Creatively solving the problem of Searching for Information

CSE 91 Midway Lecture

George Varghese
Outline

• 1.0 Reminder of the course so far
• 2.0 The Search Problem and Google Story
• 3.0 Page Rank and Random Surfers
• 4.0 Google Business Model
CSE 91 Goals vs Today’s Topic

- **Essence:** To convince you that Computer Science is not just programming but creatively solving the world’s problems using computers (Search)
- **Careers:** To show there are exciting career options that can change the world (Google, Web 2.0)
- **UCSD CSE:** To show you that UCSD CSE has a number of cool professors doing cool work (Grassroots)
- **Startups:** To give you a glimpse of how CSE ideas can convert to business opportunities (Google Story)
- **Students:** To showcase students like you (Page, Brin)
1.0 Summary of Earlier Lectures
Summary of First 3 Lectures

• Lecture 1, Rondon, Programming Languages
  – **Problem:** teaching ordinary users to program computers using intuitive languages: circle, squares
  – **Main idea:** Procedural abstraction for any domain. Circle -> Face
  – **Computer Science:** Languages, Semantics (correctness proofs)

• Lecture 2, Stefan Savage, Security
  – **Problem:** automation can be misused for say spam
  – **Main idea:** Study economic model, think like miscreant. First step: infiltrate Storm to find spam conversion rate
  – **Computer Science:** Distributed Systems, OS, Security

• Lecture 3, Griswold, Sensors and Ubicomp
  – **Problem:** Dealing with a polluted environment
  – **Main Idea:** Equipping each cell phone with cheap sensor, sharing info
  – **Computer Science:** Software Engineering, Sensor HW, security
Summary of Next 3 Lectures

• Lecture 4, Marzullo, Distributed Systems
  – Problem: complicated failures when computers work together
  – Main idea: Two generals: consensus impossible with message failures. Byzantine Generals possible by $3F + 1$ machines and $F$ failures

• Lecture 5, Ettinger, Learning Theory
  – Problem: automatic camera tracking for say Elder Care
  – Main idea: Learning theory (features, training); audio localization via 3 mikes, face detection using sliding window looking for features
  – Computer Science: Learning theory, AI, vision, speech processing

• Lecture 6, Agarwal, Power-aware computing
  – Problem: Reducing power footprint of cell phones and computers
  – Main Idea: Turn off power hungry WiFi or processor and wake up using cheaper (cell radio or USB processor) device
  – Computer Science: Systems, OS, processors, networks, sensors
Summary of Next 2 Lectures

• Lecture 7, Kriegman, Building a Makeover site
  – Problem: Image editing for the masses, Photoshop too low level
  – Main idea: Matt-gloss separation: leveraging feature into product. Photo touchup --(small profits). Cosmetics ++ (large profits)
  – Computer Science: Vision, startups

• Lecture 8, Elkan, Recommendation Systems
  – Problem: Selling the long-tail (e.g., lesser known movies) requires good automated recommendation systems. NetFlix prize
  – Computer Science: Learning theory, calculus, linear algebra
2.0 The Google Story
Search Problem

- **Older model**: Information in books in libraries. Skilled librarian great help
- **New model**: Berners-Lee invents web. Huge amount of information on web sites. Some curated (e.g., Wikipedia) but most not
- **Noise**: Lots of junk and wrong information. Manipulative web sites. How to distinguish
- **Search engines**: Automate work of reference librarian. Good quality info, fast response, very reliable (dial tone)
- **Business model**: Crawling and storing billions of pages takes money. How to get revenue without biasing search?
Search before Larry and Sergey

- **First movers**: Pinkerton’s 1994 web crawler followed links and indexed. Lycos, Altavista
- **Quality Problems**: Poor quality results. Search for “Bill Clinton” → ”Bill Clinton sucks”
- **Scale problems**: Web was growing and most engines harvested only a small fraction
- **Business model**: Nobody knew how to make money to justify a scalable infrastructure
Google Story

• Page and Brin become friends at Stanford
• Page tells advisor that he can load whole web on his computer in a week. Brin joins for “data mining”.
• They decide that Search engines are bad, and use lots of commodity PCs to download more pages.
• When they run out of machines, Andy Bechtolsheim gives them a 100K cheque
• Convince Kleiner-Perkins and Sequoia to give them 25M. Unheard of combination.
• Eric Schmidt joins as CEO several years later
A cheque for 100K with no questions asked
3.0 How it works: Page Rank
Page rank: first cut

- **Inspiration:** Page’s Dad was a scientist. Scientist’s measure quality by “citation count”
- **Idea:** So why not consider number of links that point to a page as a citation or page rank?
- **Problem:** Real science is self-selecting. But on web anyone can publish.
- **Goal:** Surely, we must distinguish a link from Wikipedia from one by Joe Blow.
Page rank: second cut

• **Intuition 1**: Importance is sum of importance of pages that link to me normalized by number of links.
• **Intuition 2**: Probability that a random surfer will land on my page
• **Formally**: $PR(A) = \frac{PR(T_1)}{C(T_1)} + \ldots + \frac{PR(T_n)}{C(T_n)}$
  - $PR(A)$ is the PageRank of page $A$,
  - $PR(T_i)$ is the PageRank of pages $T_i$ which link to page $A$,
  - $C(T_i)$ is the number of outbound links on page $T_i$
• **Problem:** Pages A, B, C get zero rank as there is an influence sink. Formally, not a stochastic matrix

• **Solution:** Allow surfer to get bored with some probability D and jump randomly to any page
Page rank: third cut

- **Intuition 2:** Probability that a random surfer will land on my page with boredom factor $D$

- **Formal:** $PR(A) = (1 - D) + D \left( \frac{PR(T_1)}{C(T_1)} + \ldots + \frac{PR(T_n)}{C(T_n)} \right)$
  - $PR(A)$ is the PageRank of page $A$,
  - $PR(T_i)$ is the PageRank of pages $T_i$ which link to page $A$,
  - $C(T_i)$ is the number of outbound links on page $T_i$
  - $D$ is damping (boredom) factor, recommended 0.85

Can be made a probability (sum to 1) by doing $(1-D)/N$
How to calculate: Take 1

- Not clear how to find because each page rank is defined in terms of others.
- Best to think of as a system of equations. Or better still as a matrix $\rightarrow$ eigenvectors
- Simple iteration works: start with 1 for all ranks and substitute and iterate till converge
Page Rank Iteration: $D = 0.5$

- Iteration | PR(A) | PR(B) | PR(C)
- 1         | 1     | 0.75  | 1.125
- 2         | 1.06  | 0.76  | 1.15
Iterative Computation is too slow

• Billion element matrix! Can take millions of iterations to converge
• Better to use power method: $A \rightarrow A$ squared $\rightarrow A$ fourth by repeated squaring of matrix
• Formally, depends on second eigenvalue which is damping factor $D$.
• Higher $D$, slower to converge. Lower $D$, completely random, no link structure used.
• **Compromise:** $D = 0.85$ suggested
Page rank: fourth cut

- **Intuition 1**: Content of web page matters as well. Keywords in italics or larger font mean that keyword is more important in page.
- **Intuition 2**: Compute an IR score based on repetition count and font for keyword. IR pioneer: Salton.
- **Formal**: Score of a page $P$ for keyword $K$ is product of IR score for $K$ on $P$ and Page Rank of $P$
  - Crawl all web pages and compute page rank
  - Index for keywords and compute IR score
  - On query, return Top $k$ scored pages that match keyword
Search Engine Optimization

- Can we manipulate Page Rank by repeating keywords?
  - Not much, IR score for repeats is bounded
- Can we fool Page Rank by pointing to each other?
  - No, sum of Page ranks of nodes in a set stays same regardless of how they point within set.
- Can do more easily today by hijacking a page!
  - See example of fitness.cornell.edu next slide
fitness.cornell.edu hijacked?
4.0 How Google makes money
Business Model

• Banner ads traditional way to make money but banner adds have poor click-through rates
• Overture invents new model: if companies pay more, more likely to be in search answers. In long-run, such a model loses user trust.
• Brin and Page want unbiased search. So they separate sponsored ads from true search
• As much of an innovation as page rank. Started making money immediately. Democratizes ads
A search for Roses produces info on Roses
But also paid for ads keyed on “Roses”
How are Ad Prices Set?

• Intuitively, some keywords are worth more money (e.g., watches) as watches cost more than say “Comics”
• Normally, price setting is done by a human. Hard to do with millions of keywords, and disallows small merchants.
• So Google does an automated auction. You bid a price for a keyword (like Ebay).
• Google also measures how often people click your ads.
• Winner of auction is ad with highest value of Price times click through rate, not highest price!
• Subject to click-through fraud (security). Can you guess how?
The Google Empire

• **News**: very different technology, Customizable newspaper. Clustering of stories automatically by topic

• **Gmail**: Ads based on your mail. Justified by anti-virus scanning programs.

• **Sets, Spell, Flu, Suggestions**: All based on vast search data. For example, Google sets finds sets listed in HTML lists
Summary

• **Problem: Search**
  – Only way to make billions of web pages useful, separate wheat from chaff
  – Page rank is a refined use of citation counts with some content (IR) score
  – Random surfer model, matrix computation, damping factor D
  – Lots more stuff, Google News, Gmail, Sets: part of our lives

• **Business models matter**
  – Overture Model led to biased search. Users wouldn’t trust
  – Automation via a second price auction. Price + popularity Very cool

• **People like you**
  – Serge and Larry were curious and passionate
  – *Did not try to make money but to solve a real world problem: still at it*
Finally the Homework

• Play with Google to look under the hood.
• Does order of words matter? Does distance between words?
• How do Ad words work?
You, too, can make the world better . .