MapReduce 实战项目之 PageRank

赵敏 老师

扫描二维码关注微信/微博
获取最新IT面试情报及权威解答

微信: ninechapter
知乎专栏: http://zhuanlan.zhihu.com/jiuzhang
微博: http://www.weibo.com/ninechapter
官网: www.jiuzhang.com

Outline

- What is PageRank
- Why is PageRank
- Basic theory behind PageRank
- Implement PageRank with MapReduce

What is PageRank

- PageRank is an algorithm used by Google Search to rank websites in their search engine results.
- Beats Yahoo!

Why is PageRank

When I search “big data”, there will be 261,000,000 results

Why is PageRank

How to rank?

Why is PageRank

How to rank in the past

- Use Title
- Use keyword density
- Manually
- ...

Why is PageRank

Disadvantage of the old rank method?

Why is PageRank

- Not accurate
- Easily manipulated by human

Why is PageRank

So PageRank came out!

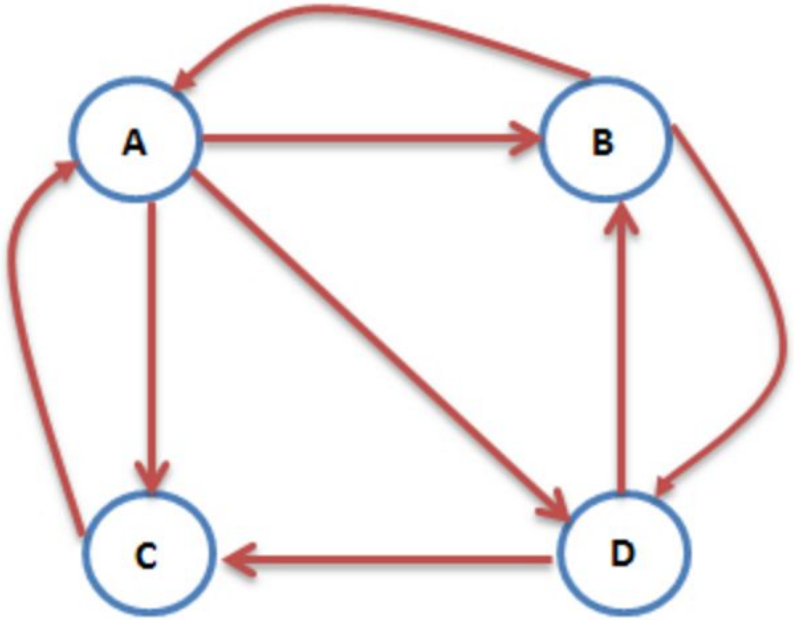
Basic Theory behind PageRank

Self-rank is not accurate, let's involve everyone to rank!

Basic Theory behind PageRank

- 数量假设：More important websites are likely to receive more links from other websites
- 质量假设：Website with higher PageRank will pass higher weight

Basic Theory behind PageRank



A → B, C, D

B → A, D

C → A

D → B, C

Basic Theory behind PageRank

- More important websites are likely to receive more links from other websites

How to represent the directivity between pages?

Basic Theory behind PageRank

九章算法

Transition Matrix

A → B, C, D

B → A, D

C → A

D → B, C

To\From	WA	WB	WC	WD
WA	0	1/2	1	0
WB	1/3	0	0	1/2
WC	1/3	0	0	1/2
WD	1/3	1/2	0	0

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

Basic Theory behind PageRank

- Website with higher PageRank will pass higher weight

How to represent the importance of each website?

Basic Theory behind PageRank

九章算法

PageRank Matrix

	PR0
A	1/4
B	1/4
C	1/4
D	1/4

Basic Theory behind PageRank

How to calculate PR1? → PR1 = PR0 * Transition Matrix

To\From	WA	WB	WC	WD
WA	0	1/2	1	0
WB	1/3	0	0	1/2
WC	1/3	0	0	1/2
WD	1/3	1/2	0	0

×

	PR0
A	1/4
B	1/4
C	1/4
D	1/4

→

	PR1
A	9/24
B	5/24
C	5/24
D	5/24

$$\mathbf{AB} = \begin{pmatrix} a & b & c \\ p & q & r \\ u & v & w \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ax + by + cz \\ px + qy + rz \\ ux + vy + wz \end{pmatrix}$$

Basic Theory behind PageRank

How to calculate PR2? → PR2 = PR1 * Transition Matrix

Basic Theory behind PageRank

How to calculate PR(N)? ⟶ PRN = PR(N-1) * Transition Matrix

Basic Theory behind PageRank

Will this iteration be infinite?

Basic Theory behind PageRank

九章算法

No! The matrix PR will finally converge

Basic Theory behind PageRank

九章算法

Can you think of some edge cases?

Basic Theory behind PageRank

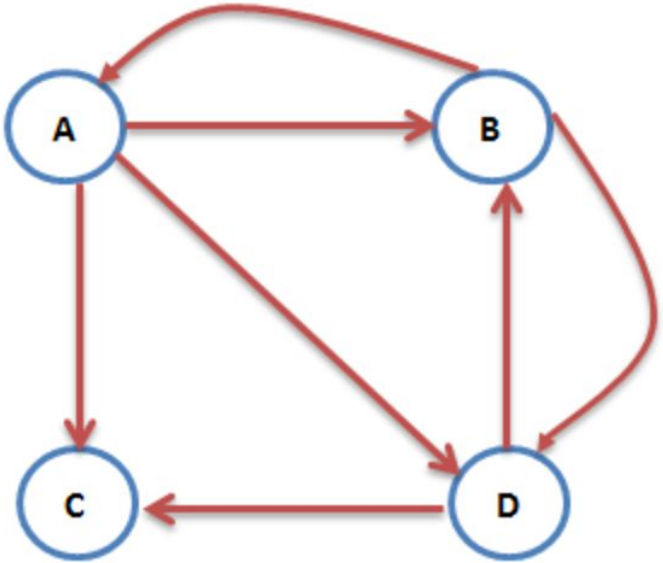
九章算法

- Dead ends
- Spider traps

Basic Theory behind PageRank

九章算法

Dead ends



To\From	WA	WB	WC	WD
WA	0	1/2	0	0
WB	1/3	0	0	1/2
WC	1/3	0	0	1/2
WD	1/3	1/2	0	0

Basic Theory behind PageRank

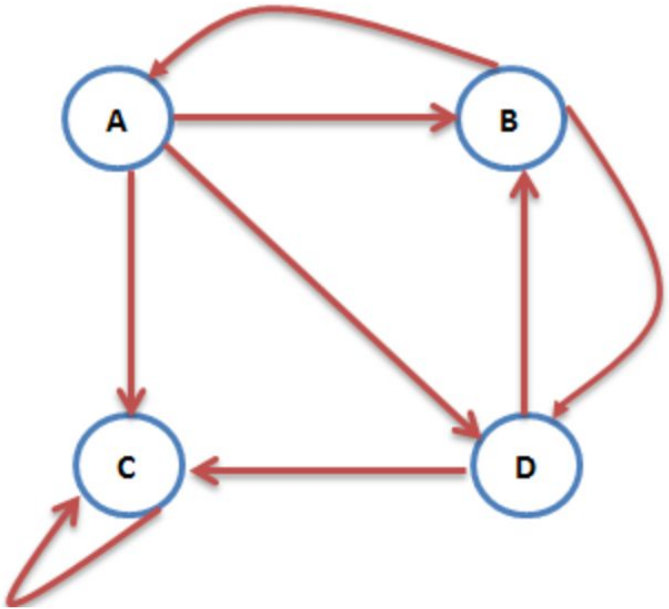
PR(N) matrix will become zeros finally

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 3/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix} \begin{bmatrix} 5/48 \\ 7/48 \\ 7/48 \\ 7/48 \end{bmatrix} \begin{bmatrix} 21/288 \\ 31/288 \\ 31/288 \\ 31/288 \end{bmatrix} \cdots \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Basic Theory behind PageRank

九章算法

Spider traps



To\From	WA	WB	WC	WD
WA	0	1/2	0	0
WB	1/3	0	0	1/2
WC	1/3	0	1	1/2
WD	1/3	1/2	0	0

Basic Theory behind PageRank

九章算法

PR(N) matrix will be dominated by one page

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} \begin{bmatrix} 3/24 \\ 5/24 \\ 11/24 \\ 5/24 \end{bmatrix} \begin{bmatrix} 5/48 \\ 7/48 \\ 29/48 \\ 7/48 \end{bmatrix} \begin{bmatrix} 21/288 \\ 31/288 \\ 205/288 \\ 31/288 \end{bmatrix} \cdots \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Basic Theory behind PageRank

How to solve this problem as a human?

Basic Theory behind PageRank

九章算法

What you will do when you see these two situations?

Basic Theory behind PageRank

九章算法

Close current page and open a new one!

Basic Theory behind PageRank

九章算法

Teleporting

$$PR(N) = PR(N-1) * \text{Transition Matrix}$$

$$PR(N) = (1-\beta)*PR(N-1)*\text{Transition Matrix} + \beta*e$$

Basic Theory behind PageRank

PR(N) = (1-β)*PR(N-1)*Transition Matrix + β*e

β = 1/5

To\From	WA	WB	WC	WD
WA	0	1/2	1	0
WB	1/3	0	0	1/2
WC	1/3	0	0	1/2
WD	1/3	1/2	0	0

×

	PR0
A	1/4
B	1/4
C	1/4
D	1/4

× (1-1/5) +

	PR0
A	1/4
B	1/4
C	1/4
D	1/4

× 1/5

Implement PageRank with MapReduce

Let’s implement on MapReduce!

Implement PageRank with MapReduce

What is the input? Matrix?

To\From	WA	WB	WC	WD
WA	0	1/2	1	0
WB	1/3	0	0	1/2
WC	1/3	0	0	1/2
WD	1/3	1/2	0	0

	PR0
A	1/4
B	1/4
C	1/4
D	1/4

Implement PageRank with MapReduce

No!

- Waste space
- Not easy to insert/delete

Implement PageRank with MapReduce

Input format

```
1 http://www1.hollins.edu/
2 http://www.hollins.edu/
3 http://www1.hollins.edu/Docs/CompTech/Network/webmail_faq.htm
4 http://www1.hollins.edu/Docs/Forms/GetForms.htm
5 http://www1.hollins.edu/Docs/misc/travel.htm
6 http://www1.hollins.edu/Docs/GVCalendar/gvmain.htm
7 http://www1.hollins.edu/docs/events/events.htm
```

```
1 2
8 2
16 2
18 2
20 2
```

Implement PageRank with MapReduce

To simplify your work

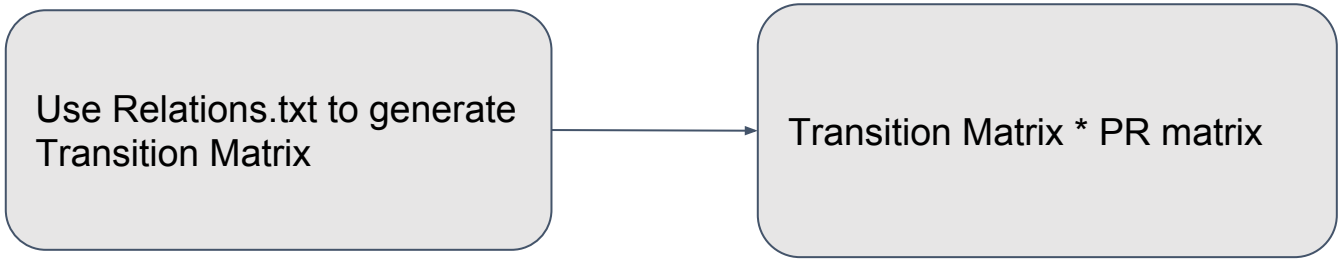
Input1: Relations.txt

```
1   2,7,8,29
2   4,9,10,26
3   1,29,31,26
```

Input2: PR.txt

```
1 1/6012
2 1/6012
3 1/6012
4 1/6012
5 1/6012
```

Implement PageRank with MapReduce



Matrix * Matrix?
You will keep everything in memory!

Implement PageRank with MapReduce

To\From	WA	WB	WC	WD
WA	0	1/2	1	0
WB	1/3	0	0	1/2
WC	1/3	0	0	1/2
WD	1/3	1/2	0	0

	PR0
A	1/4
B	1/4
C	1/4
D	1/4

	PR1
A	9/24
B	5/24
C	5/24
D	5/24

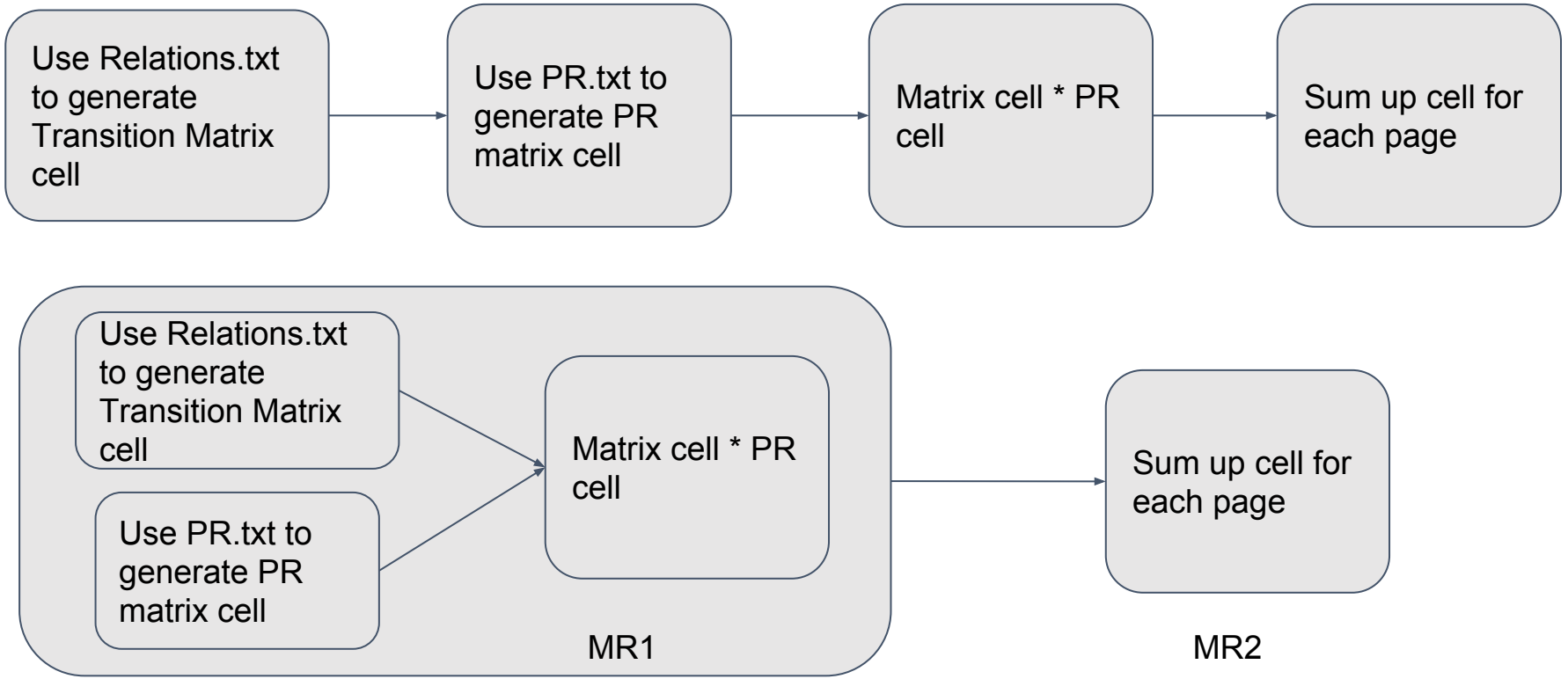
Use Relations.txt to generate Transition Matrix cell

Use PR.txt to generate PR matrix cell

Matrix cell * PR cell

Sum up cell for each page

Implement PageRank with MapReduce



Implement PageRank with MapReduce

九章算法

MR1.Mapper1

Input1: Relations.txt

```
1    2,7,8,29
2    4,9,10,26
3    1,29,31,26
```

Key: from	Value: to = relation
1	2=1/4
1	7=1/4
1	8=1/4
1	29=1/4

To\From	WA	WB	WC	WD
WA	0	1/2	1	0
WB	1/3	0	0	1/2
WC	1/3	0	0	1/2
WD	1/3	1/2	0	0

Implement PageRank with MapReduce

九章算法

MR1.Mapper2

Input2: PR.txt

```
1 1/6012
2 1/6012
3 1/6012
4 1/6012
5 1/6012
```

Key: page	Value: PR
1	1/6012
2	1/6012
3	1/6012
4	1/6012

	PR0
A	1/4
B	1/4
C	1/4
D	1/4

Implement PageRank with MapReduce

MR1.Reducer

Key: from	Value: to = relation
1	2=1/4
1	7=1/4
1	8=1/4
1	29=1/4

Key: page	Value: PR
1	1/6012
2	1/6012
3	1/6012
4	1/6012

Page	List<values>
1	2=¼, 7=¼ , 8=¼ , 29=¼ , 1/6012

Implement PageRank with MapReduce

九章算法

Page	List<values>
1	2=¼, 7=¼ , 8=¼ , 29=¼ , 1/6012

Key: to	Value: subPR
2	¼*1/6012
7	¼*1/6012
….	...

To\From	WA	WB	WC	WD
WA	0	1/2	1	0
WB	1/3	0	0	1/2
WC	1/3	0	0	1/2
WD	1/3	1/2	0	0

	PR0
A	1/4
B	1/4
C	1/4
D	1/4

	PR1
A	9/24
B	5/24
C	5/24
D	5/24

Implement PageRank with MapReduce

MR2.Mapper:

Read file generated from last MR

MR2.Reducer:

Key: to	Value: subPR
2	¼*1/6012
2	⅛ *1/6012
7	¼*1/6012
….	...

sum

Key: to	Value: totalPR
2	¼*1/6012+⅛ *1/6012
7	¼*1/6012
….	...

Overview

- What is PageRank
- Why is PageRank
- Basic theory behind PageRank
- Implement PageRank with MapReduce
- How to do Matrix Multiplication in MapReduce