - \alpha P)^{-1}
\end{align*}
\]

• With the random jump the shorter paths are more important, since the weight decreases exponentially
  • makes sense when thought of as a restart

• If \( v \) is not uniform, we can bias the random walk towards the nodes that are close to \( v \)
  • Personalized and Topic-Specific Pagerank.
Effects of random jump

• Guarantees convergence to unique distribution
• Motivated by the concept of random surfer
• Offers additional flexibility
  • personalization
  • anti-spam
• Controls the rate of convergence
  • the second eigenvalue of matrix $P''$ is $\alpha$
Random walks on undirected graphs

• For undirected graphs, the stationary distribution is proportional to the degrees of the nodes
  • Thus in this case a random walk is the same as degree popularity

• This is not longer true if we do random jumps
  • Now the short paths play a greater role, and the previous distribution does not hold.
Pagerank implementation

- Store the graph in adjacency list, or list of edges
- Keep current pagerank values and new pagerank values
- Go through edges and update the values of the destination nodes.
- Repeat until the difference ($L_1$ or $L_\infty$ difference) is below some small value $\varepsilon$. 
A (Matlab-friendly) PageRank algorithm

- Performing vanilla power method is now too expensive – the matrix is not sparse

\[ q^0 = v \]

\[ t = 1 \]

repeat

\[ q^t = (P'')^T q^{t-1} \]

\[ \delta = \left\| q^t - q^{t-1} \right\| \]

\[ t = t + 1 \]

until \( \delta < \varepsilon \)

Efficient computation of \( y = (P'')^T x \)

\[ y = aP^T x \]

\[ \beta = \| x \|_1 - \| y \|_1 \]

\[ y = y + \beta v \]

\( P = \) normalized adjacency matrix

\( P' = P + dv^T, \) where \( d_i \) is 1 if \( i \) is sink and 0 o.w.

\( P'' = \alpha P' + (1-\alpha)uv^T, \) where \( u \) is the vector of all 1s
Pagerank history

• Huge advantage for Google in the early days
  • It gave a way to get an idea for the value of a page, which was useful in many different ways
    • Put an order to the web.
  • After a while it became clear that the anchor text was probably more important for ranking
  • Also, link spam became a new (dark) art

• Flood of research
  • Numerical analysis got rejuvenated
  • Huge number of variations
  • Efficiency became a great issue.
  • Huge number of applications in different fields
    • Random walk is often referred to as PageRank.
The HITS algorithm

- Another algorithm proposed around the same time as Pagerank for using the hyperlinks to rank pages
  - Kleinberg: then an intern at IBM Almaden
  - IBM never made anything out of it
Query dependent input

Root set obtained from a text-only search engine

Root Set
Query dependent input

IN  Root Set  OUT
Query dependent input

IN

Root Set

OUT
Query dependent input

IN

Root Set

OUT

Base Set
Hubs and Authorities [K98]

- Authority is not necessarily transferred directly between authorities
- Pages have double identity
  - *hub* identity
  - *authority* identity
- **Good** hubs point to **good** authorities
- **Good** authorities are pointed by **good** hubs
Hubs and Authorities

- Two kind of weights:
  - Hub weight
  - Authority weight

- The hub weight is the sum of the authority weights of the authorities pointed to by the hub

- The authority weight is the sum of the hub weights that point to this authority.
**HITS Algorithm**

- Initialize all weights to 1.
- Repeat until convergence
  - \( O \) operation: hubs collect the weight of the authorities
    \[
    h_i = \sum_{j:i \rightarrow j} a_j
    \]
  - \( I \) operation: authorities collect the weight of the hubs
    \[
    a_i = \sum_{j:j \rightarrow i} h_j
    \]
- Normalize weights under some norm
HITS and eigenvectors

• The HITS algorithm is a power-method eigenvector computation
  • in vector terms $a^t = A^T h^{t-1}$ and $h^t = A a^{t-1}$
  • so $a^t = A^T A a^{t-1}$ and $h^t = A A^T h^{t-1}$
  • The authority weight vector $a$ is the eigenvector of $A^T A$ and the hub weight vector $h$ is the eigenvector of $A A^T$

• The vectors $a$ and $h$ are singular vectors of the matrix $A$
Example

Initialize

1  □  □  □  □  □  1
1  □  □  □  □  □  1
1  □  □  □  □  □  1
1  □  □  □  □  □  1
1  □  □  □  □  □  1
1  □  □  □  □  □  1

hubs  authorities
Example

Step 1: O operation
Example

Step 1: I operation

```
 1  □ □ □ □  □ □  6
 2  □ □ □ □  □ □  5
 3  □ □ □ □  □ □  5
 2  □ □ □ □  □ □  2
 1  □ □ □ □  □ □  1
  hubs  □ □ □ □  □ □  authorities
```
Example

Step 1: Normalization (Max norm)

1/3 -> 5/6
2/3 -> 5/6
1 -> 5/6
2/3 -> 2/6
1/3 -> 1/6

hubs -> authorities
Example

Step 2: O step

1 → 1
11/6 → 5/6
16/6 → 5/6
7/6 → 2/6
1/6 → 1/6

hubs → authorities
Example

Step 2: I step

1 1
11/6 11/6
16/6 16/6
7/6 7/6
1/6 1/6

hubs authorities
Example

Step 2: Normalization

hubs

1/16 → 7/16 → 1 → 1

6/16 → 11/16 → 1 → 27/33

1

1 → 11/16 → 1

7/16 → 23/33

authorities

7/33

1/33
Example

Convergence

| 0.4 | 1 |
| 0.75 | 0.8 |
| 1 | 0.6 |
| 0.3 | 0.14 |
| 0 | 0 |

hubs → authorities
OTHER ALGORITHMS
The SALSA algorithm [LM00]

- Perform a random walk alternating between hubs and authorities

- What does this random walk converge to?

- The graph is essentially undirected, so it will be proportional to the degree.
Social network analysis

• Evaluate the **centrality** of individuals in social networks
  
  • **degree centrality**
    • the (weighted) degree of a node
  
  • **distance centrality**
    • the average (weighted) distance of a node to the rest in the graph
      \[
      D_c(v) = \frac{1}{\sum_{u \neq v} d(v, u)}
      \]
  
  • **betweenness centrality**
    • the average number of (weighted) shortest paths that use node \(v\)
      \[
      B_c(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}
      \]
The importance of a node is measured by the weighted sum of paths that lead to this node.

\[ A^m[i,j] = \text{number of paths of length } m \text{ from } i \text{ to } j \]

Compute

\[ P = bA + b^2A^2 + \cdots + b^mA^m + \cdots = (I - bA)^{-1} - I \]

converges when \( b < \lambda_1(A) \)

Rank nodes according to the column sums of the matrix \( P \)
Bibliometrics

- Impact factor (E. Garfield 72)
  - counts the number of citations received for papers of the journal in the previous two years

- Pinsky-Narin 76
  - perform a random walk on the set of journals
  - $P_{ij} =$ the fraction of citations from journal $i$ that are directed to journal $j$