Link analysis 生科大四 C54046238 李成學

一、 實做細節

HITS.py、PageRank.py、SimRank.py 這三個檔案分別是用來儲存 Graph 的結構，RunMe.py 會計算出所有的結果，並繪製所有圖片

1. HITS.py

HITSGraph 的 constructor 有二個參數，rows 是要用來建立 graph 的二維 list，connectall 是雙向連接每一個 row 所有的節點，否則只將第一個節點連接到第二個節點。類別包含一個 dict data，data 的 key 是節點的名稱，可以將字串對應到 Node 類別的物件，Node 類別內有二個 list： parents 和 children，內容分別是這個節點的父節點名稱字串以及子節點名稱字串。HITSGraph 類別有另外二個 dict，auth 和 hub，分別將節點名稱的字串對應到該節點的 authority 和 hub 數值所有節點的 authority 和 hub 初始值都是取亂數。每次呼叫 iterate()函數會進行一次迭代，並且回傳迭代前與迭代後的數值差的平方的平均

2. PageRank.py

PageRankGraph 類別與 HITSGraph 大致相同，constructor 多一個參數 df，是在計算 pagerank 的 damping factor。裡面將 auth 和 hub 換成 pr，一樣是一個 dict，將節點名稱的字串對應到該節點的 pagerank 數值，所有節點的 pagerank 初始值都是取亂數。一樣有 iterate()函數來進行迭代，並且回傳迭代前與迭代後的數值差的平方的平均

3. SimRank.py

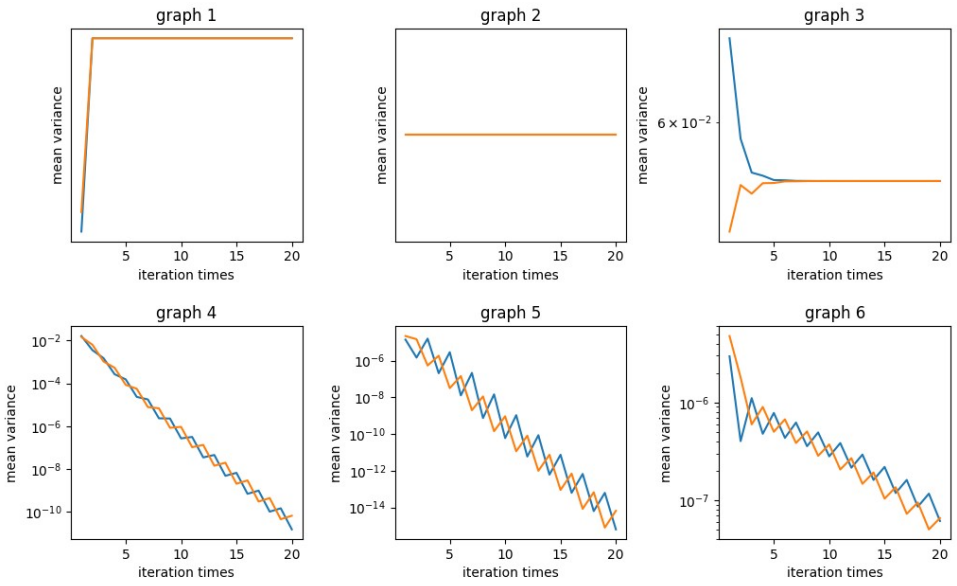
SimRankGraph 類別與前二者相同，constructor 多一個參數 c，是計算 simrank 的 decay factor。裡面沒有 auth、hub 和 pr，有另外一個 dict sim，sim 將由二個節點名稱的字串組成的 tuple 對應到這二個節點的 simrank 數值，tuple 的二個字串都會先以字串順序排序過。一樣有 iterate()函數來進行迭代，並且回傳迭代前與迭代後的數值差的平方的平均

4. RunMe.py

直接執行這個檔案即可，會到 hw3dataset 資料夾讀取所有資料並計算，將計算結果儲存於 result 資料夾內，並且顯示所有繪圖

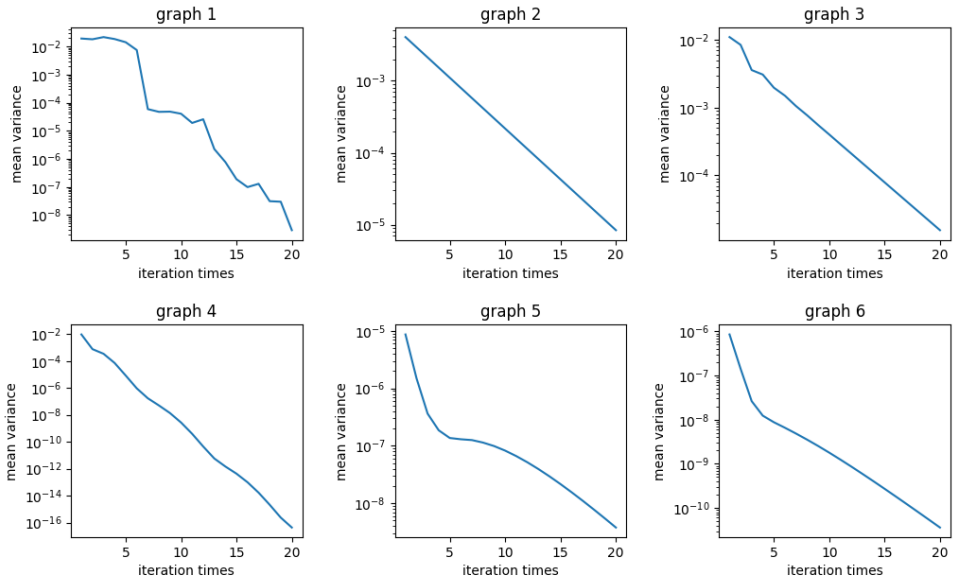
二、 結果

圖一 HITS 的藍線代表 authority，橘線代表 hub。從圖一當中可以發現，graph 1 ~ 3 都不會收斂，而 graph 4 ~ 6 則可以觀察到收斂的現象，而且可以觀察到有鋸齒形狀



圖一、HITS 收斂情形

圖二是 PageRank 的收斂情形，damping factor 取 0.15，所有的 graph 都有收斂



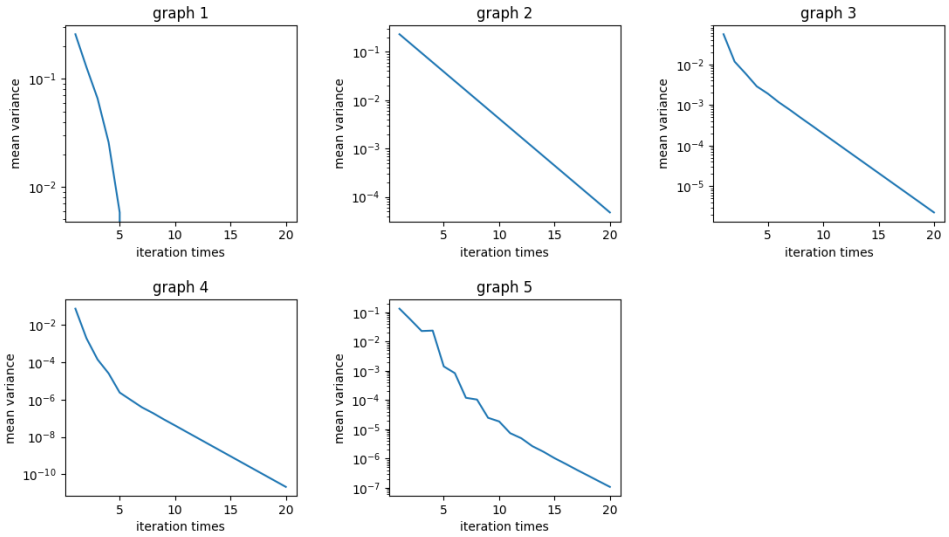
圖二、PageRank 收斂情形

HITS 與 PageRank 的執行結果位於 result/graph_x_hits_pagerank.csv，裡面列出所有節點的 pagerank、authority、hub 數值，若利用這幾個數值將節點排序，會發現即使這幾個數值都是用來量化節點的重要性，但是排序的結果不一定會完全一樣。如果要將 graph 1 ~ 3 的 node 1 增加 pagerank、authority、hub，我將所有的節點全部指向 node 1，並將 node 1 指向全部的節點，如檔案 hw3dataset/graph_x_mod.txt，經過這樣的修改，graph 1 ~ 3 的 node 1 新數值如下表所示，可以看到，node 1 的數值可以藉由這樣的方法來提昇

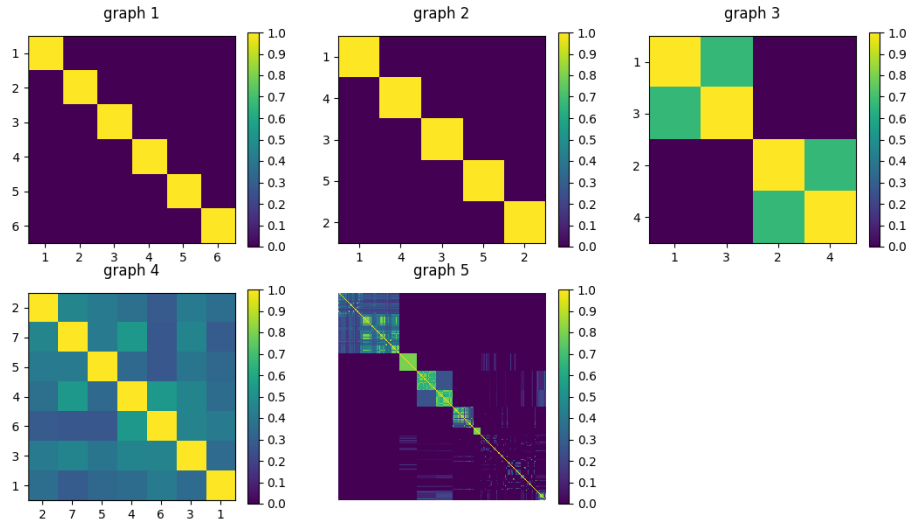
表一、經過修改的 graph 1 ~ 3 的 node 1 數值

	graph 1	graph 2	graph 3
pagerank	0.360	0.374	0.295
authority	0.270	0.289	0.281
hub	0.270	0.289	0.281

圖三是 SimRank 的收斂情形，這五個 graph 都有收斂。圖四是將所有計算出來的 simrank 做 clustering 並且以 heatmap 繪製出來。其中 graph 1 和 2 除了和自己之外，其他所有的節點 simrank 都為 0，graph 3 的(1, 3)和(2, 4)SimRank 比較高，graph 5 的節點可以分成幾群相似度較高的。計算出來得詳細數值位於 result/graph_x_simrank.csv

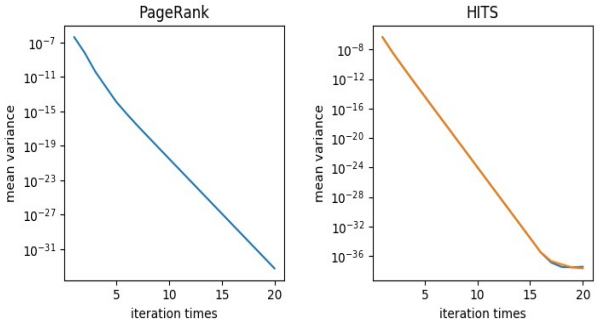


圖三、SimRank 收斂情形

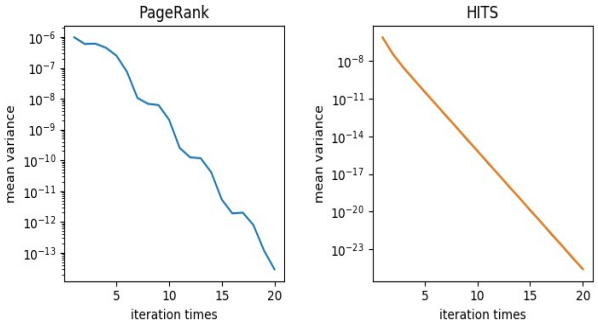


圖四、SimRank 計算結果

hw1 的 dataset 收斂情形如圖五與圖六，無論是 PageRank 或是 HITS 都會收斂。執行結果分別位於 result/hw1dataset_hits_pagerank_bidirected.csv 以及 result/hw1dataset_hits_pagerank_directed.csv



圖五、Homework 1 dataset bidirected 收斂情形



圖六、Homework 1 dataset directed 收斂情形

三、 計算效能分析

下表列出從初始值一直迭代到收斂為止，迭代的次數以及所花費的時間，在這邊收斂定義為 mean variance 小於 10^{-30} 即收斂。在 PageRank 的計算中 graph 4 的節點與邊都比 graph 1 ~ 3 還要多，但是迭代的次數以及執行時間都比較少，可能與 graph 本身的形狀有關。在 graph 5 的計算中，PageRank 的計算時間大於 HITS 的計算時間，在 graph 6 中則相反過來，因此並不能這些測試的資料當中說明何者的效率比較好。graph 5 跑 SimRank 花費的時間高達 80 秒，遠大於 graph 1 ~ 4 跑 SimRank 所花的時間，但是迭代次數並沒有比較多，每次迭代時要從所有所有節點中任意取二個，當節點數增加時，要取的節點對會以平方成長，因此花費的時間會增加很多

表二、迭代次數與計算時間

		graph 1	graph 2	graph 3	graph 4	graph 5	graph 6
HITS	次數				68	54	327
	時間(msec)				0.942	47.022	1066.717
PageRank	次數	77	198	203	40	145	128
	時間(msec)	0.516	1.133	1.056	0.403	86.439	278.782
SimRank	次數	6	152	144	86	139	
	時間(msec)	0.167	3.365	2.168	9.513	80733.251	

四、 討論

在這次測試的資料當中 PageRank 與 SimRank 都會收斂，而 HITS 在 graph 1~3 不會收斂，都在幾個值附近來回震盪，因此計算出來的結果可能是不準確的。有時候 HITS 在收斂過程中會呈現鋸齒狀，這是因為迭代時 hub 的數值會跑到 authority，authority 的數值會跑到 hub，下一次迭代時又換回來。但是並不是一定會發生，我在用 homework 1 的 dataset 跑 HITS 的時候，就沒有鋸齒狀的收斂。如果要讓某個節點的數值提昇，可以藉由指向很多人，以及被很多人指向。在使用這幾種方式分析 graph 時，花費的時間會受到 graph 本身形狀的影響，因此就目前的測資，很難判定說誰的效率比較好。SimRank 因為要兩兩取節點，所以花費的時間也會高出很多。

五、 其他問題

1. Can link analysis algorithms really find the “important” pages from Web?

   不一定，可以利用很多方法來讓自己的數值提昇，像是這次嘗試的將自己指向很多人，以及讓很多人指向自己，可以讓自己的數值提昇。而且者些演算法僅考慮 graph 的結構，沒有考慮人在瀏覽網頁時是不是真的會去點擊超連結

2. What are practical issues when implement these algorithms in a real Web?

因為世界上的網頁太多了，而且每隔網頁的超連結也很多，因此跑這些演算法每次全部迭代一遍會花很多時間

3. What is the effect of “C” parameter in SimRank?

C 就是讓相似性隨著後代逐漸遞減，例如 node 1 有 node 2、node 3 二個子節點，他們分別又有一個子節點 node 4、node 5，$s(1, 1) = 1$，$s(2, 3) = C$，$s(4, 5) = C^2$，越後代相似性會是 C 的越高次方，而逐漸遞減