1.6 Case Study: Random Surfer

- Follow links from book or film to another.
- Tool for establishing links.

World Wide Web


Web Browser

Web browser. Killer application of the 1990s.
When it was proclaimed that the Library contained all books, the first impression was one of extravagant happiness... There was no personal or world problem whose eloquent solution did not exist in some hexagon.

this inordinate hope was followed by an excessive depression. The certitude that some shelf in some hexagon held precious books and that these precious books were inaccessible seemed almost intolerable.
**PageRank**

*Google’s PageRank™ algorithm.* [Sergey Brin and Larry Page, 1998]
- Measure popularity of pages based on hyperlink structure of Web. Revolutionized access to world’s information.

**Web Graph Input Format**

- Input format:
  - N pages numbered 0 through N-1.
  - Represent each hyperlink with a pair of integers.

**Transition Matrix**

- Transition matrix. \( p[i][j] \) = prob. that surfer moves from page \( i \) to \( j \).

**90-10 Rule**

- **Model.** Web surfer chooses next page:
  - 90% of the time surfer clicks random hyperlink.
  - 10% of the time surfer types a random page.

- **Caveat.** Crude, but useful, web surfing model.
  - No one chooses links with equal probability.
  - No real potential to surf directly to each page on the web.
  - The 90-10 breakdown is just a guess.
  - It does not take the back button or bookmarks into account.
  - We can only afford to work with a small sample of the web.
  - …
```java
public class Transition {
    public static void main(String[] args) {
        int N = StdIn.readInt(); // # number of pages
        int[][] counts = new int[N][N]; // # links from page i to j
        int[] outDegree = new int[N]; // # links from page

        // accumulate link counts
        while (!StdIn.isEmpty()) {
            int i = StdIn.readInt();
            int j = StdIn.readInt();
            outDegree[i]++;
            counts[i][j]++;
        }

        // print transition matrix
        StdOut.println(N + " "+ N);
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                double p = .90*counts[i][j]/outDegree[i] + .10/N;
                StdOut.printf("%7.5f ", p);
            }
            StdOut.println();
        }
    }
}
```

**Monte Carlo Simulation**

Monte Carlo simulation.

- Surfer starts on page 0.
- Repeatedly choose next page, according to transition matrix.
- Calculate how often surfer visits each page.
Random move. Surfer is on page \textit{page}. How to choose next page \textit{j}?

- Row \textit{page} of transition matrix gives probabilities.
- Compute \textbf{cumulative} probabilities for row \textit{page}.
- Generate random number \( r \) between 0.0 and 1.0.
- Choose page \textit{j} corresponding to interval where \( r \) lies.

\[
\begin{bmatrix}
.02 & .02 & .02 & .02 \\
.02 & .02 & .38 & .20 \\
.02 & .02 & .02 & .02 \\
.47 & .02 & .47 & .02 \\
\end{bmatrix}
\]

\textit{transition matrix}

\[
\begin{array}{cccc}
0 & 1 & 2 & 3 \\
.47 & .47 & .96 & .98 \\
\end{array}
\]

\textit{cumulated sum values}

\[
\begin{array}{cccc}
j & p[page][j] \quad & \text{cumulated sum values} \\
0 & .47 & .47 & .96 & .98 \\
1 & \text{\textbackslash} & \text{\textbackslash} & \text{\textbackslash} & \text{\textbackslash} & \text{\textbackslash} & \text{\textbackslash} \\
0.0 & .47 & .49 & .96 & .98 \\
\end{array}
\]

\textit{generate: } .71, \textit{return: } \textit{page}

\[
\text{Mathematical Context}
\]

\textit{Convergence.} For the random surfer model, the fraction of time the surfer spends on each page converges to a unique \textbf{distribution}, independent of the starting page.

\[\begin{array}{cccc}
0.27 & 1.27 & 3.25 & 2.15 \\
4.07 & \text{428.671} & \text{417.205} & \text{229.519} & \text{388.162} & \text{106.498} \\
\end{array}\]

\textit{"page rank"} \\
\textit{"stationary distribution"} of Markov chain \\
\textit{"principal eigenvector"} of transition matrix
The Power Method

Q. If the surfer starts on page 0, what is the probability that surfer ends up on page i after one step?

A. First row of transition matrix.

\[
\begin{bmatrix}
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
\end{bmatrix} \times
\begin{bmatrix}
\begin{array}{cccc}
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
\end{array}
\end{bmatrix} =
\begin{bmatrix}
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
\end{bmatrix}
\]

Q. If the surfer starts on page 0, what is the probability that surfer ends up on page i after two steps?

A. Matrix-vector multiplication.

\[
\begin{bmatrix}
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
\end{bmatrix} \times
\begin{bmatrix}
\begin{array}{cccc}
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
\end{array}
\end{bmatrix} =
\begin{bmatrix}
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 \\
\end{bmatrix}
\]

Power method. Repeat until page ranks converge.
Mathematical Context

Convergence. For the random surfer model, the power method iterates converge to a unique distribution, independent of the starting page.

Random Surfer: Scientific Challenges

Google’s PageRank™ algorithm. [Sergey Brin and Larry Page, 1998]
- Rank importance of pages based on hyperlink structure of web, using 90-10 rule.
- Revolutionized access to world’s information.

Scientific challenges. Cope with 4 billion-by-4 billion matrix!
- Need data structures to enable computation.
- Need linear algebra to fully understand computation.