INTRODUCTION TO DATA SCIENCE

PageRank
OUTLINE

• Introduction

• The Basic Idea

• The Initial PageRank Model

• The Human Surfer Model

• Advanced Aspects

• Alternative Model
WHY PAGERANK?

- The major challenge of web search engines is to rank the retrieved pages.
- Most users don’t go beyond the 1-2 first pages of search results.
- First generation search engine (AltaVista) ranked results based on keywords and relevance measures.
- Easy to manipulate.
- Google introduced “link analysis” as a tool for evaluating page “quality”.
- Hyperlink-Induced Topic Search (HITS) - hubs and authorities.
PageRank is an example of **unsupervised** learning—it evaluates page quality without a training set.

<table>
<thead>
<tr>
<th>Supervised Learning</th>
<th>Unsupervised Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete</td>
<td></td>
</tr>
<tr>
<td>classification or categorization</td>
<td>PageRank</td>
</tr>
<tr>
<td>Continuous</td>
<td>Continuous</td>
</tr>
<tr>
<td>regression</td>
<td>dimensionality reduction</td>
</tr>
</tbody>
</table>
THE WEB AS DIRECTED GRAPH

Back link of u
Forward link of u

Link from i to m

Page i → Page m → Page u → Page y
Page n → Page u
Page w → Page v

Set B_u
Rank of page u is R(u)
Set F_u
Amount N_u = |F_u|
BACK LINKS AS INITIAL IDEA

• Citation analysis as basis
• Idea: Pages with a lot of back links are more important
• Intuitive approach
  \[ R(u) = \sum_{v \in B_u} 1 \]
• Extension: Each page has a “vote” of 1
  \[ R(u) = c \sum_{v \in B_u} \frac{1}{N_v} \]
• c normalizing factor (here c=1)
FROM ANALYZING **BACK LINKS** TO **PAGERANK**

*Back links*

- Easy to calculate
- Suitable for well-controlled documents such as scientific articles
- For web pages: manipulation is easy
- Not in line with the common sense notion of “relevance”

*PageRank*

- Extension of the simple analysis of *back links*
- Idea: Include the relevance of the referring (*back-link*) pages in the calculations of the ranks
- Manipulations are more difficult
OUTLINE

- Introduction
- The Basic Idea
- The Initial PageRank Model
- The Human Surfer Model
- Advanced Aspects
- Alternative Model
INTUITIVE DEFINITION OF PAGERANKS

\[ R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v} \]

- Rank spread evenly among the forward links
- Recursive calculation of \( R(u) \) until there is convergence
- Factor \( c \)
  - For normalization
  - Usually \( c > 1 \), as there are pages without links
MATHEMATICALLY, THIS IS AN EIGENVECTOR PROBLEM

- Web as matrix $A$
  - if there is an edge between $u$ and $v$ (i.e., a link from $u$ to $v$)
    \[ A_{u,v} = \frac{1}{N_u} \]
  - else
    \[ A_{u,v} = 0 \]

\[
A = \begin{bmatrix}
0,0 & 0,5 & 0,5 \\
0,0 & 0,0 & 1,0 \\
1,0 & 0,0 & 0,0 \\
\end{bmatrix}
\]

\[ R = (0,4 \ 0,2 \ 0,4) \]

- $R$ vector of page ranks
- This is the left eigenvector of $A$ to the eigenvalue $c$
  - $R = RA_c$
EIGENVECTORS AND EIGENVALUES

Definitions

Consider the square matrix $A$.

We say that $c$ is an **eigenvalue** of $A$ if there exists a non-zero vector $x$ such that $Ax = cx$.

In this case, $x$ is called a (right) **eigenvector** (corresponding to $c$), and the pair $(c, x)$ is called an **eigenpair** for $A$.

Right eigenvectors satisfy the equation $Ax = xc$.

$c_1$ is called the **dominant** eigenvalue if

$$
|c_1| \geq |c_2| \geq |c_3| \geq \ldots \geq |c_n|
$$

Example

The matrix; $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ has two eigenvectors:

$v_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ and $v_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

with eigenvalues 1 and 3 respectively.
SOLVING THE EIGENVALUE PROBLEM

Algebraic Approaches

• Various Methods

• Example: calculate the determinant

\[
\text{det}(A-cI) = 0
\]
\[
I = \text{Identity Matrix}
\]
\[
A = \begin{pmatrix}
0,0 & 0,5 & 0,5 \\
0,0 & 0,0 & 1,0 \\
1,0 & 0,0 & 0,0
\end{pmatrix}
\]
\[
\text{det}(A-cI)\]
\[
= -c^3 + 0,5 + 0 + 0,5c - 0 - 0 = 0
\]
\[
c = 1
\]

Power Iterations

• Principle

\[
x = \text{any vector with } ||x|| = 1
\]
\[
\text{eps} = \text{any value } < 1
\]
while(\text{psi} > \text{eps} < 1)
\[
\text{xTemp} = x
\]
\[
x = x \times A \quad \text{// multiply } A
\]
\[
x = x / ||x|| \quad \text{// normalise}
\]
\[
c = x^T \times A \times x \quad \text{// eigenvalue}
\]
\[
\text{psi} = ||x - \text{xTemp}||_2
\]
wend

where \( ||\bullet||_2 \) = Euclid norm

• In case of a stochastic matrix \( A \)
PROBLEMS WITH "IMPERFECT" GRAPHS

Perfect Graph

Graph with Rank Sink

Graph with dangling link (Rank Leak)

Dangling Link
**RANK SOURCE SOLVES RANK SINKS**

**Introduction to Rank Source**

- **E(u): vector of web pages**
  
  \[ R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u) \]

- where \( c \rightarrow \text{max} \)
  
  \[ \|R'\|_1 = \sum_i |x_i| = 1 \]

- **As eigenvalue problem:**
  
  \[ R' = c(A + E \otimes 1)R' \]

  where \( l = (1,1,...,1) \)

- **Simplified Version:**
  - Same Rank Source for all pages
  - Normalisation to 1
  - New formula:

  \[ R''(u) = d \sum_{v \in B_u} \frac{R''(v)}{N_v} + \frac{(1-d)}{\# \text{ Pages}} \]
DANGLING LINKS

- Reduce “distributable” PageRank
- Rather frequent
  - Pages without links
  - Pages not yet indexed by Google
  - PDFs etc.

- Removed prior to calculation
- Added with the immediate page rank after the final iteration

- Result hardly affected
OUTLINE

• Introduction
• The Basic Idea
• The Initial PageRank Model
• The Human Surfer Model
• Advanced Aspects
• Alternative Model
**MARKOV CHAIN**

- **Homogeneous discrete stochastic process** with transition matrix $P$
  - Transitions depend only on the current state (Markov property)
  - Transitions from node $i$ to node $k$ happen at discrete points of time $t=1,2,…$
  - Transition from node $i$ to node $k$ happens with probability $P_{ik}$
  - The transition probability is independent of the time $t$ (homogeneous)
  - The initial node is selected arbitrarily based on a distribution $q^0$ over $V$
  - $q^t$: row vector, whose $k$-th entry gives the likelihood of being in state $k$ after transition $t$

- It holds:

$$q^{t+1} = q^t P \Leftrightarrow q^{t+1} = qP^t$$

\[
P = \begin{pmatrix}
0.0 & 0.5 & 0.5 \\
0.0 & 0.0 & 1.0 \\
1.0 & 0.0 & 0.0
\end{pmatrix}
\]

\[
q^0 = \begin{pmatrix}
1.0 & 0.0 & 0.0
\end{pmatrix}
\]

\[
q^1 = q^0 P = \begin{pmatrix}
0.0 & 0.5 & 0.5
\end{pmatrix}
\]
Limit Distribution

\[ \lim_{n \to \infty} q^0 P^n = \lim_{n \to \infty} q^n P \]

- Intuition:
  Both states equally likely

- \( q^0 = (1,0) \) leads to
  - \( q^{2n} = (0,1) \)
  - \( q^{2n+1} = (1,0) \)

- does not always exist
- can depend on the initial distribution
- is not necessarily unique
PROPERTIES OF MARKOV CHAINS

• **Irreducibility:**
  - Any node of a Markov Chain can be reached from any node (in a finite number of steps).

• **Aperiodicity:**
  - The greatest common divisor of the length of all „round-trips“ is 1.
Wanted

- Stationary distribution such that: \( q^\infty = q^\infty P \)
- i.e., the eigenvector to the eigenvalue 1

Theorem

- Assume that \( P \) is
  - irreducible
  - aperiodic
  - finite
- Then there is a unique stationary distribution \( q^\infty \)
- Let \( N(i,t) \) be the number of visits that a random surfer pays to page \( i \) until the point in time \( t \). Then

\[
\lim_{t \to \infty} \frac{N(i,t)}{t} = q^\infty_i
\]
The web surfer starts at a randomly selected page
At each period the surfer chooses between the following alternatives:
- Follow a randomly selected link on the current page (probability \( d \))
- Jump to another page of the web without following a link (probability \((1-d)\))

\[
A' = dA + (1-d) \frac{1}{\text{Pages}} \times 1 \times 1
\]
TRANSITION MATRIX

\[ A' = \begin{pmatrix}
0 & 0.4 & 0.4 & 0 & 0 \\
0 & 0 & 0.4 & 0.4 & 0 \\
0.8 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.8 \\
0 & 0 & 0 & 0.8 & 0 \\
\end{pmatrix} \]

\[ A = \begin{pmatrix}
0 & 0.5 & 0.5 & 0 & 0 \\
0 & 0 & 0.5 & 0.5 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
\end{pmatrix} \]

\[ d = 0.8 \]

\[ A' + (1-d) \frac{1}{\text{# Pages}} \]

\[ (1-d) \frac{1}{\text{# Pages}} \]

\[ 1 \times 1 \]
• *Steady State* is a distribution vector satisfying

\[ R = RA' \]

• Can be regarded as a special form of

\[ R' = cR'(A + E \otimes 1) \]

  – Normalised to 1
  – *Rank of Source* same for all pages

• *Dangling Links*

  – Can either be removed
  – Or be treated as a page linking to all other pages
THE ROLE OF D

- \( d = 0.85 \)
- E equally distributed
- *Dangling Links* added for final iteration

- \( d = 0 \)
- E equally distributed
- *Dangling Links* added for final iteration

- \( d \leq 1 \)
- E only for one page, e.g. private home page
- *Dangling Links* added for final iteration

- \( d = 0.85 \) reportedly used by Google (at least initially)
- Probably what Google does
- Additional adaptations are applied, algorithm is optimized

- Extreme case: All pages are equally likely
- Assumes that all pages are equally important
- Comparable to the simple search engines

- Mirrors user preferences
- Assumes that the page is representative
- Alternatively one could derive E from historic user behaviour (e.g., using web logs)
NON-UNIFORM TELEPORTATION

Sports

Teleport with 10% probability to a Sports page
OUTLINE

• Introduction
• The Basic Idea
• The Initial PageRank Model
• The Human Surfer Model
• Advanced Aspects
• Alternative Model
CONVERGENCE & RUNTIME OF POWER ITERATION

• Convergence ensured by adapting the transition matrix
• The number of required iterations
  – Depends on the distance to second eigenvalue and thus the value $d$
  – Is less affected by the number of links
• Google calculates PageRank regularly, updates are released appr. every day

Convergence ($d=0.85$)

![Graph showing convergence and number of iterations]

Source: Brin+: The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, Stanford University, 1999
Artificial creation of back links
- Across domains
- Linked

Purchasing links
- E.g., Banner on a page with high Page Rank

Create Google-tailored pages
- Multiple linked pages
- Links to bad pages using JavaScript

- Theoretically possible
- Anti-Spamming mechanisms exist
  - PageRank 0
  - BadRank

- Possible
- Costs money, so a bit controlled

- To a certain extent feasible
- Too much might lead to exclusion from page rank calculation
AGENDA

• Introduction
• The Basic Idea
• The Initial PageRank Model
• The *Human Surfer* Model
• Advanced Aspects
• *Alternative Model*
**Hypertext Included Topic Selection**

- Web as directed graph
- Algorithm operates on a part of the graph
- Algorithm runs subject-specific and distinguishes
  - “expert” pages (Authorities) for a topic
  - pages linking to Authorities (Hubs)
- HITS is based on balance of Hubs and Authorities

**Salsa**

- Extends HITS for probabilities
- undirected graph
- *Hub Walk* and *Authority Walk*
HIGH-LEVEL SCHEME

Extract from the web a base set of pages that could be good hubs or authorities.

From these, identify a small set of top hub and authority pages; iterative algorithm.
BASE SET

Given text query (say soccer), use a text index to get all pages containing soccer.

• Call this the root set of pages.

Add in any page that either

• points to a page in the root set, or
• is pointed to by a page in the root set.

Call this the base set.
ASSEMBLING THE BASE SET

• Root set typically 200-1000 nodes.

• Base set may have up to 5000 nodes.

• How do you find the base set nodes?
  • Follow out-links by parsing root set pages.
  • Get in-links (and out-links) from a \textit{connectivity server}.
  • (Actually, suffices to text-index strings of the form \texttt{href= "URL"} to get in-links to \texttt{URL}.)
DISTILLING HUBS AND AUTHORITIES

Compute, for each page $x$ in the base set, a hub score $h(x)$ and an authority score $a(x)$.

1. Initialize: for all $x$, $h(x) \leftarrow 1$; $a(x) \leftarrow 1$;

2. Iteratively update all $h(x), a(x)$;

3. After iterations
   1. output pages with highest $h()$ scores as top hubs
   2. highest $a()$ scores as top authorities.
ITERATIVE UPDATE

Repeat the following updates, for all \( x \):

\[
h(x) \leftarrow \sum_{y} a(y)
\]

\[
a(x) \leftarrow \sum_{y} h(y)
\]
SCALING

To prevent the $h()$ and $a()$ values from getting too big, can scale down after each iteration.

Scaling factor doesn’t really matter:

- we only care about the relative values of the scores.
• Claim: relative values of scores will converge after a few iterations:
  • in fact, suitably scaled, \( h() \) and \( a() \) scores settle into a steady state!
• We only require the relative orders of the \( h() \) and \( a() \) scores - not their absolute values.
• In practice, \( \sim 5 \) iterations get you close to stability.
THINGS TO NOTE

- Pulled together good pages regardless of language of page content.

- Use only link analysis after base set assembled
  - iterative scoring is query-independent.

- Iterative computation after text index retrieval - significant overhead.
PROOF OF CONVERGENCE

$n \times n$ adjacency matrix $A$:

- each of the $n$ pages in the base set has a row and column in the matrix.
- Entry $A_{ij} = 1$ if page $i$ links to page $j$, else $= 0$.  

\[ 
\begin{array}{ccc}
1 & 2 & 3 \\
1 & 0 & 1 & 0 \\
2 & 1 & 1 & 1 \\
3 & 1 & 0 & 0 \\
\end{array} \]
HUB/AUTHORITY VECTORS

View the hub scores \( h() \) and the authority scores \( a() \) as vectors with \( n \) components.

Recall the iterative updates

\[
h(x) \leftarrow \sum_{y \leftarrow x} a(y)
\]

\[
a(x) \leftarrow \sum_{y \leftarrow x} h(y)
\]
REWRITE IN MATRIX FORM

• $h = Aa$.
• $a = A^t h$.

• Substituting, $h = AA^t h$ and $a = A^t Aa$.

• Thus, $h$ is an eigenvector of $AA^t$ and $a$ is an eigenvector of $A^t A$.

• Further, our algorithm is a particular, known algorithm for computing eigenvectors: the power iteration method.

Guaranteed to converge.
ISSUES

Topic Drift

- Off-topic pages can cause off-topic “authorities” to be returned
  - E.g., the neighborhood graph can be about a “super topic”
- Mutually Reinforcing Affiliates
  - Affiliated pages/sites can boost each others’ scores
  - Linkage between affiliated pages is not a useful signal
LITERATURE


SLIDES CAN BE FOUND AT:
TEACHINGDATASCIENCE.ORG