

HW2 Report

105062581 蕭文逸

- Top Ten vertices sorted by rank: (20 iteration)

1056	0.000632
1054	0.000629
1536	0.000524
171	0.000512
453	0.000496
407	0.000485
263	0.000480
4664	0.000470
261	0.000463
410	0.000462

- Implementation:

基本上分成兩步驟(兩次 map reduce)，第一次是整合 Page Rank 值和 Link 關係，第二次時才會有矩陣運算。

== Phase1 ==

- Mapper1

把 edge 和 vertex 關係改寫成需要的模式。以範例來看，例如：

```
1 2
1 3
2 4
3 1
3 4
3 5
5 1
5 4
```

把它寫成這樣的 pair: $\langle k, v \rangle = \langle 1, (2, 3) \rangle, \langle 2, (4) \rangle, \langle 3, (1,4,5) \rangle$ 等等

同時把 Page Rank 值也 MAP 出去: $\langle k, v \rangle = \langle 0, PR \rangle$

這樣設計，在下一階段時就可以知道 Out Degree 是多少，才能計算 M 的值。

- Reducer1

不做任何事

== Phase2 ==

- Mapper2

上一階段是類似預處理的步驟，這一階段，是要模擬矩陣乘法，把需要用到的值賦予相同 key，方便在 Reducer 計算。

以剛剛第一個 pair 為例，這一階段的 mapper 接收到 $\langle 1, (2, 3) \rangle$ 時，會多分出兩條 pair: $\langle 2, (1, \text{Value}) \rangle$ 和 $\langle 3, (1, \text{Value}) \rangle$ ，其中 Value 值是 1 除以 2(Out-degree)。

然後剛剛的 Pagerank pair $\langle 0, \text{PR} \rangle$ 則會變成

$\langle 1, (1, \text{PR}) \rangle, \langle 1, (2, \text{PR}) \rangle, \langle 1, (3, \text{PR}) \rangle, \langle 1, (4, \text{PR}) \rangle$

這是因為新的 Pager rank 值所用到的東西如下圖：

M					r
0.5					
0.5					

新的 PR 要用到的是 ("從 i 射進來的 $1/\text{degree}$ " * $\text{PR}(i)$)

其中 A 是由 M 矩陣來的，在這邊是先為了把 M 矩陣的值求出，等等在 Reducer 階段會求出 A 並相加。

- Reducer2

$$A = \text{beta} * M + (1 - \text{beta}) * (1/N)_{N \times N}$$

所以這一階段所以 key 相同都 group 在一起。接下來就是要總和算出這一輪 PR。以上個例子繼續來看。

新的 $\text{PR}(1)$ 將會是

$$[\text{beta} * \langle 1, (1, \text{Value}) \rangle + (1 - \text{beta}) * (1/N)] * \langle 1, (1, \text{PR}) \rangle$$

由於 piar 會少(即 $M=0$ 的地方)，所以要寫一些判斷式補足。

$\langle \langle \text{Deadend} \rangle \rangle$

預設是 1(即有 Deadend 手動)，所以算完 PR 後還要在算 S 並重新校正，才是真正的 PR

== Iteration ==

上面兩 PHASE 會 Iteration 20 次。