L23: PageRank

Jeff M. Phillips

April 13, 2020
Final Report

At most 4 pages/student. Don’t cram in too much!

- **Succinct title (and names)**
- Problem definition and motivation.
- Explain your Data.

- **key idea**
- What did you do (which techniques, an implementation, a comparison, an extension)
- What did you learn? Artifacts (charts, plots, examples, math) and Intuition (in words, did it work?)
Web page similarity (Search)

Inverted Index

Search

- Define most relevant webpages

Query: "apple"

Set \{apple\}

Search & format

\text{cosine}, \text{tf-idf}, \text{bag-of-words}, \text{word vector}

Help index (copy to ranked web pages)
Crawlers: program that walks around web:
1) read page
2) follow random hyperlinks

Spammers: build flood pages w/ hyperlink to your page w/ hyperlink tag.
0. Indexes: Alternative to Search Engine

Yahoo! and LookSmart

Built an organized, curated collection of websites
Page Rank

\[
S(p_i, \text{turn}) = f(\text{text}(p_i), \text{links } p_i, \mathbf{g}(i))
\]

• Pages are important if linked to by other important web pages.
• Page is important if a "random surfer" were to find it.

Web is a big graph \( G = (V, E) \)
- \( V = \{ s, t \} \) & all pages \( 3 \)
- \( E = \{ E_{is} \} = \text{link } p_i \rightarrow p_s \) \( 3 \)

Define \( MC \rightarrow \mathbf{g} \in \text{converged vector distribution} \)
- \( \mathbf{g}(i) \) says how important page \( i \) is.
Compute $q^*$ of Webgraph

- Keep track of crawlers: how frequent return.
- Buy big computer: Compute $\text{eig}(P)$
- Precompute $P^* = P \cdot P \cdot P \cdots P$

- $q^* = q_0$ last night
  - \( \text{for } j = 1 \text{ to } 50 \)
    - \( q_j = P \cdot q_{j-1} \)

Power method
Anatomy of Web

is this $G$ ergodic?

ANATOMY of WEB

Strongly Connected Component

Tubes

IN

OUT

no IN links

IN

OUT

tendrils

disconnected

tendrils IN

OUT

$\div$
Anatomy of Web
Can we make $G$ ergodic?

- Teleportation/taxation
  - about once every $T$ steps
  - jump to random node.

$P \text{ prob trans } (G)$

$R = (1 - \beta) P + \beta \frac{Q}{\lambda}$

$\lambda \to \text{ dense}$

$\beta = 0.15$

$R_{Gi} = ((1 - \beta) P + \beta G) \delta_{i1} - (1 - \beta) P \delta_{i1} + \beta \frac{1}{n}$

$n \times 1 \text{ vector}$
Spam Farms

90% of web

Google counter

A Google search bar this structure

Your web page

Spam Farm
Trust Rank (2015?)

Only teleport to trusted pages.

\[ r_t \leftarrow q_x, \text{ pagerank} \]

\[ t \in q_x \text{ pages} \cap k \]

\[ t \in q_x \text{ trusted teleport} \]

\[ \frac{r(j) - t(j)}{r(j)} \text{ if large } \rightarrow \text{ spam} \]

\[ \text{truthfulness of webpage} \]
Word Count

Consider as input all of English Wikipedia stored in DFS. Goal is to count how many times each word is used.
Inverted Index

Consider as input all of English Wikipedia stored in DFS. Goal is to build an index, so each word has a list of pages it is in.
Phrases

Consider as input all of English Wikipedia stored in DFS. Goal is to build an index, on 3-grams (sequence of 3 words) that appears on exactly one page, with link to page.
Label Propagation (Graph)

Consider a large graph $G = (V, E)$ (e.g., a social network), with a subset of notes $V' \subseteq V$ with labels (e.g., \{pos, neg\}). Each node stores its label (if any) and edges. Assign a vertex a label if (a) unlabeled, (b) has $\geq 5$ labeled neighbors, (c) based on majority vote.
Label Propagation (Embedding)

Consider a data set $X \subset \mathbb{R}^d$, with a subset of points $X' \subset X$ with labels (e.g., \{pos, neg\}). Implicitly defines graph with $V = X$ and $E$ using $k = 20$ nearest neighbors.

Assign a vertex a label if (a) unlabeled, (b) has $\geq 5$ labeled neighbors, (c) based on majority vote.
Example PageRank

\[
M = \begin{bmatrix}
0 & 1/2 & 0 & 0 \\
1/3 & 0 & 1 & 1/2 \\
1/3 & 0 & 0 & 1/2 \\
1/3 & 1/2 & 0 & 0 \\
\end{bmatrix}
\]
Example PageRank

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

Stripes:

\[ M_1 = \begin{bmatrix} 0 \\ 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \quad M_2 = \begin{bmatrix} 1/2 \\ 0 \\ 0 \\ 1/2 \end{bmatrix} \quad M_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad M_4 = \begin{bmatrix} 0 \\ 1/2 \\ 1/2 \\ 0 \end{bmatrix} \]

These are stored as \((1 : (1/3, 2), (1/3, 3), (1/3, 4)), (2 : (1/2, 1)(1/2, 4)), (3 : (1, 3)), \text{ and } (4 : (1/3, 1), (1/2, 2))\).
Example PageRank

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

Blocks:

\[M_{1,1} = \begin{bmatrix}
0 & 1/2 \\
1/3 & 0 \\
1/3 & 1/2
\end{bmatrix} \quad M_{1,2} = \begin{bmatrix}
0 & 0 \\
1 & 1/2 \\
1/3 & 1/2
\end{bmatrix} \quad M_{2,1} = \begin{bmatrix}
1/3 & 0 \\
1/3 & 1/2 \\
1/3 & 1/2
\end{bmatrix} \quad M_{2,2} = \begin{bmatrix}
0 & 1/2 \\
0 & 0 \\
0 & 0
\end{bmatrix}
\]

These are stored as (1 : (1/2, 2)), (2 : (1/3, 1)), as (2 : (1, 3), (1/2, 4)), as (3 : (1/3, 1)), (4 : (1/3, 1), (1/2, 2)), and as (3 : (1/2, 4)).