PART 1. BATCH COMPUTING MODELS FOR BIG DATA ANALYTICS
1. DISTRIBUTED MODEL FOR SCALABLE BATCH COMPUTING - MapReduce

Sangmi Lee Pallickara
Computer Science, Colorado State University
http://www.cs.colostate.edu/~cs535

FAQs
- Programming Assignment 1
  - Due Sept. 28
  - Submission via Canvas
  - Please check the course Web Page at least twice a week
- Use the configuration file with memory size 2GB per node
  - Current one on the web page

Today's topics
- MapReduce with examples
- Understanding PageRank algorithm and implementing with MapReduce

Matrix-Vector Multiplication Using MapReduce

Matrix-Vector multiplication using MapReduce (1/3)
- Suppose we have an \( n \times n \) matrix \( M \), whose element in row \( i \) and column \( j \) will be denoted \( M_{ij} \)
- \( \mathbf{v} \) is a \( n \times 1 \) column vector
- Then the matrix-vector product is the vector \( \mathbf{x} \) of length \( n \), whose \( i^{th} \) element \( x_i \) is given by:

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

Matrix-Vector multiplication using MapReduce (2/3)
- If \( n = 100 \), we do not need MapReduce
- However, if this calculation is a part of ranking Web pages (\( n \) could be \( 10^7 \)) performed by the search engine?
  - The vector \( \mathbf{v} \) cannot fit in main memory
Matrix-Vector multiplication using MapReduce (3/3)

- The matrix $M$ and the vector $v$ will each be stored in a file of the DFS (e.g. HDFS)
- Assume that row-column coordinates of each matrix element will be discoverable
  - Either from its position in the file or explicit coordinates

The Map function

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

The Reduce function

- Sums all the values associated with a given key $i$
  - $n$ number of pairs per $i$
- The result will be a pair $(i, x_i)$
  - Sort pairs in ascending order of $i$
  - $(x_0, x_1, ..., x_i)$

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 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

Programming Assignment 1:
Estimating PageRank Values of Wikipedia Articles using MapReduce
Objectives

- The goal of this programming assignment is to enable you to gain experience in:
  - Implementing iterative algorithms to estimate PageRank values of Wikipedia articles
  - Designing and implementing batch layer computations using Apache Spark and HDFS

Tasks

- Estimation of PageRank values under ideal conditions
- Estimation of the PageRank values while considering dead-end articles
- Analysis of the above results
  - Implement a Wiki bomb

Requirements

- A sorted list of Wikipedia pages based on their ideal PageRank value in descending order
- A sorted list (in descending order) of Wikipedia pages based on their PageRank value with taxation
- Average difference between the ideal PageRank (from A) and PageRank with taxation (from B) for each Wikipedia article

Requirements - Continued

- This computation should be performed in your own Hadoop cluster with 10 machines
- You should present results from at least 25 iterations
- You should use the algorithms included in this description
  - You are NOT allowed to use existing PageRank implementations
- Demonstration of your software should be on machines in CSB-120
- Demonstration will include an interview discussing implementation and design details
- Your submission should include your source codes
  - Your submission should be via Canvas

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 are many algorithms 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

- 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 (2/3)

- Now the random surfer at B has probability of $\frac{1}{2}$ of being at A, $\frac{1}{2}$ of being at D and 0 of being at B or C
- Transition matrix $M$
  - What happens to random surfers after one step?
  - $M$ has $n$ rows and columns
  - What is the transition matrix for this example?

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

- The first column
  - A surfer at A has a $\frac{1}{3}$ probability of being at each of the other pages next
- The second column
  - A surfer at B has a $\frac{1}{2}$ probability of being at A next and the same for being at D

What does this matrix mean? (1/4)

- The probability distribution for the location of a random surfer
  - A column vector whose $i$th component is the probability that the surfer is at page $i$
- If we surf any of the $n$ pages of the Web with equal probability
  - The initial vector $v_0$ will have $\frac{1}{n}$ for each component
  - If $M$ is the transition matrix of the Web
    - After one step, the distribution of the surfer will be $Mv_0$
    - After two steps, $M(Mv_0) = M^2v_0$ and so on
  - Multiplying the initial vector $v_0$ by $M$ a total of $i$ times
    - The distribution of the surfer after $i$ steps

What does this matrix mean? (2/4)

- 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? (3/4)

- 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
  2. There are no dead ends
     - Dead ends==Nodes that have no arcs out

What does this matrix mean? (4/4)

- 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
- For the Web, 50-75 iterations are sufficient to converge to within the error limits of double-precision arithmetic
• Suppose we apply this process to the matrix $M$

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

- The initial vector $v_0$ and $v_1$ after multiplying $M$

\[
v_0 = \begin{bmatrix}
0 \\
1/3 \\
1/3 \\
0 \\
\end{bmatrix},
v_1 = \begin{bmatrix}
1/2 \\
0 \\
0 \\
1/2 \\
\end{bmatrix}
\]

- The sequence of approximations to the limit we get by multiplying at each step by $M$ is:

\[
\begin{array}{cccc}
\frac{1}{4} & 9/24 & 15/48 & 3/9 \\
\frac{1}{4} & 5/24 & 11/48 & 2/9 \\
\frac{1}{4} & 5/24 & 11/48 & 2/9 \\
\frac{1}{4} & 5/24 & 11/48 & 2/9 \\
\end{array}
\]

- 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

Avoiding Dead Ends

- If we allow dead ends

  - The transition matrix of the Web is no longer stochastic
  - Some of the columns will sum to 0 rather than 1

- If we compute $Mv$ for increasing powers of a substochastic matrix

  - Some of all of the components of the vector go to 0
  - A matrix whose column sums are at most 1
  - Importance “drains out” of the Web
  - No information about the relative importance of pages

Example: Dead end

- Remove the arc from C to A?

- C becomes a dead end

- If a random surfer reaches C, they disappear at the next round

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

Approaches to dealing with dead ends

- Recursive deletion
  - Drop dead ends from the graph
  - Drop their incoming arcs as well
  - Doing so may create more dead ends
  - Drop those new dead ends

- Taxation
  - Modify the process by which random surfers are assumed to move about the Web
Example of recursive deletion (1/4)

![Graph with nodes A, B, C, D, and E connected by directed edges]

Example of recursive deletion (2/4)
- The final matrix for the graph is:

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

- Matrix:

\[
\begin{bmatrix}
\frac{1}{3} & \frac{1}{6} & \frac{1}{12} & \frac{1}{24} & \ldots & \frac{1}{9} \\
\frac{1}{3} & \frac{3}{6} & \frac{5}{12} & \frac{11}{24} & \ldots & \frac{4}{9} \\
\frac{1}{3} & \frac{2}{6} & \frac{4}{12} & \frac{8}{24} & \ldots & \frac{3}{9}
\end{bmatrix}
\]

Example of recursive deletion (3/4)
- We still need to compute PageRank for deleted nodes (C and E)
- C was the last to be deleted
  - We know all its predecessors have PageRanks (A and D)
  - Therefore,
    - PageRank of C = \(\frac{1}{3} \times \frac{2}{9} + \frac{1}{2} \times \frac{3}{9} = \frac{13}{54}\)

Example of recursive deletion (4/4)
- Now, we can compute the PageRank for E
  - Only one predecessor, C
    - The PageRank of E is the same as that of C (13/54)
  - The sums of the PageRanks exceeds 1
    - It cannot represent the distribution of a random surfer
    - But it provides a good estimate

PageRank using Taxation (1/5)
- To avoid dead ends, we modify the calculation of PageRank
  - Allow each random surfer a small probability of teleporting to a random page
  - Rather than following an out-link from their current page

PageRank using Taxation (2/5)
- The iterative step, where we compute a new vector estimate of PageRanks \(v\) from the current PageRank estimate \(v\) and the transition matrix \(M\) is:

\[
v' = \beta M v + (1 - \beta) e / n
\]

- Where \(\beta\) is a chosen constant
  - Usually in the range 0.8 to 0.9
- \(e\) is a vector with 1’s for the appropriate number of components
- \(n\) is the number of nodes in the Web graph
PageRank using Taxation (3/5)

- The term $\beta M v$ represents the case where:
  - With probability $\beta$, the random surfer decides to follow an out-link from their present page.

- The term $(1-\beta)e/n$ is a vector:
  - Each of whose components has value $(1-\beta)/n$.
  - Represents the introduction, with probability $1-\beta$, of a new random surfer at a random page.

$$v' = \beta M v + (1-\beta)e / n$$

PageRank using Taxation (4/5)

- If the graph has no dead ends:
  - The probability of introducing a new random surfer is exactly equal to the probability that the random surfer will decide not to follow a link from their current page.
  - Surfer decides either to follow a link or teleport to a random page.

PageRank using Taxation (5/5)

- If the graph has dead ends:
  - The surfer goes nowhere.
  - The term $(1-\beta)e/n$ does not depend on the sum of the components of the vector $v$; there will be some fraction of a surfer operating on Web.
  - When there are dead ends, the sum of the components of $v$ may be less than 1.
  - But it will never reach 0.

Example of Taxation (1/2)

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

$$v' = \beta M v + (1-\beta)e / n$$

$$\beta = 0.8$$

$$v' = \begin{bmatrix} 4/15 & 0 & 0 & 1/20 \\ 0 & 2/5 & 0 & 1/20 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{bmatrix} v + \begin{bmatrix} 1/20 \\ 1/20 \\ 1/20 \\ 1/20 \end{bmatrix}$$

Example of Taxation (2/2)

- For the first few iterations:

<table>
<thead>
<tr>
<th>Iteration</th>
<th>v_1</th>
<th>v_2</th>
<th>v_3</th>
<th>v_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/4</td>
<td>9/60</td>
<td>41/300</td>
<td>543/4500</td>
</tr>
<tr>
<td>2</td>
<td>1/360</td>
<td>13/60</td>
<td>53/300</td>
<td>707/4500</td>
</tr>
<tr>
<td>3</td>
<td>1/360</td>
<td>13/60</td>
<td>53/300</td>
<td>707/4500</td>
</tr>
<tr>
<td>4</td>
<td>1/360</td>
<td>13/60</td>
<td>53/300</td>
<td>707/4500</td>
</tr>
</tbody>
</table>

Questions?