MDA HW2 Report 108062533 洪翎恩

語言 python / 環境 pyspark / 執行 python3 HW2.py

方法

1. 讀黨並 Map 成('vi' , 'vj')

```
def readfile(line):
    wordlist = line.split("\t")
    maplist = []
    for item in wordlist:
        maplist.append(item)
    return (maplist[0],maplist[1])
```

2. 計算每個 vertex 的 outdegree
   a. Map 成(vi , 1) 表示 vi 的其中一個 out 邊

```
def deg(line):
    wordlist = line.split("\t")
    maplist = []
    for item in wordlist:
        maplist.append(item)
    return (maplist[0],1)
```

   b. Reduce 計算 vi 的 out degree

```
Degree=Input_data.map(deg).reduceByKey(lambda x,y : x+y)
```

3. 初始所有 vertex 的 PR 值 ,並 map 成(vi,1/10876)

```
def init_r(x):
    return(x,1/NUM_NODES)
```

```
#initial PageRank
node_array=[]
for i in range(0,NUM_NODES):
    node_array.append(str(i))

init_PR = sc.parallelize(node_array).map(init_r)
old_PR=init_PR
```

4. 計算 $\frac{r_i}{d_i}$, 先將 old_PR 跟 Degree join 在一起方便直接對 values 計算

```
# Ri/Di
RD= old_PR.join(Degree).mapValues(lambda x: x[0]/x[1])
```

5. 計算 $\sum_{i\to j}\beta\frac{r_i}{d_i}+(1-\beta)\frac{1}{N}$
   a. Graph 跟 RD join 後 map 和 reduce 成 (vj,∑ri/di)
   b. 計算

```
# R'j
pre_Rj=Graph.join(RD).map(lambda x : (x[1][0],x[1][1])).reduceByKey(lambda x,y : x+y)
pre_Rj=pre_Rj.mapValues(lambda x: B*x+((1-B)/NUM_NODES))
```

6. 解決 dead-end, where: $S=\sum_j r_j^{new}$

```
# re-insert the leaked PageRank: Rj=pre_Rj + (1-S)/N
S=pre_Rj.values().sum()
new_PR= pre_Rj.mapValues(lambda x : x+((1-S)/NUM_NODES)).sortByKey(True)
```

7. 計算

```
#compute the diffence between newPR and old one
EP=new_PR.join(old_PR).mapValues(lambda x: abs(x[0]-x[1])).values().sum()
```

8. 持續 4~7 直到收斂為止

輸出

```
1054 0.0006304712006489466
1056 0.0006303121965655017
1536 0.0005257167394985884
171 0.0005132799773146703
453 0.000496782365290902
407 0.0004862900497564125
263 0.0004810005198684766 5
4664 0.00047234479236452163
261 0.0004644379633066404 5
410 0.0004625028462375345
```