FAQs

• Duplicated printout for PA1
  
  In reducer
  
  For (String s: record.keySet)
      
      context.write (id, new Text(s + "---" + record.get(s)));
      record.clear();
  
  output:
      and-2---1
      are-1---1

• Your combiner creates the key as " and-2" and the frequency of this key is recorded as 1

• PA2 has been posted

Topics

• Large-scale Analytics 1. Web-Scale Link and Social Network Analysis

This material is built based on,

  • Chapter 5

  • http://infolab.stanford.edu/~ullman/mmds.html

What are these?

• JumpStation
• Go
• Infoseek
• Snap
• Direct Hit
• Lycos
• AltaVista
• Excite
• Yahoo
• Google
Early Search Engines

• They worked by crawling the Web and listing the terms
  • Words or other strings of characters other than white space
  • In an inverted index

• An inverted index is a data structure that makes it easy, given a term, to find (pointer to) all the places where that term occurs

Inverted index (1/2)

• Inverted index
  • For given texts,
    - T[0] = "Colorado State University"
    - T[1] = "Colorado water source"
    - T[2] = "University of Colorado"

• We have the following inverted file index

<table>
<thead>
<tr>
<th>Term</th>
<th>Set of Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colorado</td>
<td>{0, 1, 2}</td>
</tr>
<tr>
<td>State</td>
<td>{0}</td>
</tr>
<tr>
<td>Source</td>
<td>{1}</td>
</tr>
<tr>
<td>University</td>
<td>{0, 2}</td>
</tr>
<tr>
<td>Water</td>
<td>{1}</td>
</tr>
</tbody>
</table>

Inverted index (2/2)

• A term search for the terms, "Colorado", "State", and "University" would give the set
  \(\{0, 1, 2\} \cap \{0\} \cap \{0, 2\} = \{0\}\)

Term spam

• If you were selling toaster on the Web
  • All you care about was that people would see your page
  • You could add a term like "movie" to your page
    • Add thousands of times
    • It does not even need to show
    • Give the same color as background to the letters
  • A search engine would think this page is very important one about "movie"
  • You could go to the search engine and search "movie" and see the first listed page
  • Copy that page with the same color as background

Part 1. Large Scale Data Analytics

1. Web-Scale Link Analysis and Social Network Analysis

Web-Scale Link Analysis: PageRank Algorithm
PageRank

- Goals
  - Providing effective summaries for the search results
  - Ordering/Ranking results

- Simulate random Web surfers
  - Pages that would have a large number of surfers were considered more “important” than pages that would rarely be visited

- The content of a page was judged not only by the terms appearing on that page
  - But by the terms used in or near the links to that page

Definition of PageRank

- A function that assigns a real number to each page in the Web
- The higher the PageRank of a page, the more “important” it is
- There is NOT one fixed algorithm for assignment of PageRank

Example

- Page A has links to B, C and D
- Page B has links to A and D
- Page C has a link to A
- Page D has links to B and C
Example [2/5]

• Suppose that a random surfer starts at page A
• Page B, C and D will be the next with probability 1/3
• 0 probability of being at A

Example [3/5]

• Now suppose the random surfer at B
• B has probability of 1/2 of being at A, 1/2 of being at D and 0 of being at B or C

Example [4/5]

• Transition matrix $M$
  • What happens to random surfers after one step
  • $M$ has $n$ rows and columns ($n$ pages)
  • What is the transition matrix for this example?

Example [5/5]

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

• The first column
  • A surfer at A has a 1/3 probability of next being at each of the other pages
• The second column
  • A surfer at B has a 1/3 probability of being next at A and the same for being at D

What does this matrix mean? [1/6]

• The probability distribution for the location of a random surfer
  • A column vector whose $j$th component is the probability that the surfer is at page $j$

What does this matrix mean? [2/6]

• If we surf at any of the $n$ pages of the Web with equal probability
  • The initial vector $v_0$ will have $1/n$ for each component
  • If $M$ is the transition matrix of the Web
    • After the first step, the distribution of the surfer will be $Mv_0$
    • After two steps, $Mv_0 = M^2v_0$ and so on
What does this matrix mean? [3/6]
• Multiplying the initial vector \( v_0 \) by \( M \) a total of \( i \) times
  • The distribution of the surfer after \( i \) steps

\[
\begin{pmatrix}
0 & 1/2 & 1/3 & 1/3 \\
1/3 & 0 & 0 & 1/2 \\
1/3 & 0 & 0 & 1/2 \\
1/3 & 1/2 & 0 & 0 \\
\end{pmatrix}
\]

The probability for being in the current location

• The distribution of the surfer after the first step

The probability for being in the next possible location

The probability for the next step from the current location

The probability for the next step

What does this matrix mean? [4/6]
• The probability \( x_i \) that a random surfer will be at node \( i \) at the next step
  \[
x_i = \sum_j M_{ij} v_j
\]
• \( m_{ij} \) is the probability that a surfer at node \( j \) will move to node \( i \) at the next step
• \( v_j \) is the probability that the surfer was at node \( j \) at the previous step

What does this matrix mean? [5/6]
• The distribution of the surfer approaches a limiting distribution \( v \) that satisfies \( v = Mv \) provided two conditions are met:
  1. The graph is strongly connected
     • It is possible to get from any node to any other node
  2. There are no dead ends
     • Nodes that have no arcs out

What does this matrix mean? [6/6]
• The limit is reached when multiplying the distribution by \( M \) another time does not change the distribution
  • The limiting \( v \) is an eigenvector of \( M \)
  • Since \( M \) is stochastic (its columns each add up to 1), \( v \) is the principle eigenvector
  • Its associated eigenvalue is the largest of all eigenvalues

• The principle eigenvector of \( M \)
  • Where the surfer is most likely to be after a long time

• For the Web, 50-75 iterations are sufficient to converge to within the error limits of double-precision arithmetic

Example

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

• Suppose we apply this process to the matrix \( M \)
• The initial vector \( v_0 \) and \( v_1 \) after multiplying \( M \)

Example

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

• Suppose we apply this process to the matrix \( M \)
• The initial vector \( v_0 \) and \( v_1 \) after multiplying \( M \)

\[
\begin{pmatrix}
0 & 1/2 & 1/3 & 1/3 \\
1/3 & 0 & 0 & 1/2 \\
1/3 & 0 & 0 & 1/2 \\
1/3 & 1/2 & 0 & 0 \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
0.25 & 0.25 & 0.25 & 0.25 \\
0.25 & 0.25 & 0.25 & 0.25 \\
0.25 & 0.25 & 0.25 & 0.25 \\
0.25 & 0.25 & 0.25 & 0.25 \\
\end{pmatrix}
\]
What is the \( v_2 \)?

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

- Suppose we apply this process to the matrix \( M \)
- The initial vector \( v_0 \) and \( v_1 \) after multiplying \( M \)

Example continued

- The sequence of approximations to the limit
- We get by multiplying at each step by \( M \)

\[
\begin{array}{cccc}
\frac{1}{24} & \frac{5}{48} & \frac{11}{32} & \frac{3}{9} \\
\frac{1}{24} & \frac{5}{48} & \frac{11}{32} & \frac{7}{32} \\
\frac{1}{24} & \frac{5}{48} & \frac{11}{32} & \frac{7}{32} \\
\frac{1}{24} & \frac{5}{48} & \frac{11}{32} & \frac{7}{32} \\
\end{array}
\]

- This difference in probability is not noticeable
- In the real Web, there are billions of nodes of greatly varying importance
  - The probability of being at a node like www.amazon.com is orders of magnitude greater than others

Matrix-vector Multiplication by MapReduce

1. Suppose we have an \( n \times n \) matrix \( M \) whose element in row \( i \) and column \( j \) will be denoted \( M_{ij} \)
2. Then the matrix-vector product is the vector \( x \) of length \( n \), whose \( i \)th element \( x_i \) is given by,

\[
x_i = \sum_{j=1}^{n} M_{ij} v_j
\]
The Map function

- The Map function is written to apply to one element of $M$
- Each Map task will operate on a chunk of the matrix $M$
- From each matrix element $m_{ij}$, it produces the key-value pair $(i, m_{ij})$
- All terms of the sum that make up the component $x_k$ of the matrix-vector product will get the same key, $k$

The Reduce function

- Sums all the values associated with a given key $k$
- The result will be a pair $(k, x_k)$
  
$$x_k = M_{k0} x_0 + M_{k1} x_1 + M_{k2} x_2 + \ldots + M_{kn} x_n$$

If the vector $v$ cannot fit in main memory?

- It is possible that the vector $v$ is so large that it will not fit in its main memory entirely
- We can divide the matrix into vertical stripes of equal width and divide the vector into an equal number of horizontal stripes of the same height
- The goal is to use enough stripes so that the portion of the vector in one stripe can fit into main memory
Structure of Web (1/3)

- Is the Web strongly connected?

Structure of Web (2/3)

- Strongly Connected Component
  - Consisting of pages reachable from the In Component but not unable to reach the SCC
  - Consisting of pages reachable from the SCC, but unable to reach the SCC

- In Component
  - Consisting of pages reachable from the SCC, but not unable to reach the SCC
  - Consisting of pages that could reach by following links but not reachable from the SCC

- Out Component
  - Consisting of pages reachable from the In Component but not unable to reach the In Component

- Tendrils In
  - Consisting of pages reachable from the In Component and able to reach the output-component, but unable to reach the SCC or be reached from the SCC

- Tendrils Out
  - Consisting of pages that could reach by following links but not reachable from the SCC

Structure of Web (3/3)

- Tubes
- Pages reachable from the in-component and able to reach the output-component, but unable to reach the SCC or be reached from the SCC

- Isolated components
- Unreachable from the large components

Anomalies from the Web structure

- These structures violate the assumptions needed for the Markov process iteration to converge to a limit
  - When a random surfer enters the out-component, they can never leave
  - Surfers starting in either the SCC or in-component are going to wind up in either the out-component or a tendril off the in-component
  - No page in the SCC or in-component winds up with any probability of a surfer being there

- Nothing in the SCC or in-component will be of any importance

Problems we need to avoid

- Dead end
  - A page that has no links out
  - Surfers reaching such a page will disappear
  - In the limit, no page that can reach a dead end can have any PageRank at all

- Spider traps
  - Groups of pages that all have outlinks but they never link to any other pages
Questions?