Data Mining Project 3 Link Analysis

資訊所 P76091048 **仰凱駿**

---

1) Dataset

a) graph_.txt - graph_6.txt :作業提供的graph資料集。

b) graph_7.txt:由 Project 1 的 IBM data，經由transaction_to_graph.py 處理而得的graph 資料集。

---

2) Program(使用colab)

a) DataMining_hw3.ipynb:利用上述dataset 中的資料進行link_analyze, 計算各項評分。

b) main.py :將上述program移植到本地端，在本地端執行。

---

3) Linked Analysis & Result Comparsion and Dusscussion：

a) Linked Analysis:

利用上述dataset中的演算法進行link_analyze,計算各項評分。其中包含 HITS,PageRank, SimRank 算法。

i) HITS

網頁分成兩種, 一為權威型(authority), 一為目錄型(hub), 而我們再依據數學公式去計算authority權重及hub權重, 分別為「連進

來的網頁的hub權重總和」與「連進來的網頁的authority權重總和」,這兩者皆為越大越好。

ii) PageRank

PageRank為網頁被看到的可能性, 每個網頁都有自己的PageRank,該PageRank來自於所有連結到該網站的網站其PageRank / 該網站的連結數總和, 如以下數學公式：

$$PR(P_i) = \frac{(d)}{n} + (1-d) \times \sum_{l_{j,j} \in E} PR(P_j) / \text{Outdegree}(P_j)$$

D(damping factor)=0.1~0.15
n=|page set|

iii) SimRank

SimRank為運算兩node之間關聯性的算法，數學表現式如下：

(1) 當$a = b$時，$s(a,b) = 1$.

(2) 當$\mathcal{I}(a) = \emptyset$或者$\mathcal{I}(b) = \emptyset$時，$s(a,b) = 0$.

(3) 其他情況下，

$$s(a,b) = \frac{C}{|\mathcal{I}(a)|\,|\mathcal{I}(b)|} \sum_{i=1}^{|\mathcal{I}(a)|} \sum_{j=1}^{|\mathcal{I}(b)|} s(\mathcal{I}_i(a), \mathcal{I}_j(b))$$

其中，$0 < C < 1$是一個阻尼係數.

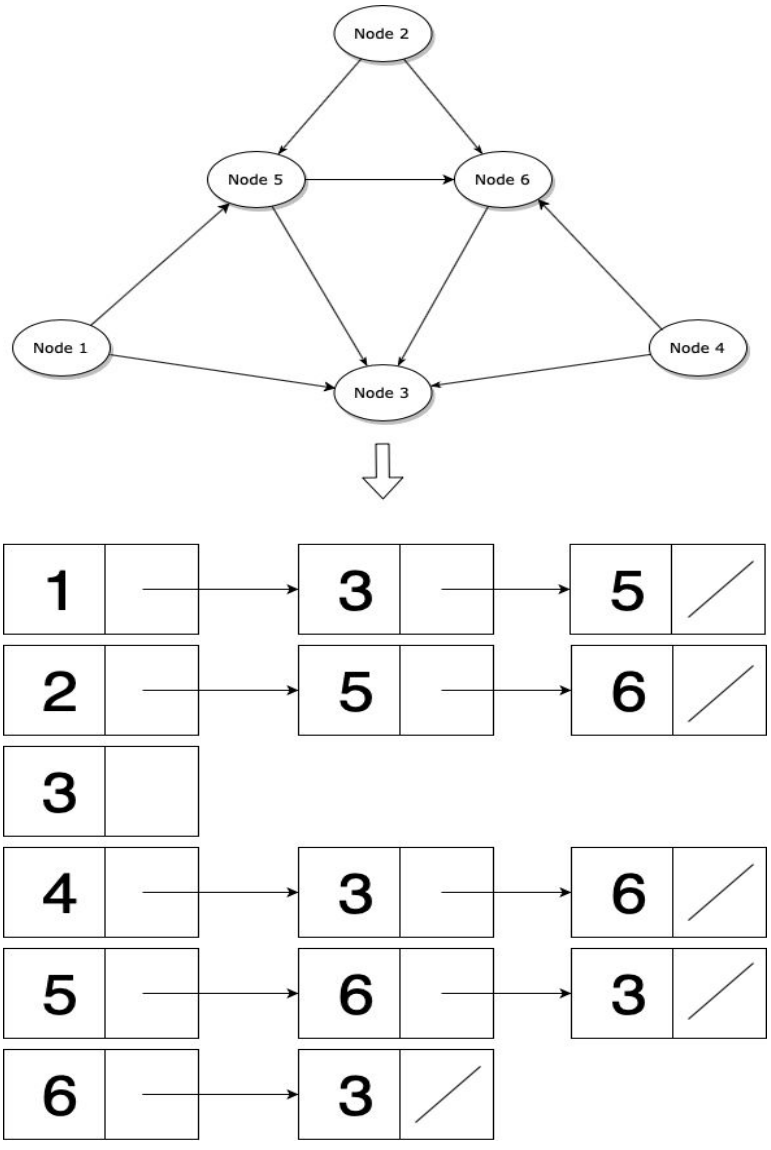
其中亦可以用矩陣形式來進行計算，數學表現式：

$$\begin{cases} \mathbf{S}^{(0)} = \mathbf{I}_n \\ \mathbf{S}^{(k+1)} = \left(c \cdot \mathbf{Q} \cdot \mathbf{S}^{(k)} \cdot \mathbf{Q}^T \right) \bigvee \mathbf{I}_n \quad (\forall k = 0, 1, \cdots) \end{cases}$$

$$[\mathbf{Q}]_{i,j} = \begin{cases} 1/|\mathcal{I}(i)|, & \text{if } \exists \text{ edge } (j \to i) \in \mathcal{E}; \\ 0, & \text{otherwise.} \end{cases}$$

b) Complexity Analyize

實作的部分我是藉由Adjacency list來進行實作,因為當陣列過於稀疏時會導致空間的浪費以及提高Scan的次數,造成複雜度提高,故選擇使用Adjacency list來進行實作,示意圖:



i) HITS：

計算hub與authority時，必須要掃過圖形表示法中的每一比資料來檢查，各個點的in-neighbor以及out-neighbor，在使用adj_matrix 時為O(V^2)的時間複雜度，而在adj_list時O(V+E)的時間複雜度。

ii) PageRank：

PageRank的計算方法與上雷同，均為adj_matrix 時為O(V^2)的時間複雜度，而在adj_list時O(V+E)的時間複雜度。

iii) SimRank：

一般的Straightforward iterative的時間複雜度是O(K*V^4)，若以matrix form來進行實作，共需要做矩陣乘法，時間複雜度為O(V^3)，空間複雜度為O(V^2)，故SimRank我是以matrix form來進行實作。(K is the numbers of iterations)

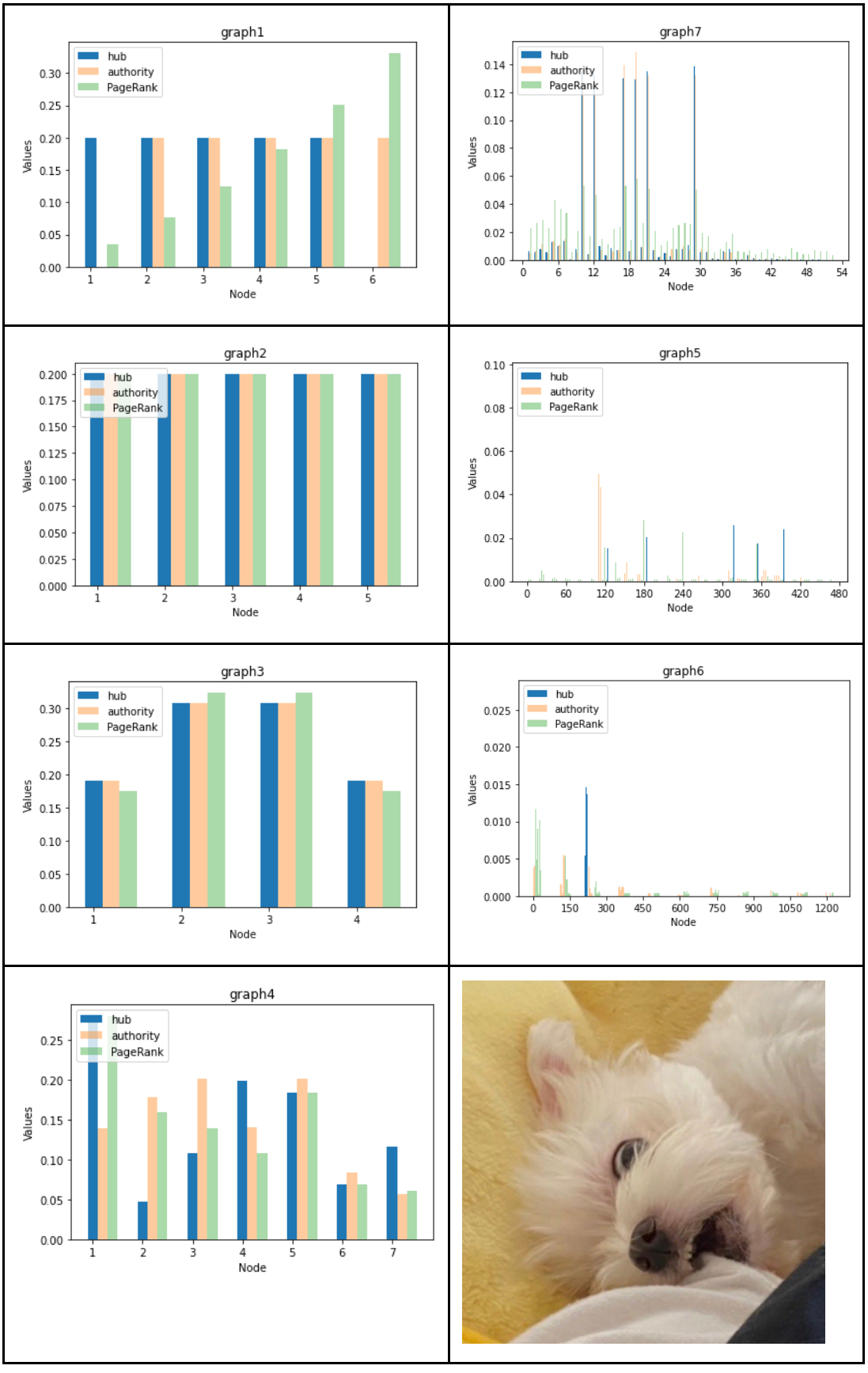
iv) Conclusion：

Time Comp.	adj_list	adj_matrix
HITS	O(V+E)	O(V^2)
PageRank	O(V+E)	O(V^2)

SimRank	Time Comp.	Space Comp.
general	O(K*V^4)	O(V^2)
matrix form	O(K*V^3)	O(V^2)

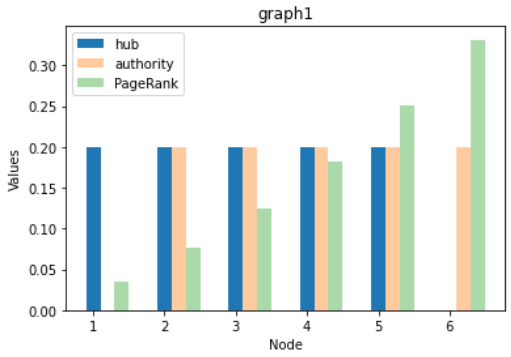
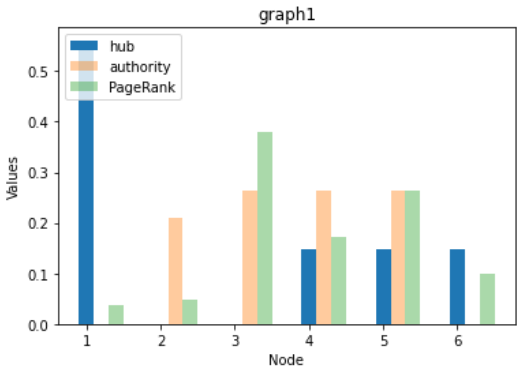
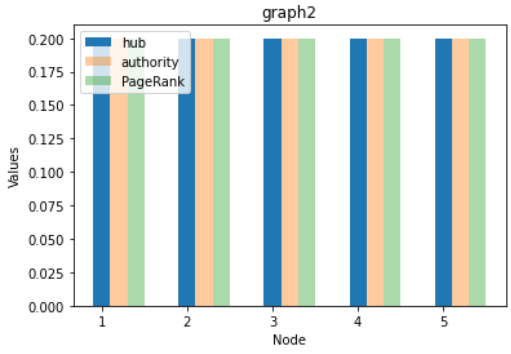
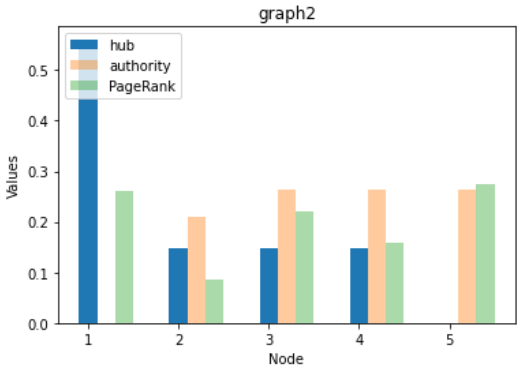
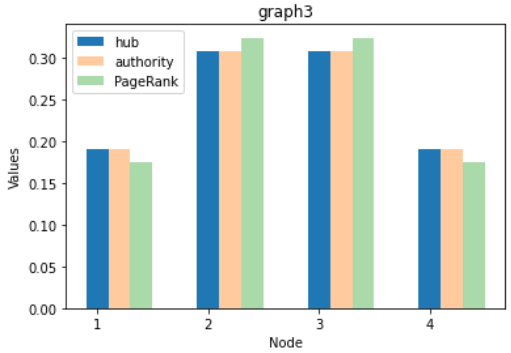
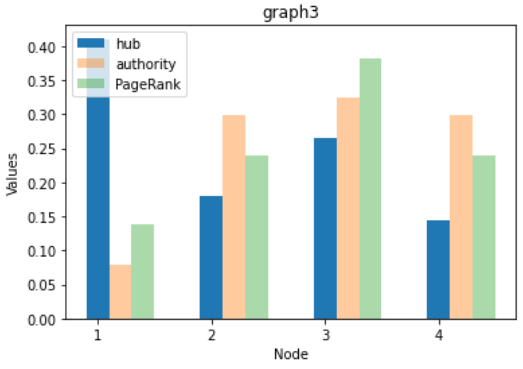
c) Result Comparsion

i) Result of graph 1~7

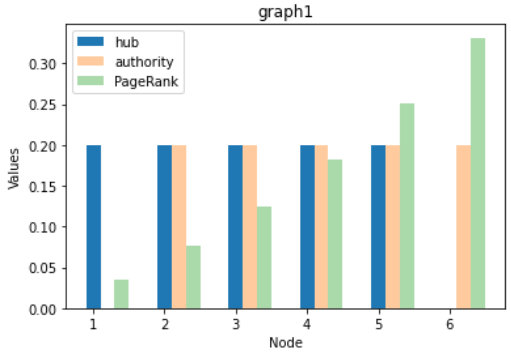
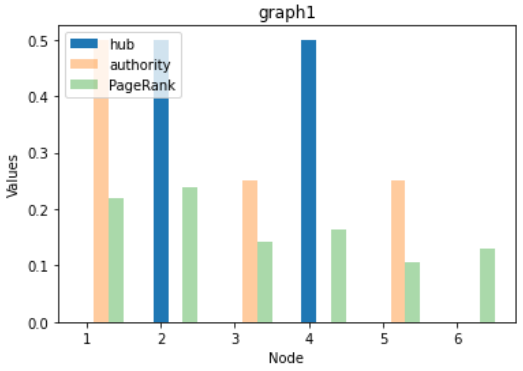
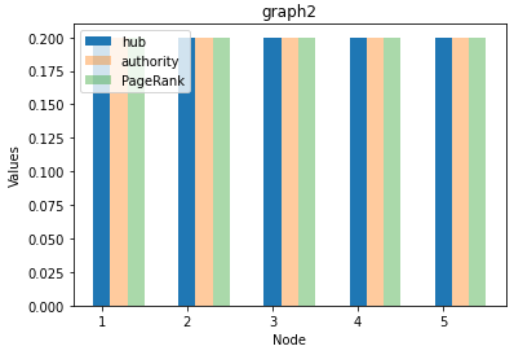
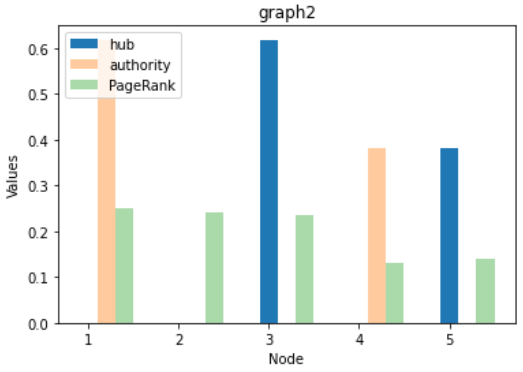
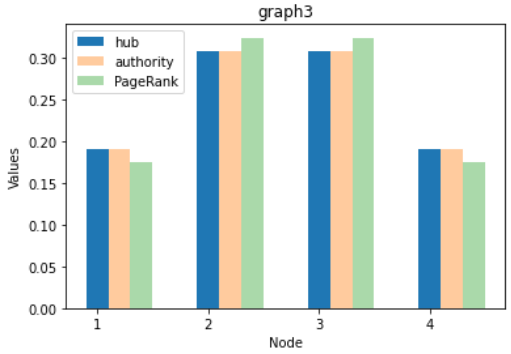
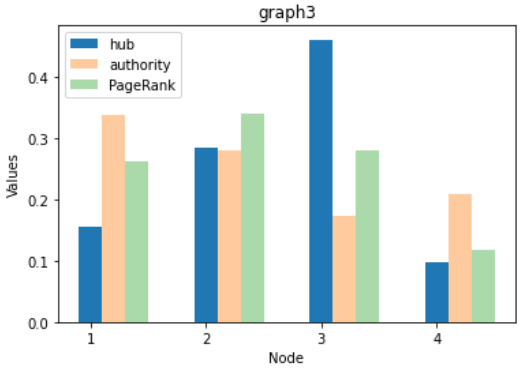


ii) 找到 Graph 1 到 3，增加 Node 1 的 hub, authority 及 PageRank 的方法 根據上面各圖，可以得知該 node 連接的 node 數越多，其 hub、authority 及 PageRank 值容易較高，以下為各實驗。

(1) 增加 Node 1 的Child 數量

原圖	增加link
	
	
	

(2) 增加 Node 1 的Parent 數量

原圖	增加link
	
	
	

d) Conclusion of Result

i) 若 Graph 為單向，如 Graph1，則有 child 或者 parents 的 node 其 hub 與 authority皆相同，而 PageRank 則隨著 parents 數量增加而遞增。

ii) 若 Graph 為單向有環，如 Graph2，同樣地有 child 或者 parents 的 node 其 hub 與authority 皆相同，而 PageRank 會隨著 node 數增加而逐漸一致。

iii) 若 Graph 為多環，如 Graph3、Graph4，則可以看出連接數目越多 node，有較高的 hub、authority 及 PageRank 值。

iv) 而 Graph5、Graph6，node 數及 edge 數較多，根據圖來看大部分值都較低，而有少數 幾個 node 其各項評分都較高，可能是這些 node 所連接的 node 都集中在少數的原 因。

v) 至於從 IBMdata 產生出的 Graph7，可以看出大部分的 hub 與 authority 相同，可能 是大部分所連接的 node 沒有重複，幾乎形成單向 Grapg 的原因，且可能是 IBMdata 產生規則的關係，幾乎每幾百個 node 就有固定重複的圖形。

vi) SimRank 方面，若有環或者單向等等則 SimRank 即為 0，因此僅有 Graph 4 的 SimRank 不等於 0。

vii) 並且，根據將同 Graph 中的變動同一個 node 連接的 child 或者 parents 數，會影響 到 hub、authority 及 PageRank 值，若一 node 其 child 數較多，則該 node 的 authority 較高，而若一 node 其 parents 數較多，則該 node 的 hub、PageRank 值較 高。

4) Concluson

這次Project實作了三種經典的Rank Algorithm，實驗中發現有 cycle 對於數值影響很高，也發現對於 HITS 而言，只要增加幾條link就會對結果產生不小的影響。 而 PageRank 算法雖然較穩定，但其數值 與 Authority 仍具有一定的關聯，且在link較少的情況下是較推薦用adj_list的(因為時間複雜度接近O(V)。至於 SimRank 雖然能有效比對兩 node 之相似性，但其運算成

本較高，光是實驗中所用之 Graph 其運算時間就使用了數分鐘，即使使用Matrix form依然需要消耗很多時間O(V^3)，實務上不可能採用。

---

5) Question & Discussion

a) 若 Graph 為單向，如 Graph1，則有 child 或者 parents 的 node 其 hub 與 authority 皆相同，而 PageRank 則隨著 parents 數量增加而遞增。

b) 可以，link analysis algorithms 能夠有效計算出該網站的權重並且能夠給出非常具體的指標，實際上也有多的商業搜尋引擎已經實際引用了這個算法。

c) link非常多的情況下Adjacency list的時間複雜度與Adjacency matrix差不多，但若是在link較少的網站，則推薦使用Adjacency list來實作。

d) 而 Graph5、Graph6，node 數及 edge 數較多，根據圖來看大部分值都較低，而有少數 幾個 node 其各項評分都較高，可能是這些 node 所連接的 node 都集中在少數的原 因。

e) 至於從 IBMdata 產生出的 Graph7，可以看出大部分的 hub 與 authority 相同，可能是大部分所連接的 node 沒有重複，幾乎形成單向 Grapg 的原因，且可能是 IBMdata 產生規則的關係，幾乎每幾百個 node 就有固定重複的圖形。

f) 當C越小SimRank會變得越小 Decay越快，C越大則Decay越慢，C越小很容易讓關係越近的關係度越大，關心越疏遠的關係度越小，擴大差距。

---

6) References

[1] Weiren Yu. A Space and Time Efficient Algorithm for SimRank Computation

[2] Weiren Yu, SimRank*: effective and scalable pairwise similarity search based on graph topology