Challenges of Web Search

- Distributed data
  - data is stored on millions of machines with varying network characteristics
- Volatile data
  - new computers and data can be added and removed easily
  - dangling links and relocation problems
- Large volume
- Unstructured and redundant data
  - not all HTML pages are well structured
  - much of the Web is repeated (mirrored or copied)
Challenges of Web Search

- Quality of data
  - data can be false, invalid (e.g. out of date), SPAM
  - poorly written, can contain grammatical errors

- Heterogeneous data
  - multiple media types, multiple formats, different languages

- Unsophisticated users
  - information need may be unclear
  - may have difficulty formulating a useful query

Web Challenges – Size of Vocabulary

- Heap’s law: $V = Kn^\beta$
  - $\beta$ is typically between 0.4 and 0.6, so vocabulary size $V$ grows roughly with the square root of the text size $n$

- 99% of distinct words in the VLC2 collection are not dictionary head-words (Hawking, Very Large Scale Information Retrieval)
• A characteristic of the Web is its hyperlink structure
• Web search engines exploit properties of the structure to try and overcome some of the web-specific challenges
• Basic idea: hyperlink structure can be used to infer the validity / popularity / importance of a page
  – similar to citation analysis in academic publishing
  – number of links to a page correspond with page’s importance
  – links coming from an important page are indicators of other important pages
  – Anchor text describes the page
  ∗ can be a useful source of text in addition to the text on the page itself, eg Big Blue → IBM

PageRank

• PageRank is query-independent and provides a global importance score for every page on the web
  – can be calculated once for all queries
  – but can’t be tuned for any one particular query
• PageRank has a simple intuitive interpretation:
  – PageRank score for a page is the probability a random surfer would visit that page
• PageRank is/was used by Google
  – PageRank is combined with other measures such as TF×IDF
• Pages with many backlinks are typically more important than pages with few backlinks
• But pages with few backlinks can also be important
  – some links, e.g. from Yahoo, are more important than other links

PageRank Scoring

• Consider a browser doing a random walk on the Web
  – start at a random page
  – at each step go to another page along one of the out-links, each link having equal probability
• Each page has a long-term visit rate (the “steady state”)
  – use the visit rate as the score
Simplified PageRank

\[ R(u) = d \sum_{v : v \rightarrow u} \frac{R(v)}{N_v} \]

- \( u \) is a web page
- \( N_v \) is the number of links from \( v \)

Teleporting

- Web is full of dead-ends
  - “long-term visit rate” doesn’t make sense
- A page may have no in-links
- Teleporting: jump to any page on the Web at random (with equal probability \( 1/N \))
  - when there are no out-links use teleporting
  - otherwise use teleporting with probability \( \alpha \), or follow a link chosen at random with probability \( 1 - \alpha \)
\[ R(u) = (1 - \alpha) \sum_{v: v \rightarrow u} \frac{R(v)}{N_v} + \alpha E(u) \]

- \( E(u) \) is a prior distribution over web pages
- Typical value of \( \alpha \) is 0.1
- \( R(u) \) can be calculated using an iterative algorithm

**Probabilistic Interpretation of PageRank**

- PageRank models the behaviour of a "random surfer"
- Surfer randomly clicks on links, sometimes jumping to any page at random based on \( E \)
- Probability of a random jump is \( \alpha \)
- PageRank for a page is the probability that the random surfer finds himself on that page
A Markov chain consists of \( n \) states plus an \( n \times n \) transition probability matrix \( P \).

At each step, we are in exactly one of the states.

For \( 1 \leq i, j \leq n \), the matrix entry \( P_{ij} \) tells us the probability of \( j \) being the next state given the current state is \( i \).

For all \( i \), \( \sum_{j=1}^{n} P_{ij} = 1 \).

Markov chains are abstractions of random walks – crucial property is that the distribution over next states only depends on the current state, and not how the state was arrived at.

Random Surfer as a Markov Chain

Each state represents a web page; each transition probability represents the probability of moving from one page to another – transition probabilities include teleportation.

Let \( \bar{x}^t \) be the probability vector for time \( t \).

\( x_i^t \) is the probability of being in state \( i \) at time \( t \).

We can compute the surfer’s distribution over the web pages at any time given only the initial distribution and the transition probability matrix \( P \).

\[
\bar{x}^t = \bar{x}^0 P^t
\]
A Markov chain is *ergodic* if the following two conditions hold:

- For any two states $i, j$, there is an integer $k \geq 2$ such that there is a sequence of $k$ states $s_1 = i, s_2, \ldots, s_k = j$ such that $\forall l, 1 \leq l \leq k - 1$, the transition probability $P_{s_l,s_{l+1}} > 0$

- There exists a time $T_0$ such that for all states $j$, and for all choices of start state $i$ in the Markov chain, and for all $t > T_0$, the probability of being in state $j$ at time $t$ is $> 0$

---

**Theorem**: For any ergodic Markov chain, there is a unique steady-state probability distribution over the states, $\pi$, such that if $N(i, t)$ is the number of visits to state $i$ in $t$ steps, then

$$\lim_{t \to \infty} \frac{N(i, t)}{t} = \pi(i),$$

where $\pi(i) > 0$ is the steady-state probability for state $i$.

(Introduction to IR, ch.21)

- $\pi(i)$ is the PageRank for state/web page $i$
• The *left eigenvectors* of the transition probability matrix $P$ are $N$-vectors $\pi$ such that

$$\pi P = \lambda \pi$$

• We want the eigenvector with eigenvalue 1 (this is known as the *principal* left eigenvector of the matrix $P$, and it has the largest eigenvalue)

• This makes $\pi$ the steady-state distribution we’re looking for

---

**PageRank Computation**

• There are many ways to calculate the principal left eigenvector of the transition matrix

• One simple way:
  – Start with any distribution, eg $x = (1, 0, \ldots, 0)$
  – After one step, distribution is $x P$
  – After two steps, distribution is $x P^2$
  – For large $k$, $x P^k = a$, where $a$ is the steady state
  – Algorithm: keep multiplying $x$ by $P$ until the product looks stable
• Putting all the probability mass from $E$ onto a single page produces a personalised importance ranking relative to that page.

• $E$ gives the probabilities of jumping to pages via a random jump.

• Putting all the mass on one page emphasises pages "close to" that page.

HITS

• Hypertext Induced Topic Search (Kleinberg)
  – “Hyperlinks encode a considerable amount of latent human judgement”
  – “The creator of page $p$, by including a link to page $q$, has in some measure conferred authority on $q$”

• Example: consider the query "Harvard"
  – www.harvard.edu may not use Harvard most often
  – but many pages containing the term Harvard will point at www.harvard.edu

• But some links are created for reasons other than conferral of authority, e.g. navigational purposes, advertisements.

• Need also to balance criteria of relevance and popularity
  – e.g. lots of pages point at www.google.com
Hubs and Authorities (for a given query)

- An **authority** is a page which has many relevant pages pointing at it
  - authorities are likely to be relevant (precision)
  - there should be overlap between the sets of pages which point at authorities
- A **hub** is a page which links to many authorities
  - hubs help find relevant pages (recall)
  - hubs ”pull-together” authorities on a common topic
  - hubs allow us to ignore non-relevant pages with a high in-degree
- Relationship between hubs and authorities is mutually reinforcing:
  - a good hub points to many good authorities
  - a good authority is pointed at by many good hubs

Finding Hubs and Authorities

- Suppose we are given some query $\sigma$
- We wish to find authoritative pages with respect to $\sigma$, restricting computation to a relatively small set of pages:
  - recover top-$n$ pages using some search engine: the *root set*
  - add pages which link to the root set and pages which the root set link: the *base set*
- Base set might contain a few thousand documents, with many authorities
  - how do we find the authorities?
• Each page $p$ has a hub weight $h_p$ and authority weight $a_p$
• Initially set all weights to 1
• Update weights iteratively:

$$h_p \leftarrow \sum_{q:p \rightarrow q} a_q$$

$$a_p \leftarrow \sum_{q:q \rightarrow p} h_q$$

– $p \rightarrow q$ means $p$ points at $q$
– weights are normalised after each iteration
– can prove this algorithm converges

• Pages for a given query can then be weighted by their hub and authority weights

Calculating Hub and Authority Weights

Loop($G$, $k$):
$G$: a collection of $n$ linked pages
$K$: a natural number
Let $z$ denote the vector $(1,1,1,...,1) \in \mathbb{R}^n$
Set $a_0 := z$
Set $h_0 := z$
For $i = 1,2,...,k$
Update $\alpha_{i-1}$ obtaining new weights $\alpha'_i$
Update $\bar{h}_{i-1}$ obtaining new weights $\bar{h}'_i$
Normalise $\alpha'_i$ obtaining $\alpha_i$
Normalise $\bar{h}'_i$ obtaining $\bar{h}_i$
Return $(\alpha_k, \bar{h}_k)$
### Example Results for HITS

<table>
<thead>
<tr>
<th>Query</th>
<th>Top Authorities</th>
</tr>
</thead>
<tbody>
<tr>
<td>censorship</td>
<td>.378 <a href="http://www.eff.org/">http://www.eff.org/</a> The Electronic Frontier Foundation</td>
</tr>
<tr>
<td></td>
<td>.344 <a href="http://www.eff.org/blueribbon.html">http://www.eff.org/blueribbon.html</a> Campaign for online free speech</td>
</tr>
<tr>
<td></td>
<td>.238 <a href="http://www.cdt.org/">http://www.cdt.org/</a> Center for democracy &amp; technology</td>
</tr>
<tr>
<td></td>
<td>.235 <a href="http://www.vtw.org/">http://www.vtw.org/</a> Voters telecommunications watch</td>
</tr>
<tr>
<td>&quot;search engines&quot;</td>
<td>.346 <a href="http://www.yahoo.com/">http://www.yahoo.com/</a> Yahoo</td>
</tr>
<tr>
<td></td>
<td>.291 <a href="http://www.excite.com/">http://www.excite.com/</a> Excite</td>
</tr>
<tr>
<td></td>
<td>.239 <a href="http://www.mckinley.com/">http://www.mckinley.com/</a> Welcome to Magellan</td>
</tr>
<tr>
<td></td>
<td>.231 <a href="http://www.lycos.com/">http://www.lycos.com/</a> Lycos home page</td>
</tr>
<tr>
<td></td>
<td>.231 <a href="http://www.altavista.digital.com">http://www.altavista.digital.com</a> AltaVista</td>
</tr>
<tr>
<td>Gates</td>
<td>.643 <a href="http://www.roadahead.com/">http://www.roadahead.com/</a> Bill Gates: The Road Ahead</td>
</tr>
<tr>
<td></td>
<td>.458 <a href="http://www.microsoft.com/">http://www.microsoft.com/</a> Welcome to Microsoft</td>
</tr>
<tr>
<td></td>
<td>.440 <a href="http://www.microsoft.com/corpinfo">http://www.microsoft.com/corpinfo</a></td>
</tr>
</tbody>
</table>

### References

- Introduction to Information Retrieval (online), ch. 21, book material and accompanying slides
- Authoritative Sources in a Hyperlinked Environment (1999), Jon Kleinberg, Journal of the ACM
- The PageRank Citation Ranking: Bringing Order to the Web (1998), Lawrence Page et al.
- The Anatomy of a Large-Scale Hypertextual Web Search Engine, Sergey Brin and Lawrence Page

available online