PageRank & Random walks

Danai Koutra
University of Michigan, Ann Arbor
Admin stuff

• The 1\textsuperscript{st} assignment is out (Canvas)
• Due October 2\textsuperscript{nd}
• Any questions?

• More questions? Post them on Canvas.
Projects

• Find your team today!
• Need project ideas?

• If you have a team already
  ✦ Identify a plan
  ✦ Talk about the details:
    ▪ Do you have a specific problem?
    ▪ Do you have data? Do you need to crawl it?
    ▪ How are you going to evaluate your approach?
    ▪ Do you need high-end servers for your implementations?
Additional datasets

1 TB uncompressed

I have every publicly available Reddit comment for research. ~1.7 billion comments @ 250 GB compressed. Any interest in this? (self.datasets) submitted 2 months ago by Stuck_In_the_Matrix

I am currently doing a massive analysis of Reddit's entire publicly available comment dataset. The dataset is ~1.7 billion JSON objects complete with the comment, score, author, subreddit, position in comment tree and other fields that are available through Reddit's API.

I'm currently doing NLP analysis and also putting the entire dataset into a large searchable database using Sphinxsearch (also testing ElasticSearch).

This dataset is over 1 terabyte uncompressed, so this would be best for large research projects. If you're interested in a sample month of comments, that can be arranged as well. I am trying to find a place to host this large dataset -- I'm reaching out to Amazon since they have open data initiatives.

**EDIT:** I'm putting up a Digital Ocean box with 2 TB of bandwidth and will throw an entire month's worth of comments up (~5 gigs compressed) It's now a torrent. This will give you guys an opportunity to examine the data. The file is structured with JSON blocks delimited by new lines (~\
.

One month of comments is now available here:

**Download Link:** Torrent

**Direct Magnet File:** magnet:?xt=urn:btih:32916ad30ce4c90ee4c47a95bd0075e44ac15dd2&dn=RC%5F2015-01.bz2&tr=udp%3A%2F%2Ftracker.openbittorrent.com%3A80&tr=udp%3A%2F%2Fopen.demoni.com%3A1337&tr=udp%3A%2F%2Ftracker.coppersurfer.tk%3A6969&tr=udp%3A%2F%2Ftracker.leechers-paradise.org%3A6969

**Tracker:** udp://tracker.openbittorrent.com:80

**Total Comments:** 53,851,542

**Compression Type:** bzip2 (5,452,413,560 bytes compressed | 31,648,374,106 bytes uncompressed)

**md5:** a3fc3d9db18786e4486381a7f37d08e2 RC_2015-01.bz2

The dataset is ~1.7 billion JSON objects complete with the comment, score, author, subreddit, position in comment tree, and more.
Additional datasets

The arXiv as Dataset

12th July 2015

The arXiv is a repository of over 1 million preprints in physics, mathematics and computer science. It is truly open access, and the preprints are an excellent dataset for testing out all sorts of language modelling / machine learning prototypes.

What’s available?

Full-text and article metadata are published in two different ways.

- The preprint metadata (title, abstract, authors, categories) are published via the OAI protocol for metadata harvesting (OAI-PMH) and via the arXiv API
- The full-text of all preprints is made available in a huge data dump on S3, either as PDFs or as TeX source files.
Last time

• Patterns and laws for time-evolving graphs
  ✷ Diameter
  ✷ Connected components
  ✷ Structures
  ✷ Popularity in blogs

• Generating realistic graphs
  ✷ Kronecker graphs
Today

• **PageRank [Yuru]**

• **Fast Random Walks [Haad]**

• **HITS [Danai] – on Thursday**
Today - Feedback

- Grade yourself, and your peers grade
- If (your grade $\approx$ peer’s average grade)
  - you get your grade
- Otherwise
  - you get the peer grade

- Peers:
  - Correlation between the grade you give and the mean grades for all the presentation (up to 10% credit)
The PageRank Citation Ranking: Bringing Order to the Web

Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd

presented by
Yuru Shao, Haad Khan
09/22/2015
Problem Definition

• How to provide high quality search results to users?
  • To give them the pages they want most

• Search engines need an *objective, mechanical, and effective* method for measuring the relative importance of web pages
Challenges

• The World Wide Web is large and heterogeneous
  • The Indexed Web contains at least 4.77 billion pages (Monday, 21 September, 2015) [1]
• Web pages are diverse
  • significantly different from academic publications
• Search engines have to contend with inexperienced users deal with pages that can affect ranking functions

PageRank

• Based on link structure of the web
• The Web is a graph
  • pages = nodes
  • links = edges
  • forward links = out-edges
  • backlinks = in-edges
• A and B are backlinks of C
  • C is a forward link of A & B
PageRank - Basic Idea

• Taking advantage of the link structure of the Web to produce a global importance ranking of every single page
• A page has high rank if the sum of the ranks of its backlinks is high
  • A page with high rank may have many backlinks
  • Or has a few highly ranked backlinks
Simplified PageRank

- $u$ is a web page
- $F_u = \text{set of pages } u \text{ points to (forward links)}$
- $N_u = |F_u| \text{ (# of forward links)}$
- $B_u = \text{set of pages pointing to } u \text{ (backlinks)}$
- $c$ is normalization factor

The rank of a page is divided evenly to contribute to the ranks of its forward links.

The rank of a page is calculated from all its backlinks.
Suppose \( c = 1 \)

\[
\begin{align*}
B_u &= \{a, b\} \\
\mathbf{R}(u) &= c \sum_{v \in B_u} \frac{\mathbf{R}(v)}{N_v} = \frac{\mathbf{R}(a)}{N_a} + \frac{\mathbf{R}(b)}{N_b} = \frac{53}{2} + \frac{50}{2} = 51.5
\end{align*}
\]
Rank Sink

- The loop will accumulate rank but never distribute any rank
  - There are no out-edges
PageRank

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

where \[ ||R'|| = 1 \]

E(u), some vector over the web pages that corresponds to a source of rank

- Based on random surfer model
  - \( E \) - probability that a user visits a page
  - \( E \) vector is uniform over all web pages with \( ||E||_1 = 0.15 \)
PageRank

- $A$, a square matrix with the rows and cols corresponding to web pages
- $A_{u,v} = 1$ if there is an edge from $u$ to $v$, otherwise $A_{u,v} = 0$

\[
R' = c(AR' + E) \\
= c(A + E \times 1)R'
\]
Computing PageRank

\[
R_0 \leftarrow S \\
\text{loop:} \\
R_{i+1} \leftarrow AR_i \\
d \leftarrow \|R_i\|_1 - \|R_{i+1}\|_1 \\
R_{i+1} \leftarrow R_{i+1} + dE \\
\delta \leftarrow \|R_{i+1} - R_i\|_1 \\
\text{while } \delta > \epsilon
\]

PageRank computation converges quickly
Strengths

• The PageRank algorithm is neat and versatile
• Implementation and experiments
  • PageRank is effective
• Discussed other applications
  • Web traffic estimation
  • Backlink prediction
  • User navigation
Weaknesses

• Objective importance ≠ Subjective importance
  • How to optimize it for each individual user
• Could be abused?
  • An attacker creates a large number of pages that point to a malicious page
  • The malicious page could have high rank
Conclusion

- This paper presents PageRank algorithm for measuring relative importance of web pages
- Better query results and good scalability
- PageRank has a wide range of applications
Personalized Random Walk with Restarts (RWR)

\[ (1 - c) y = [I - cAD^{-1}] x \]
Student Presentation 2
Fast Random Walks with Restart and Its applications

Authors: Hanghang Tong, Christos Faloutsos and Jia-Yu Pan

Presented by:
Haad Khan, Yuru Shao
Problem Definition

Imagine someone gives you a HUGE network. How would you estimate structure, size etc? One solution is random walk.
Random Walk

Given a graph, a starting vertex go to a random neighbour. We repeat the process of going to random neighbour.
Random Walk with Restart

A). RWR iteratively estimates the proximity (affinity score) between two nodes
B). Starting at a node, the walker faces two choices at each step: either moving to a randomly chosen neighbor, or jumping back to the starting node.
C). Probability to return is ‘c’ called 'the restarting probability'
Random walk with restart

Nearby nodes, higher scores
More red, more relevant

Ranking vector
r4
Applications

Applications
1. Content-based Image Retrieval
2. Personalized PageRank
3. Anomaly Detection
4. Link Prediction
5. Semi-supervised Learning
Motivation: Ranking

Google
[Brin+ ‘98][Tong+ ’06][Ji+ ‘11]...
Motivation: Fraud Detection

Lax controls?

fraudsters?

[Neville+ ‘05][Chau+ ‘07][McGlohon+ ’09]...
RWR Algorithm

\[ \vec{r}_i = c \vec{W} \vec{r}_i + (1 - c) \vec{e}_i \]

\[ \vec{r}_i = (1 - c)(I - c \vec{W})^{-1} \vec{e}_i \]  \hspace{1cm} (i)

\[ = (1 - c)Q^{-1} \vec{e}_i \]

R(i,j) : Steady State Probability
W : Weighted Graph
\( c \) : probability
\( e \) : starting vector

Problems/Challenges:
Iterating equation to solve linear system is slow \( O(mE) \) with \( E \) edges and \( m \) number of iterations
Mentioned as “on the fly”
Solution: Balance between online and offline
Basic Idea

Find Community

Fix the remaining

Combine

TONG ICDM 2006
Fast RWR Algorithm (Basic Idea)

Also referred to as B LIN
The Q-inverse in the eq (1) can be precomputed. Paper refers to it as Precompute.
Pre-computational stage

Q: Efficiently compute $Q^{-1}$
A: A few small, instead of ONE BIG, matrices inversions
Fast RWR Algorithm (Basic Idea)

Partition graph into $k$ partitions
Decompose $W$ into $W_1 + W_2$. $W_1$ contains all in partition links and $W_2$ contains all cross partition links
Compute and store $Q_1$
Do low rank approximation for $W_2 = USV$
Compute and Store

Online Computation

$$\tilde{\Lambda} = (S^{-1} - cVQ_1^{-1}U)^{-1}.$$ 

$$r_i^* = (1 - c)(Q_1^{-1}e_i + cQ_1^{-1}U\tilde{\Lambda}VQ_1^{-1}e_i).$$
Sherman-Morrison Lemma [23]: if $X^{-1}$ exists, then:

$$(X - USV)^{-1} = X^{-1} + X^{-1} U \tilde{\Lambda} VX^{-1}$$

where $\tilde{\Lambda} = (S^{-1} - VX^{-1} U)^{-1}$
On-Line Query Stage

Q: Efficiently recover one column of Q
A: A few, instead of MANY, matrix-vector multiplication
Computation

Pre-computational Costs:

(1) partitioning the whole graph;
(2) inversion of each $I - c\bar{W}_{1,i}, (i = 1, ..., k)$;
(3) low-rank approximation on $\bar{W}_2$;
(4) inversion of $(S^{-1} - VQ_1^{-1}U)$. 
Strengths

1. Design of algorithm which significantly reduces computation and storage costs.
2. Proof of error bound
3. Real world tests
Weaknesses

1. On the fly method can also balance with online and offline methods. Paper does not mention this approach.
2. Establishes result of the onthefly method as ground truth and compares the algorithm results with the best performing algorithm.
3. It sacrifices precision for speed.
Questions

Random walk with restart
Applications
NB_LIN algorithm
Strengths
Weaknesses
Perseus: vision & demo
Perseus: vision & demo
On Thursday

Belief Propagation
What to read before class

- **NetProbe** [Abhilash]

- **FaBP (Fast Belief Propagation)**
  - Wolfgang Gatterbauer, Stephan Guennemann, Danai Koutra, Christos Faloutsos. **Linearized and Single-Pass Belief Propagation.** Proceedings of the VLDB Endowment, Volume 8(4) (VLDB'15), August 2015. (Chapter 5 of "Exploring and Making Sense of Large Graphs")