Similarity & Link Analysis

Stony Brook University
CSE545, Fall 2016
Finding Similar “Items”

Real World

Digital World

Records / Mentions

(http://blog.soton.ac.uk/hive/2012/05/10/recommendation-system-of-hive/)

(http://www.datacommunitydc.org/blog/2013/08/entity-resolution-for-big-data)
Finding Similar “Items”: What we will cover

- Shingling
- Minhashing
- Locality-sensitive hashing
- Distance Metrics
Document Similarity

**Challenge:** How to represent the document in a way that can be efficiently encoded and compared?
Shingles

Goal: Convert documents to sets
Shingles

Goal: Convert documents to sets

**k-shingles** (aka “character n-grams”) - sequence of $k$ characters

E.g. $k=2$ doc=”abcdabd”
    singles(doc, 2) = \{ab, bc, cd, da, bd\}
Shingles

Goal: Convert documents to sets

**k-shingles** (aka “character n-grams”) - sequence of $k$ characters

E.g. $k=2$ doc="abcdabd"

$singles(doc, 2) = \{ab, bc, cd, da, bd\}$

- Similar documents have many common shingles
- Changing words or order has minimal effect.
- In practice use $5 < k < 10$
Shingles

Goal: Convert documents to sets

- Similar documents have many common shingles
- Changing words or order has minimal effect.
- In practice use $5 < k < 10$

Large enough that any given shingle appearing a document is highly unlikely (e.g. < .1% chance)

Can hash large shingles to smaller (e.g. 9-shingles into 4 bytes)

Can also use words (aka n-grams).
Shingles

**Problem:** Even if hashing, sets of shingles are large (e.g. 4 bytes => 4x the size of the document).
Minhashing

Goal: Convert sets to shorter ids, signatures
Minhashing - Background

Goal: Convert sets to shorter ids, signatures

Characteristic Matrix, $X$:

<table>
<thead>
<tr>
<th>Element</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$b$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$c$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$d$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$e$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

(Jeskev et al., 2014; http://www.mmds.org/)

often very sparse! (lots of zeros)

Jaccard Similarity:

$$sim(S_1, S_2) = \frac{S_1 \cap S_2}{S_1 \cup S_2}$$
Minhashing - Background

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>bc</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>de</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ah</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ha</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ed</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ca</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Jaccard Similarity:

$$\text{sim}(S_1, S_2) = \frac{S_1 \cap S_2}{S_1 \cup S_2}$$
**Minhashing - Background**

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ab</strong></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>bc</strong></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td><strong>de</strong></td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>ah</strong></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>ha</strong></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>ed</strong></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>ca</strong></td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Jaccard Similarity:**

$$sim(S_1, S_2) = \frac{S_1 \cap S_2}{S_1 \cup S_2}$$
## Minhashing - Background

### Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>bc</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>de</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ah</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ha</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ed</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ca</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

### Jaccard Similarity:

\[
sim(S_1, S_2) = \frac{S_1 \cap S_2}{S_1 \cup S_2}
\]

\[
sim(S_1, S_2) = \frac{3}{6}
\]

# both have / # at least one has
Shingles

**Problem:** Even if hashing, sets of shingles are large (e.g. 4 bytes => 4x the size of the document).
Minhashing

Characteristic Matrix: $X$

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>bc</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>de</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ah</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ha</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ed</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ca</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(Leskovec et al., 2014; http://www.mmds.org/)

Approximate Approach:
1) Instead of keeping whole characteristic matrix, just keep first row where 1 is encountered.

2) Shuffle and repeat to get a “signature” for each set.

Idea: We don’t need to actually shuffle we can just use hash functions.
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>bc</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>de</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ah</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ha</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ed</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ca</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Minhash function: $h$

- Based on permutation of rows in the characteristic matrix, $h$ maps sets to first row where set appears.

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>bc</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>de</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ah</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ha</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ed</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ca</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Minhash function: $h$
- Based on permutation of rows in the characteristic matrix, $h$ maps sets to first row where set appears.

(Leskovec at al., 2014; http://www.mmds.org/)
**Minhashing**

**Characteristic Matrix:**

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ab</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>bc</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>de</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>ah</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>ha</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>ed</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>ca</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Minhash function:** $h$
- Based on permutation of rows in the characteristic matrix, $h$ maps sets to first row where set appears.

(Leskovec et al., 2014; [http://www.mmds.org/](http://www.mmds.org/))
**Minhashing**

**Characteristic Matrix:**

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ab</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>bc</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>de</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>ah</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>ha</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>ed</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>ca</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Minhash function:** $h$
- Based on permutation of rows in the characteristic matrix, $h$ maps sets to first row where set appears.

$h(S_1) = ed$  #permuted row 2
$h(S_2) = ha$  #permuted row 1
$h(S_3) =$

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>(S_1)</th>
<th>(S_2)</th>
<th>(S_3)</th>
<th>(S_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ab</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>bc</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>de</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>ah</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>ha</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>ed</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>ca</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Minhash function: \(h\)
- Based on permutation of rows in the characteristic matrix, \(h\) maps sets to first row where set appears.

\(h(S_1) = ed\)  \#permuted row 2
\(h(S_2) = ha\)  \#permuted row 1
\(h(S_3) = ed\)  \#permuted row 2
\(h(S_4) = 3 4 7 6 1 2 5 1\)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ab</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>bc</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>de</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>ah</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>ha</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>ed</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>ca</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Minhash function: $h$

- Based on permutation of rows in the characteristic matrix, $h$ maps sets to first row where set appears.

$h(S_1) = ed$  #permuted row 2
$h(S_2) = ha$  #permuted row 1
$h(S_3) = ed$  #permuted row 2
$h(S_4) = ha$  #permuted row 1

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ab</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>bc</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>de</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>ah</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>ha</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>ed</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>ca</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Minhash function: $h$
- Based on permutation of rows in the characteristic matrix, $h$ maps sets to rows.

Signature matrix: $M$
- Record first row where each set had a 1 in the given permutation

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

$h_1(S_1) = ed$ #permuted row
$h_1(S_2) = ha$ #permuted row
$h_1(S_3) = ed$ #permuted row
$h_1(S_4) = ed$ #permuted row

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ab</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>bc</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>de</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>ah</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>ha</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>ed</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>ca</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Minhash function: $h$
- Based on permutation of rows in the characteristic matrix, $h$ maps sets to rows.

Signature matrix: $M$
- Record first row where each set had a 1 in the given permutation

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

$\text{Minhash function: } h$

1. $h_1(S_1) = \text{ed}$ #permuted row
2. $h_1(S_2) = \text{ha}$ #permuted row
3. $h_1(S_2) = \text{ed}$ #permuted row

(Leskovec et al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ab</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>bc</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>de</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>ah</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>ha</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>ed</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>ca</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Minhash function: $h$
- Based on permutation of rows in the characteristic matrix, $h$ maps sets to rows.

Signature matrix: $M$
- Record first row where each set had a 1 in the given permutation

$\begin{array}{cccc}
S_1 & S_2 & S_3 & S_4 \\
h_1 & 2 & 1 & 2 & 1 \\
\end{array}$

1. $h_1(S_2) = ha$ #permuted row
2. $h_1(S_1) = ed$ #permuted row
3. $\text{(Leskovec et al., 2014; http://www.mmds.org/)}$
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
<td>ab</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>bc</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>de</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>ah</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>ha</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>ed</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>ca</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Minhash function: $h$
- Based on permutation of rows in the characteristic matrix, $h$ maps sets to rows.

Signature matrix: $M$
- Record first row where each set had a 1 in the given permutation
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
<td>ab</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>bc</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>de</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>ah</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>ha</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>ed</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>ca</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Minhash function: $h$
- Based on permutation of rows in the characteristic matrix, $h$ maps sets to rows.

Signature matrix: $M$
- Record first row where each set had a 1 in the given permutation

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$h_2$</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Minhash function: $h$
- Based on permutation of rows in the characteristic matrix, $h$ maps sets to rows.

Signature matrix: $M$
- Record first row where each set had a 1 in the given permutation

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$h_2$</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>$h_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Leskovec et al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>S₁</th>
<th>S₂</th>
<th>S₃</th>
<th>S₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>ab</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>bc</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td>de</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6</td>
<td>ah</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td>ha</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2</td>
<td>ed</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td>ca</td>
<td>1</td>
</tr>
</tbody>
</table>

Minhash function: h
- Based on permutation of rows in the characteristic matrix, h maps sets to rows.

Signature matrix: M
- Record first row where each set had a 1 in the given permutation

<table>
<thead>
<tr>
<th></th>
<th>S₁</th>
<th>S₂</th>
<th>S₃</th>
<th>S₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>h₁</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>h₂</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>h₃</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix: $X$

Minhash function: $h$
- Based on permutation of rows in the characteristic matrix, $h$ maps sets to rows.

Signature matrix: $M$
- Record first row where each set had a 1 in the given permutation

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>ab</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>bc</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td>de</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6</td>
<td>ah</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td>ha</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2</td>
<td>ed</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td>ca</td>
<td>1</td>
</tr>
</tbody>
</table>

Property of signature matrix:
The probability for any $h_i$ (i.e. any row), that $h_i(S_1) = h_i(S_2)$ is the same as $\text{Sim}(S_1, S_2)$

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>ab</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>bc</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td>de</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6</td>
<td>ah</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td>ha</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2</td>
<td>ed</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td>ca</td>
<td>1</td>
</tr>
</tbody>
</table>

Minhash function:
Based on permutation of rows in the characteristic matrix, $h$ maps sets to rows.

Signature matrix:
Record first row where each set had a 1 in the given permutation

Property of signature matrix:
The probability for any $h_i$ (i.e. any row), that $h_i(S_1) = h_i(S_2)$ is the same as $\text{Sim}(S_1, S_2)$

Thus, similarity of signatures $S_1, S_2$ is the fraction of minhash functions (i.e. rows) in which they agree.

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Property of signature matrix:
The probability for any $h_i$ (i.e. any row), that $h_i(S_1) = h_i(S_2)$ is the same as $\text{Sim}(S_1, S_2)$

Thus, similarity of signatures $S_1, S_2$ is the fraction of minhash functions (i.e. rows) in which they agree.

Estimate with a random sample of permutations (i.e. ~100)

(Leskovec et al., 2014; http://www.mmds.org/)
### Minhashing

#### Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>ab</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>6</td>
<td>an</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>ha</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td>ca</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2</td>
<td>ed</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6</td>
<td>an</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Minhash function:

$h_i$ based on permutation of rows in the characteristic matrix. $h_i$ maps sets to rows.

#### Signature matrix:

$M$ records the first row where each set had a 1 in the given permutation.

<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$h_2$</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>$h_3$</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

#### Property of signature matrix:

The probability for any $h_i$ (i.e. any row), that $h_i(S_1) = h_i(S_2)$ is the same as $\text{Sim}(S_1, S_2)$.

Thus, similarity of signatures $S_1$, $S_2$ is the fraction of minhash functions (i.e. rows) in which they agree.

#### Estimate with a random sample of permutations (i.e. ~100):

$\text{Estimated } \text{Sim}(S_1, S_3) = \frac{\text{agree}}{\text{all}} = \frac{2}{3}$

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>S₁</th>
<th>S₂</th>
<th>S₃</th>
<th>S₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>ab</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>bc</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td>de</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6</td>
<td>ah</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td>ha</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2</td>
<td>ed</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td>ca</td>
<td>1</td>
</tr>
</tbody>
</table>

Minhash function:

Based on permutation of rows in the characteristic matrix, \( h \) maps sets to rows.

Signature matrix:

Record first row where each set had a 1 in the given permutation

\[ S_1 \quad S_2 \quad S_3 \quad S_4 \]

\[ h_1 \quad 2 \quad 1 \quad 2 \quad 1 \]

\[ h_2 \quad 2 \quad 1 \quad 4 \quad 1 \]

\[ h_3 \quad 1 \quad 2 \quad 1 \quad 2 \]

Property of signature matrix:

The probability for any \( h_i \) (i.e. any row), that \( h_i(S_1) = h_i(S_2) \) is the same as \( \text{Sim}(S_1, S_2) \)

Thus, similarity of signatures \( S_1, S_2 \) is the fraction of minhash functions (i.e. rows) in which they agree.

Estimated \( \text{Sim}(S_1, S_3) = \frac{\text{agree}}{\text{all}} = \frac{2}{3} \)

Real \( \text{Sim}(S_1, S_3) = \frac{\text{Type a}}{\text{a} + \text{b} + \text{c}} = \frac{3}{4} \)

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

Characteristic Matrix:

<table>
<thead>
<tr>
<th></th>
<th>S₁</th>
<th>S₂</th>
<th>S₃</th>
<th>S₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>ab</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>bc</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>7</td>
<td>de</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6</td>
<td>ah</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>1</td>
<td>ha</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2</td>
<td>ed</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td>ca</td>
<td>1</td>
</tr>
</tbody>
</table>

Minhash function: Based on permutation of rows in the characteristic matrix, h maps sets to rows.

Signature matrix: Record first row where each set had a 1 in the given permutation.

<table>
<thead>
<tr>
<th></th>
<th>S₁</th>
<th>S₂</th>
<th>S₃</th>
<th>S₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>h₁</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>h₂</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>h₃</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Property of signature matrix:
The probability for any $h_i$ (i.e. any row), that $h_i(S_1) = h_i(S_2)$ is the same as Sim($S_1$, $S_2$)

Thus, similarity of signatures $S_1$, $S_2$ is the fraction of minhash functions (i.e. rows) in which they agree.

Estimated Sim($S_1$, $S_3$) = agree / all = 2/3

Real Sim($S_1$, $S_3$) = Type a / (a + b + c) = 3/4

Try Sim($S_2$, $S_4$) and Sim($S_1$, $S_2$)

(Leskovec at al., 2014; http://www.mmds.org/)
Minhashing

In Practice

Problem:
- Can’t reasonably do permutations (huge space)
- Can’t randomly grab rows according to an order (random disk seeks = slow!)
Minhashing

In Practice

Problem:
- Can’t reasonably do permutations (huge space)
- Can’t randomly grab rows according to an order (random disk seeks = slow!)

Solution: Use “random” hash functions.

Setup:
- Pick \(~100\) hash functions, hashes
- Store \(M[i][s] = a\) potential minimum \(h_i(r)\)
- \#initialized to infinity (num hashes x num sets)
Minhashing

Solution: Use “random” hash functions.

● Setup:
  ○ Pick ~100 hash functions, hashes
  ○ Store $M[i][s] = \text{a potential minimum } h_i(r)$

    *initialized to infinity (num hashes x num sets)*

● Algorithm:

```python
for r in rows of cm:  # cm is characteristic matrix
    compute $h_i(r)$ for all $i$ in hashes  # precompute 100 values
    for each set $s$ in row $r$:
        if cm[r][s] == 1:
            for $i$ in hashes:  # check which hash produces smallest value
                if $h_i(r) < M[i][s]$:
                    $M[i][s] = h_i(r)$
```
Minhashing

Solution: Use “random” hash functions.

● Setup:
  ○ Pick ~100 hash functions, hashes
  ○ Store $M[i][s] = \text{a potential minimum } h_i(r)\#$initialized to infinity (num hashs x num sets)

● Algorithm:
  
  for $r$ in rows of $cm$: # cm is characteristic matrix
  compute $h_i(r)$ for all $i$ in hashes # precompute 100 values
  for each set $s$ in row $r$:
    if $cm[r][s] == 1$:
      for $i$ in hashes: # check which hash produces smallest value
        if $h_i(r) < M[i][s]$: $M[i][s] = h_i(r)$

Known as “efficient minhashing”.
Minhashing

What hash functions to use?

Start with 2 decent hash functions

e.g. \( h_a(x) = \text{ascii(string)} \% \text{large_prime_number} \)
\( h_b(x) = (3 \times \text{ascii(string)} + 16) \% \text{large_prime_number} \)

Minhashing

What hash functions to use?

Start with 2 decent hash functions

e.g. \( h_a(x) = \text{ascii(string)} \mod \text{large_prime_number} \)
\( h_b(x) = (3 \times \text{ascii(string)} + 16) \mod \text{large_prime_number} \)

Add together multiplying the second times i:

\( h_i(x) = h_a(x) + i \times h_b(x) \)
e.g. \( h_5(x) = h_a(x) + 5 \times h_b(x) \)

Minhashing

**Problem:** Even if hashing, sets of shingles are large (e.g. 4 bytes => 4x the size of the document).
Minhashing

**Problem**: Even if hashing, sets of shingles are large (e.g. 4 bytes => 4x the size of the document).

**New Problem**: Even if the size of signatures are small, it can be computationally expensive to find similar pairs.

E.g. 1m documents; 1,000,000 choose 2 = 500,000,000,000 pairs
Locality-Sensitive Hashing

**Goal:** find pairs of minhashes likely to be similar (in order to then test more precisely for similarity).

**Candidate pairs:** pairs of elements to be evaluated for similarity.
Locality-Sensitive Hashing

**Goal:** find pairs of minhashes likely to be similar (in order to then test more precisely for similarity).

**Candidate pairs:** pairs of elements to be evaluated for similarity.

If we wanted the similarity for all pairs of documents, could anything be done?
Locality-Sensitive Hashing

**Goal:** find pairs of minhashes likely to be similar (in order to then test more precisely for similarity).

**Candidate pairs:** pairs of elements to be evaluated for similarity.

**Approach:** Hash multiple times over subsets of data: similar items are likely in the same bucket once.
Locality-Sensitive Hashing

Goal: find pairs of minhashes likely to be similar (in order to then test more precisely for similarity).

Candidate pairs: pairs of elements to be evaluated for similarity.

Approach: Hash multiple times over subsets of data: similar items are likely in the same bucket once.

Approach from MinHash: Hash columns of signature matrix Candidate pairs end up in the same bucket.
Locality-Sensitive Hashing

Step 1: Add bands

Signature matrix $M$

$b$ bands

$r$ rows per band

One signature

(Leskovec at al., 2014; http://www.mmds.org/)
Locality-Sensitive Hashing

Can be tuned to catch most true-positives with least false-positives.

Step 1: Add bands

Signature matrix \( M \)

(Leskovec at al., 2014; http://www.mmds.org/)
Locality-Sensitive Hashing

Step 1: Add bands
Step 2: Hash columns within bands

(Leskovec at al., 2014; http://www.mmds.org/)
Locality-Sensitive Hashing

Step 1: Add bands
Step 2: Hash columns within bands

Columns 6 and 7 are surely different.

Matrix $M$

$r$ rows

$b$ bands

(Leskovec at al., 2014; http://www.mmds.org/)
Locality-Sensitive Hashing

Step 1: Add bands
Step 2: Hash columns within bands

Columns 2 and 6 are probably identical (candidate pair)
Columns 6 and 7 are surely different.

Matrix $M$
r rows
b bands

(Leskovec et al., 2014; http://www.mmds.org/)
Locality-Sensitive Hashing

Criteria for being candidate pair:

- They end up in same bucket for at least 1 band.

Step 1: Add bands
Step 2: Hash columns within bands

(columns of Matrix $M$ are mapped to buckets and bands)

Columns 2 and 6 are probably identical (candidate pair)

Columns 6 and 7 are surely different.

(Leskovec at al., 2014; http://www.mmds.org/)
Locality-Sensitive Hashing

Columns 2 and 6 are probably identical (candidate pair)
Columns 6 and 7 are surely different.

Step 1: Add bands
Step 2: Hash columns within bands

Simplification:
There are enough buckets compared to rows per band that columns must be identical in order to hash to the same bucket.

Thus, we only need to check if identical within a band.

(Leskovec at al., 2014; http://www.mmds.org/)
Document Similarity Pipeline

Shingling → Minhashing → Locality-sensitive hashing
Realistic Example: Probabilities of agreement

- 100,000 documents
- 100 random permutations/hash functions/rows

=> if 4byte integers then 40Mb to hold signature matrix
=> still 100k choose 2 is a lot (~5billion)
Realistic Example: Probabilities of agreement

- 100,000 documents
- 100 random permutations/hash functions/rows
  => if 4byte integers then 40Mb to hold signature matrix
  => still 100k choose 2 is a lot (~5billion)
- 20 bands of 5 rows
- Want 80% Jaccard Similarity; for any row \( p(S_1 == S_2) = .8 \)
Realistic Example: Probabilities of agreement

- 100,000 documents
- 100 random permutations/hash functions/rows
  => if 4byte integers then 40Mb to hold signature matrix
  => still 100k choose 2 is a lot (~5billion)
- 20 bands of 5 rows
- Want 80% Jaccard Similarity; for any row $p(S_1 == S_2) = 0.8$

$P(S_1 == S_2 | b)$: probability S1 and S2 agree within a given band
Realistic Example: Probabilities of agreement

- 100,000 documents
- 100 random permutations/hash functions/rows
  => if 4byte integers then 40Mb to hold signature matrix
  => still 100k choose 2 is a lot (~5billion)
- 20 bands of 5 rows
- Want 80% Jaccard Similarity ; for any row \( p(S_1 == S_2) = .8 \)

\[
P(S_1 == S_2 | b): \text{probability } S_1 \text{ and } S_2 \text{ agree within a given band} = 0.8^5 = .328 \quad \Rightarrow \quad P(S_1 != S_2 | b) = 1-.328 = .672
\]

\[
P(S_1 != S_2): \text{probability } S_1 \text{ and } S_2 \text{ do not agree in any band}
\]
Realistic Example: Probabilities of agreement

- 100,000 documents
- 100 random permutations/hash functions/rows
  => if 4byte integers then 40Mb to hold signature matrix
  => still 100k choose 2 is a lot (~5 billion)
- 20 bands of 5 rows
- Want 80% Jaccard Similarity; for any row \( p(S_1 == S_2) = .8 \)

\[ P(S_1 == S_2 | b): \text {probability S1 and S2 agree within a given band} = 0.8^5 = .328 \]
\[ \Rightarrow P(S_1 != S_2 | b) = 1-.328 = .672 \]

\[ P(S_1 != S_2): \text {probability S1 and S2 do not agree in any band} = .672^{20} = .00035 \]
Realistic Example: Probabilities of agreement

- 100,000 documents
- 100 random permutations/hash functions/rows
  => if 4-byte integers then 40Mb to hold signature matrix
  => still 100k choose 2 is a lot (~5 billion)
- 20 bands of 5 rows
- Want 80% Jaccard Similarity; for any row \( p(S_1 == S_2) = .8 \)

\[
P(S_1 == S_2 | b): \text{probability } S_1 \text{ and } S_2 \text{ agree within a given band} \]
\[
= 0.8^5 = .328 \quad \Rightarrow \quad P(S_1 != S_2 | b) = 1-.328 = .672
\]

\[
P(S_1 != S_2): \text{probability } S_1 \text{ and } S_2 \text{ do not agree in any band} \]
\[
= .672^{20} = .00035
\]

What if wanting 40% Jaccard Similarity?
Distance Metrics

Pipeline gives us a way to find *near-neighbors* in *high-dimensional space* based on Jaccard Distance (1 - Jaccard Sim).
Distance Metrics

Pipeline gives us a way to find *near-neighbors* in *high-dimensional space* based on Jaccard Distance (1 - Jaccard Sim).

Typical properties of a distance metric, $d$:

$d(x, x) = 0$

$d(x, y) = d(y, x)$

$d(x, y) \leq d(x, z) + d(z, y)$

(http://rosalind.info/glossary/euclidean-distance/)
Distance Metrics

Pipeline gives us a way to find near-neighbors in high-dimensional space based on Jaccard Distance (1 - Jaccard Sim).

There are other metrics of similarity. e.g:

- Euclidean Distance
- Cosine Distance
- Edit Distance
- Hamming Distance
Distance Metrics

Pipeline gives us a way to find *near-neighbors* in *high-dimensional space* based on Jaccard Distance (1 - Jaccard Sim).

There are other metrics of similarity. e.g:

- Euclidean Distance
- Cosine Distance
- Edit Distance
- Hamming Distance

\[
\text{distance}(X, Y) = \sqrt{\sum_{i}^{n} (x_i - y_i)^2} \quad (\text{“L2 Norm”})
\]
Distance Metrics

Pipeline gives us a way to find *near-neighbors* in *high-dimensional space* based on Jaccard Distance \((1 - \text{Jaccard Sim})\).

There are other metrics of similarity. e.g:

- **Euclidean Distance**

- **Cosine Distance**

- **Edit Distance**

- **Hamming Distance**

\[
\text{distance}(X, Y) = \sqrt{\sum_{i} (x_i - y_i)^2} \quad \text{("L2 Norm")}
\]
Locality Sensitive Hashing - Theory

LSH Can be generalized to many distance metrics by converting output to a probability and providing a lower bound on probability of being similar.
Locality Sensitive Hashing - Theory

LSH Can be generalized to many distance metrics by converting output to a probability and providing a lower bound on probability of being similar.

E.g. for euclidean distance:

- Choose random lines (analogous to hash functions in minhashing)
- Project the two points onto each line; match if two points within an interval
Link Analysis
The Web, circa 1998
The Web, circa 1998

Match keywords, language (information retrieval)

Explore directory
The Web, circa 1998

- Easy to game with “term spam”
- Match keywords, language (information retrieval)
- Explore directory
- Time-consuming;
- Not open-ended
The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure and produce much text and hyperlinks.

The PageRank Citation Ranking:
Bringing Order to the Web

January 29, 1998

Abstract

The importance of a Web page is an inherently subjective matter, which depends on the readers' interests, knowledge, and attitudes. But there is still much that can be said objectively about its importance, which we refer to as PageRank. This paper introduces PageRank, an algorithm for...
PageRank

**Key Idea:** Consider the citations of the website.
PageRank

Key Idea: Consider the citations of the website.

Who links to it? and what are their citations?
PageRank

Key Idea: Consider the citations of the website.

Who links to it? and what are their citations?

Innovation 1: What pages would a “random Web surfer” end up at?

Innovation 2: Not just own terms but what terms are used by citations?
PageRank

View 1: Flow Model: in-links as votes

Innovation 1: What pages would a “random Web surfer” end up at?

Innovation 2: Not just own terms but what terms are used by citations?
PageRank

View 1: Flow Model:

- in-links (citations) as votes
- but, citations from important pages should count more.

=> Use recursion to figure out if each page is important.

Innovation 1: What pages would a “random Web surfer” end up at?

Innovation 2: Not just own terms but what terms are used by citations?
PageRank

View 1: Flow Model:

How to compute?

Each page \((j)\) has an importance (i.e. rank, \(r_j\))

\[
vote_j = \frac{r_i}{n_j}
\]

\((n_j \text{ is } |\text{out-links}|)\)

\[
r_j = \sum_{i \in \text{inLinks}(j)} vote_i
\]
PageRank

View 1: Flow Model:

How to compute?

Each page \((j)\) has an importance (i.e. rank, \(r_j\))

\[
vote_j = \frac{r_i}{n_j}
\]

\[
r_j = \sum_{i \in \text{inLinks}(j)} vote_i
\]

\((n_j \text{ is } |\text{out-links}|)\)
PageRank

View 1: Flow Model:

How to compute?

Each page \((j)\) has an importance (i.e. rank, \(r_j\))

\[
vote_j = \frac{r_j}{n_j}
\]

\[
r_j = \sum_{i \in \text{inLinks}(j)} vote_i
\]

\((n_j\) is \(|\text{out-links}|\))
PageRank

View 1: Flow Model:

A System of Equations:

Each page ($j$) has an importance (i.e. rank, $r_j$)

$vote_j = \frac{r_i}{n_j}$

$(n_j \text{ is } |\text{out-links}|)$

$r_j = \sum_{i \in \text{inLinks}(j)} vote_i$
PageRank

View 1: Flow Model:

A System of Equations:

\[
\begin{align*}
  r_A &= \frac{r_B}{2} + \frac{r_C}{3} \\
  r_B &= \frac{r_A}{3} + \frac{r_C}{3} + \frac{r_D}{2} \\
  r_C &= \frac{r_A}{3} + \frac{r_B}{2} + \frac{r_D}{2} \\
  r_D &= \frac{r_A}{3} + \frac{r_B}{2} \\
\end{align*}
\]

How to compute?

Each page \((j)\) has an importance (i.e. rank, \(r_j\))

\[
\text{vote}_j = \frac{r_j}{n_j}
\]

\(n_j\) is \(|\text{out-links}|\)

\[
 r_j = \sum_{i \in \text{inLinks}(j)} \text{vote}_i
\]
PageRank

View 1: Flow Model: Solve

1 = r_A + r_B + r_C + r_D

How to compute?

Each page \((j)\) has an importance (i.e. rank, \(r_j\))

\[
vote_j = \frac{r_j}{n_j} \quad (n_j \text{ is } |\text{out-links}|)
\]

\[
r_j = \sum_{i \in \text{inLinks}(j)} vote_i
\]
PageRank

1 = r_A + r_B + r_C + r_D

\[
\begin{align*}
    r_A &= \frac{r_B}{2} + \frac{r_C}{1} \\
    r_B &= \frac{r_A}{3} + \frac{r_D}{2} \\
    r_C &= \frac{r_A}{3} + \frac{r_D}{2} \\
    r_D &= \frac{r_A}{3} + \frac{r_B}{2}
\end{align*}
\]

Transition Matrix, M

<table>
<thead>
<tr>
<th>to \ from</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1/2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>C</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>D</td>
<td>1/3</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Innovation: What pages would a “random Web surfer” end up at?
To start: N=4 nodes, so $r = [\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4},]$

View 2: Matrix Formulation

$1 = r_A + r_B + r_C + r_D$

$r_A = \frac{r_B}{2} + \frac{r_C}{3}$

$r_B = \frac{r_A}{3} + \frac{1}{2} r_D$

$r_C = \frac{r_A}{2} + \frac{r_D}{3}$

$r_D = \frac{r_A}{3} + \frac{r_B}{2}$

<table>
<thead>
<tr>
<th></th>
<th>to A</th>
<th>to B</th>
<th>to C</th>
<th>to D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1/2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>C</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>D</td>
<td>1/3</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Transition Matrix, $M$
Innovation: What pages would a “random Web surfer” end up at?
To start: N=4 nodes, so \( r = \left[ \frac{\frac{1}{4}}{} , \frac{\frac{1}{4}}{} , \frac{\frac{1}{4}}{} , \frac{\frac{1}{4}}{} \right] \)
after 1st iteration: \( M \cdot r = \left[ \frac{\frac{3}{8}}{} , \frac{\frac{5}{24}}{} , \frac{\frac{5}{24}}{} , \frac{\frac{5}{24}}{} \right] \)
after 2nd iteration: \( M(M \cdot r) = M^2 \cdot r = \left[ \frac{\frac{15}{48}}{} , \frac{\frac{11}{48}}{} , \ldots \right] \)

View 2: Matrix Formulation

\[
1 = r_A + r_B + r_C + r_D
\]

\[
\begin{align*}
    r_A &= \frac{r_B}{2} + \frac{r_C}{3} \\
    r_B &= \frac{2r_A}{3} + \frac{r_D}{2} \\
    r_C &= \frac{3r_A}{2} + \frac{r_D}{3} \\
    r_D &= \frac{3r_A}{2} + \frac{r_B}{2}
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>to</th>
<th>from</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>1/2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1/3</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>C</td>
<td>1/3</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>D</td>
<td>1/3</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Transition Matrix, M
Innovation: What pages would a “random Web surfer” end up at?

To start: N=4 nodes, so \( r = \left[ \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right] \)

after 1st iteration: \( M \cdot r = \left[ \frac{3}{8}, \frac{5}{24}, \frac{5}{24}, \frac{5}{24} \right] \)

after 2nd iteration: \( M(M \cdot r) = M^2 \cdot r = \left[ \frac{15}{48}, \frac{11}{48}, \ldots \right] \)

**Power iteration algorithm**

initialize: \( r[0] = [1/N, \ldots, 1/N], \) \( r[-1]=[0,\ldots,0] \)

while (err_norm(\( r[t], r[t-1]) > \text{min\_err}):

\[
\text{err\_norm}(v1, v2) = \|v1 - v2\| \ #L1\ norm
\]

“Transition Matrix”, \( M \)
Innovation: What pages would a “random Web surfer” end up at?

To start: N=4 nodes, so \( r = \left[ \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right] \)

after 1st iteration: \( M \cdot r = \left[ \frac{3}{8}, \frac{5}{24}, \frac{5}{24}, \frac{5}{24} \right] \)

after 2nd iteration: \( M(M \cdot r) = M^2 \cdot r = \left[ \frac{15}{48}, \frac{11}{48}, \ldots \right] \)

Power iteration algorithm

initialize: \( r[0] = [1/N, ..., 1/N], \)
\( r[-1]=[0,...,0] \)

while (err_norm\((r[t],r[t-1]\))>min_err):

\( r[t+1] = M \cdot r[t] \)

\( t+=1 \)

solution = \( r[t] \)

err_norm(v1, v2) = |v1 - v2|  #L1 norm

<table>
<thead>
<tr>
<th>to \ from</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1/2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>C</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td>D</td>
<td>1/3</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

“Transition Matrix”, \( M \)
As $err\_\text{norm}$ gets smaller we are moving toward: $r = M \cdot r$

View 3: Eigenvectors:

**Power iteration algorithm**

initialize: $r[0] = [1/N, ..., 1/N]$
$r[-1]=[0,...,0]$

while $(err\_\text{norm}(r[t],r[t-1])>\text{min}\_\text{err})$:
    $r[t+1] = M \cdot r[t]$
    $t+=1$

solution = $r[t]$

$err\_\text{norm}(v1, v2) = |v1 - v2| \ #L1 \text{ norm}$
Power iteration algorithm

initialize: $r[0] = [1/N, ..., 1/N]$  
$r[-1]=[0,...,0]$

while (err_norm($r[t], r[t-1]$)>min_err):
    $r[t+1] = M \cdot r[t]$
    $t+=1$

solution = $r[t]$

err_norm($v1, v2$) = $|v1 - v2|$ #L1 norm

As err_norm gets smaller we are moving toward: $r = M \cdot r$

View 3: Eigenvectors:
We are actually just finding the **eigenvector** of M.

$x$ is an **eigenvector** of $\lambda$ if: $A \cdot x = \lambda \cdot x$
As err_norm gets smaller we are moving toward: \( r = M \cdot r \)

**View 3: Eigenvectors:**
We are actually just finding the *eigenvector* of M.

---

**Power iteration algorithm**

initialize: \( r[0] = [1/N, \ldots, 1/N] \)
\( r[-1] = [0, \ldots, 0] \)

while (err_norm\((r[t], r[t-1]) > \text{min\_err})

\( r[t+1] = M \cdot r[t] \)
\( t += 1 \)
solution = \( r[t] \)

err_norm\((v1, v2) = |v1 - v2| \ #L1\ norm

---

\( x \) is an *eigenvector* of \( \lambda \) if:
\( A \cdot x = \lambda \cdot x \)

\( A = 1 \)
since columns of M sum to 1.
thus, \( 1r = Mr \)
Where is surfer at time t+1? \( p(t+1) = M \cdot p(t) \)

Suppose: \( p(t+1) = p(t) \), then \( p(t) \) is a stationary distribution of a random walk. Thus, \( r \) is a stationary distribution. Probability of being at given node.
Where is surfer at time $t+1$? $p(t+1) = M \cdot p(t)$

Suppose: $p(t+1) = p(t)$, then $p(t)$ is a stationary distribution of a random walk.

Thus, $r$ is a stationary distribution. Probability of being at given node.

**aka 1st order Markov Process**

- Rich probabilistic theory. One finding:
  - Stationary distributions have a unique distribution if:
    - No “dead-ends”: a node can’t propagate its rank
    - No “spider traps”: set of nodes with no way out.

Also known as being stochastic, irreducible, and aperiodic.
View 4: Markov Process - Problems for vanilla PI

aka 1st order Markov Process
- Rich probabilistic theory. One finding:
  - Stationary distributions have a unique distribution if:
    - No "dead-ends": a node can't propagate its rank
    - No "spider traps": set of nodes with no way out.
- Also known as being stochastic, irreducible, and aperiodic.

What would \( r \) converge to?

<table>
<thead>
<tr>
<th>to \ from</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>1/3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
aka 1st order Markov Process

- Rich probabilistic theory. One finding:
  - Stationary distributions have a unique distribution if:
    - No “dead-ends”: a node can’t propagate its rank
    - No “spider traps”: set of nodes with no way out.

Also known as being stochastic, irreducible, and aperiodic.
aka 1st order Markov Process

- Rich probabilistic theory. One finding:
  - Stationary distributions have a unique distribution if:
    - Same node doesn’t repeat at regular intervals
    - Columns sum to 1
    - Non-zero chance of going to any other node

Also known as being stochastic, irreducible, and aperiodic.
Goals:
No “dead-ends”
No “spider traps”

The “Google” PageRank Formulation
Add teleportation: At each step, two choices
1. Follow a random link (probability, $\beta = \sim .85$)
2. Teleport to a random node (probability, $1-\beta$)
Goals:
No “dead-ends”
No “spider traps”

The “Google” PageRank Formulation
Add teleportation: At each step, two choices
1. Follow a random link (probability, $\beta = \sim .85$)
2. Teleport to a random node (probability, $1-\beta$)

<table>
<thead>
<tr>
<th>to</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>$\frac{1}{3}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>$\frac{1}{3}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>$\frac{1}{3}$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The “Google” PageRank Formulation

Add teleportation: At each step, two choices

1. Follow a random link (probability, $\beta = \sim .85$)
2. Teleport to a random node (probability, $1-\beta$)

Goals:
No “dead-ends”
No “spider traps”
**Goals:**
- No “dead-ends”
- No “spider traps”

**The “Google” PageRank Formulation**
Add teleportation: At each step, two choices

1. Follow a random link (probability, $\beta = \sim .85$)
2. Teleport to a random node (probability, $1-\beta$)

<table>
<thead>
<tr>
<th>to \ from</th>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$0+.15*\frac{1}{4}$</td>
<td>$0+.15*\frac{1}{4}$</td>
<td>$85<em>1+.15</em>\frac{1}{4}$</td>
<td>$0+.15*\frac{1}{4}$</td>
</tr>
<tr>
<td>$B$</td>
<td>$.85*\frac{1}{3}+.15*\frac{1}{4}$</td>
<td>$0+.15*\frac{1}{4}$</td>
<td>$0+.15*\frac{1}{4}$</td>
<td>$\text{.85<em>1+.15</em>\frac{1}{4}}$</td>
</tr>
<tr>
<td>$C$</td>
<td>$.85*\frac{1}{3}+.15*\frac{1}{4}$</td>
<td>$0+.15*\frac{1}{4}$</td>
<td>$0+.15*\frac{1}{4}$</td>
<td>$0+.15*\frac{1}{4}$</td>
</tr>
<tr>
<td>$D$</td>
<td>$.85*\frac{1}{3}+.15*\frac{1}{4}$</td>
<td>$\text{.85<em>1+.15</em>\frac{1}{4}}$</td>
<td>$0+.15*\frac{1}{4}$</td>
<td>$0+.15*\frac{1}{4}$</td>
</tr>
</tbody>
</table>
**Goals:**
No “dead-ends”
No “spider traps”

**The “Google” PageRank Formulation**
Add teleportation: At each step, two choices
1. Follow a random link (probability, $\beta = \sim .85$)
2. Teleport to a random node (probability, $1-\beta$)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>$\frac{1}{3}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>$\frac{1}{3}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>$\frac{1}{3}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Goals:
No “dead-ends”
No “spider traps”

The “Google” PageRank Formulation
Add teleportation: At each step, two choices
1. Follow a random link (probability, \( \beta = \sim 0.85 \))
2. Teleport to a random node (probability, \( 1 - \beta \))

\[
\begin{array}{cccc}
\text{to} & \text{from} & A & B & C & D \\
A & 0 & \frac{1}{4} & 1 & 0 \\
B & \frac{1}{3} & \frac{1}{4} & 0 & 1 \\
C & \frac{1}{3} & \frac{1}{4} & 0 & 0 \\
D & \frac{1}{3} & \frac{1}{4} & 0 & 0 \\
\end{array}
\]
Goals:
No “dead-ends”
No “spider traps”

The “Google” PageRank Formulation
Add teleportation: At each step, two choices
1. Follow a random link (probability, $\beta = ~.85$)
2. Teleport to a random node (probability, $1-\beta$)

<table>
<thead>
<tr>
<th>to \ from</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0.85\times\frac{1}{4}+0.15\times\frac{1}{4}</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>\frac{1}{3}</td>
<td>0.85\times\frac{1}{4}+0.15\times\frac{1}{4}</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>\frac{1}{3}</td>
<td>0.85\times\frac{1}{4}+0.15\times\frac{1}{4}</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>\frac{1}{3}</td>
<td>0.85\times\frac{1}{4}+0.15\times\frac{1}{4}</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The “Google” PageRank Formulation
Add teleportation: At each step, two choices
1. Follow a random link (probability, $\beta = \sim .85$)
2. Teleport to a random node (probability, $1-\beta$)
   (Teleport from a dead-end has probability 1)

<table>
<thead>
<tr>
<th>to \ from</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0+.15(\frac{1}{4})</td>
<td>1(\frac{1}{4})</td>
<td>85*1+.15(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
</tr>
<tr>
<td>B</td>
<td>.85(\frac{1}{3})+.15(\frac{1}{4})</td>
<td>1(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
<td>.85*1+.15(\frac{1}{4})</td>
</tr>
<tr>
<td>C</td>
<td>.85(\frac{1}{3})+.15(\frac{1}{4})</td>
<td>1(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
</tr>
<tr>
<td>D</td>
<td>.85(\frac{1}{3})+.15(\frac{1}{4})</td>
<td>1(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
</tr>
</tbody>
</table>
Goals:
No “dead-ends”
No “spider traps”

Teleportation, as Flow Model:

\[
    r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta) \frac{1}{N}
\]

(Brin and Page, 1998)

<table>
<thead>
<tr>
<th>to \ from</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0+.15(\frac{1}{4})</td>
<td>1(\frac{1}{4})</td>
<td>85(\frac{1}{3})+.15(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
</tr>
<tr>
<td>B</td>
<td>.85(\frac{1}{3})+.15(\frac{1}{4})</td>
<td>1(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
<td>.85(\frac{1}{3})+.15(\frac{1}{4})</td>
</tr>
<tr>
<td>C</td>
<td>.85(\frac{1}{3})+.15(\frac{1}{4})</td>
<td>1(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
</tr>
<tr>
<td>D</td>
<td>.85(\frac{1}{3})+.15(\frac{1}{4})</td>
<td>1(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
<td>0+.15(\frac{1}{4})</td>
</tr>
</tbody>
</table>
Goals:
No “dead-ends”
No “spider traps”

Teleportation, as Flow Model:
\[ r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta) \frac{1}{N} \]
(Brin and Page, 1998)

Teleportation, as Matrix Model:
\[ M' = \beta M + (1 - \beta) \left[ \frac{1}{N} \right]_{N \times N} \]
Goals:
No “dead-ends”
No “spider traps”

Teleportation, as Flow Model:
\[ r_j = \sum_{i \rightarrow j} \beta \frac{r_i}{d_i} + (1 - \beta) \frac{1}{N} \]
(Brin and Page, 1998)

Teleportation, as Matrix Model:
\[ M' = \beta M + (1 - \beta) \left[ \frac{1}{N} \right]_{N \times N} \]

<table>
<thead>
<tr>
<th>to \ from</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0+.15*(\frac{1}{4})</td>
<td>.85*(\frac{1}{4})+.15*(\frac{1}{4})</td>
<td>85*1+.15*(\frac{1}{4})</td>
<td>0+.15*(\frac{1}{4})</td>
</tr>
<tr>
<td>B</td>
<td>.85*(\frac{1}{3})+.15*(\frac{1}{4})</td>
<td>.85*(\frac{1}{4})+.15*(\frac{1}{4})</td>
<td>0+.15*(\frac{1}{4})</td>
<td>.85*1+.15*(\frac{1}{4})</td>
</tr>
<tr>
<td>C</td>
<td>.85*(\frac{1}{3})+.15*(\frac{1}{4})</td>
<td>.85*(\frac{1}{4})+.15*(\frac{1}{4})</td>
<td>0+.15*(\frac{1}{4})</td>
<td>0+.15*(\frac{1}{4})</td>
</tr>
<tr>
<td>D</td>
<td>.85*(\frac{1}{3})+.15*(\frac{1}{4})</td>
<td>.85*(\frac{1}{4})+.15*(\frac{1}{4})</td>
<td>0+.15*(\frac{1}{4})</td>
<td>0+.15*(\frac{1}{4})</td>
</tr>
</tbody>
</table>
Teleportation, as Flow Model:

\[ r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta) \frac{1}{N} \]

(Brin and Page, 1998)

Teleportation, as Matrix Model:

\[ M' = \beta M + (1 - \beta) \left[ \frac{1}{N} \right]_{N \times N} \]

---

**Goals:**
- No “dead-ends”
- No “spider traps”

To apply: run power iterations over \( M' \) instead of \( M \).
**Goals:**
No “dead-ends”
No “spider traps”

**Teleportation, as Flow Model:**

\[ r_j = \sum_{i \rightarrow j} \beta \frac{r_i}{d_i} + (1 - \beta) \frac{1}{N} \]

(Brin and Page, 1998)

**Teleportation, as Matrix Model:**

\[ M' = \beta M + (1 - \beta) \left[ \frac{1}{N} \right]_{N \times N} \]

**Steps:**
1. Compute M
2. Add 1/N to all dead-ends.
3. Convert M to M’
4. Run Power Iterations.

<table>
<thead>
<tr>
<th>to \ from</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0+.15*¼</td>
<td>1*¼</td>
<td>85<em>1+.15</em>¼</td>
<td>0+.15*¼</td>
</tr>
<tr>
<td>B</td>
<td>.85<em>¼+.15</em>¼</td>
<td>1*¼</td>
<td>0+.15*¼</td>
<td>.85<em>1+.15</em>¼</td>
</tr>
<tr>
<td>C</td>
<td>.85<em>¼+.15</em>¼</td>
<td>1*¼</td>
<td>0+.15*¼</td>
<td>0+.15*¼</td>
</tr>
<tr>
<td>D</td>
<td>.85<em>¼+.15</em>¼</td>
<td>1*¼</td>
<td>0+.15*¼</td>
<td>0+.15*¼</td>
</tr>
</tbody>
</table>