

- 作業目的：

- 實做 HITS、PageRank、SimRank 演算法，並對其結果進行討論。

- 實做細節：

- HITS：

根據虛擬碼將指向同個結點的 hub 值加總計算為其結點 authority 值，將其結點指到的 authority 值加總為其結點的 hub 值。在每個 iteration 時都將其值標準化，使其在 0~1 之間，重複這些步驟值到其值收斂。輸出的部分為了方便閱讀，將其輸出為小數點後 4 位。

- PageRank：

將每個結點的 pagerank 預設值為一，並進行計算。每個 iteration 將每個結點的 pagerank 值重新更新為連進此結點的結點的 pagerank / vertex，一直重覆此步驟值到收斂。輸出的部分為了方便閱讀，將其輸出為小數點後 4 位。其 damp factor = 0.15。

- SimRank：

假如 S 函式相比的結點為本身，將其 S 值設為一。首先先計算出每個結點的 inlink 數，為的是要將每輪計算 S 值標準化為 0~1 之間，再來將所有結點的 S 值預設為一，根據每個結點有相關的 inlink 來進行每 S 值的更新，其中 C 為一阻尼系數。一直重複上數更新步驟值到收斂。

- Graph_trans:

IBM-data-generator 產生的資料，我採用的資料共有 541 筆交易，10 種商品。

- Graph_rules:

為一香菇有毒或者可食用和各自的屬性相關，共有 22 個屬性及一個 class。

- 結果分析與討論：

- 在 HITS 演算法的部分，從圖一和圖二可以很明顯的發現假如沒有點連入其節點，其 Authority 值就會是 0；假如其節點沒有連出去其他點，其 hub 值會為 0。而其演算法因為要計算 authority 和 hub 值，在節點數或是 link 數較多時比起 Pagerank 會花較多時間在計算上。

<pre>graph: 1 Authority: ['0.0000', '0.2000', '0.2000', '0.2000', '0.2000', '0.2000'] Hub: ['0.2000', '0.2000', '0.2000', '0.2000', '0.2000', '0.0000']</pre>	<pre>graph: 2 Authority: ['0.2000', '0.2000', '0.2000', '0.2000', '0.2000'] Hub: ['0.2000', '0.2000', '0.2000', '0.2000', '0.2000']</pre>
---	---

- 在 PageRank 演算法部分，調整其 damp factor 會造成不一樣的計算結果，當 damp factor 越高時，其所有結點的 PageRank 會越趨於平均，

而當 damp factor 為 0 時就會出現某些點只有被連入但沒有連出，變成所謂的「黑洞」。而在計算的效率方面，因為只需要計算一個 pagerank 的值，比起 HITS 較有效率。

隨著 damp factor 越高，值越平均。

```
graph: 1
['0.0313', '0.0700', '0.1169', '0.1727', '0.2540', '0.3551']
Highest PageRank node and its value:
Node: 6 PageRank Value: 0.3551

graph: 1
['0.0379', '0.0808', '0.1287', '0.1816', '0.2478', '0.3231']
Highest PageRank node and its value:
Node: 6 PageRank Value: 0.3231

graph: 1
['0.0662', '0.1187', '0.1604', '0.1935', '0.2200', '0.2411']
Highest PageRank node and its value:
Node: 6 PageRank Value: 0.2411
```

- 在 SimRank 方面因為計算的是兩個點之間的相似度，所以和本身的相似度為一，與 HITS 和 Pagerank 計算的值較為不同。圖一和圖二的 SimRank 除了本身為 1 以外其餘都是 0，因為每個點都是單向連接，並沒有有所為「相似節點」。而因為做了標準化的關係，除了自己這個節點以外的直行加總為 1。效率方面我認為不能和 HITS 或 PageRank 相比，因為計算的東西不一樣。
- 在 transaction data 的圖中，我採用的資料總共有 541 筆交易，10 種商品。在 HITS 演算法出來的 Authority 值有數字的只有前 10 個，因為根據後面的 transaction 做出來的節點並沒有被連入，所以其 authority 值都為 0，authority 值我認為是在全部的 transaction 中被選出來越多次的值會越高。而我認為 Hub 值沒有意義，因為這並不是一個網路的圖，只是一堆的 transaction 連到那 10 種商品而已。在 PageRank 中的解釋也和 HITS 的 Authority 相似，前十個節點有明顯不同的值，而且被 transaction 選出越多次的商品其 PageRank 值越高。

```
graph: 7
['0.0017', '0.0006', '0.0953', '0.1159', '0.0966', '0.0430', '0.1993', '0.2030', '0.0852', '0.0003',
Highest PageRank node and its value:
Node: 8 PageRank Value: 0.2030
```

- 在 association rule 的圖中總共有 22 種屬性和一個 class 我根據他們做出來的 association graph 執行 HITS 和 PageRank。由於有些節點根本沒有被連到，所以其 Authority、Hub 都為 0。而 PageRank 有 damp factor 所以沒有任何一點為 0。出來的結果和 transaction data 相似，因為節點 2 和其它的 attribute 有明顯的關聯，所以被連入和連出較多，造成其 Authority、Hub、PageRank 都比較高。可以證明 attribute 2 為一個和其它節點重要的關聯節點。

```
graph: 8
Authority:
['0.1667', '0.6611', '0.0235', '0.0000', '0.0937', '0.3307', '0.4946', '0.3307',
, '0.0000', '0.0937']
Hub:
['0.2818', '0.5024', '0.0000', '0.0000', '0.0000', '0.2818', '0.2196', '0.2196',
, '0.0000', '0.0000']
```

```
graph: 8
['0.0681', '0.2960', '0.0241', '0.0069', '0.0294', '0.0964', '0.1098', '0.0815',
, '0.0069', '0.0395']
Highest PageRank node and its value:
Node: 2 PageRank Value: 0.2960
```

● 如何將圖中的 node1 的 hub、authority、Pagerank 值提升？

■ Hub 值：

實驗的結果發現，要讓 hub 值提升有兩種方法。

1. 將 node 1 的 outlink 變多，多連結到其它 node
2. 將其它 node 之間沒有連到 node1 的 link 刪除

在實驗時也發現了假如刪除和 node1 有連接的 link 的話，並不會對 hub 值的增加有幫助，通常不會改變其 hub 值。

```
graph: 1
Authority:
['0.0000', '0.5256', '0.8506', '0.0096', '0.0096', '0.0096']
Hub:
['0.8506', '0.5257', '0.0059', '0.0059', '0.0059', '0.0000']

C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 1
Authority:
['0.0000', '0.7057', '0.7057', '0.0000', '0.0441', '0.0441']
Hub:
['0.9990', '0.0000', '0.0000', '0.0312', '0.0312', '0.0000']

C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 1
Authority:
['0.0000', '0.5773', '0.5773', '0.5773', '0.0071', '0.0071']
Hub:
['1.0000', '0.0000', '0.0000', '0.0041', '0.0041', '0.0000']

C:\Users\wei\Desktop\碩一\資料探勘\project3>
```

```
graph: 2
Authority:
['0.0000', '0.4597', '0.6280', '0.6280', '0.0017']
Hub:
['0.8881', '0.3251', '0.3251', '0.0009', '0.0000']

C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 2
Authority:
['0.0017', '0.4597', '0.6280', '0.6280', '0.0000']
Hub:
['0.8881', '0.3251', '0.3251', '0.0000', '0.0009']

C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 2
Authority:
['0.0030', '0.4999', '0.7072', '0.4999', '0.0000']
Hub:
['0.9239', '0.3827', '0.0000', '0.0000', '0.0016']

C:\Users\wei\Desktop\碩一\資料探勘\project3>
```

```
graph: 3
Authority:
['0.0752', '0.6998', '0.1217', '0.6998']
Hub:
['0.7023', '0.0988', '0.7023', '0.0610']

C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 3
Authority:
['0.0096', '0.5256', '0.0096', '0.8506']
Hub:
['0.8506', '0.0059', '0.5257', '0.0059']

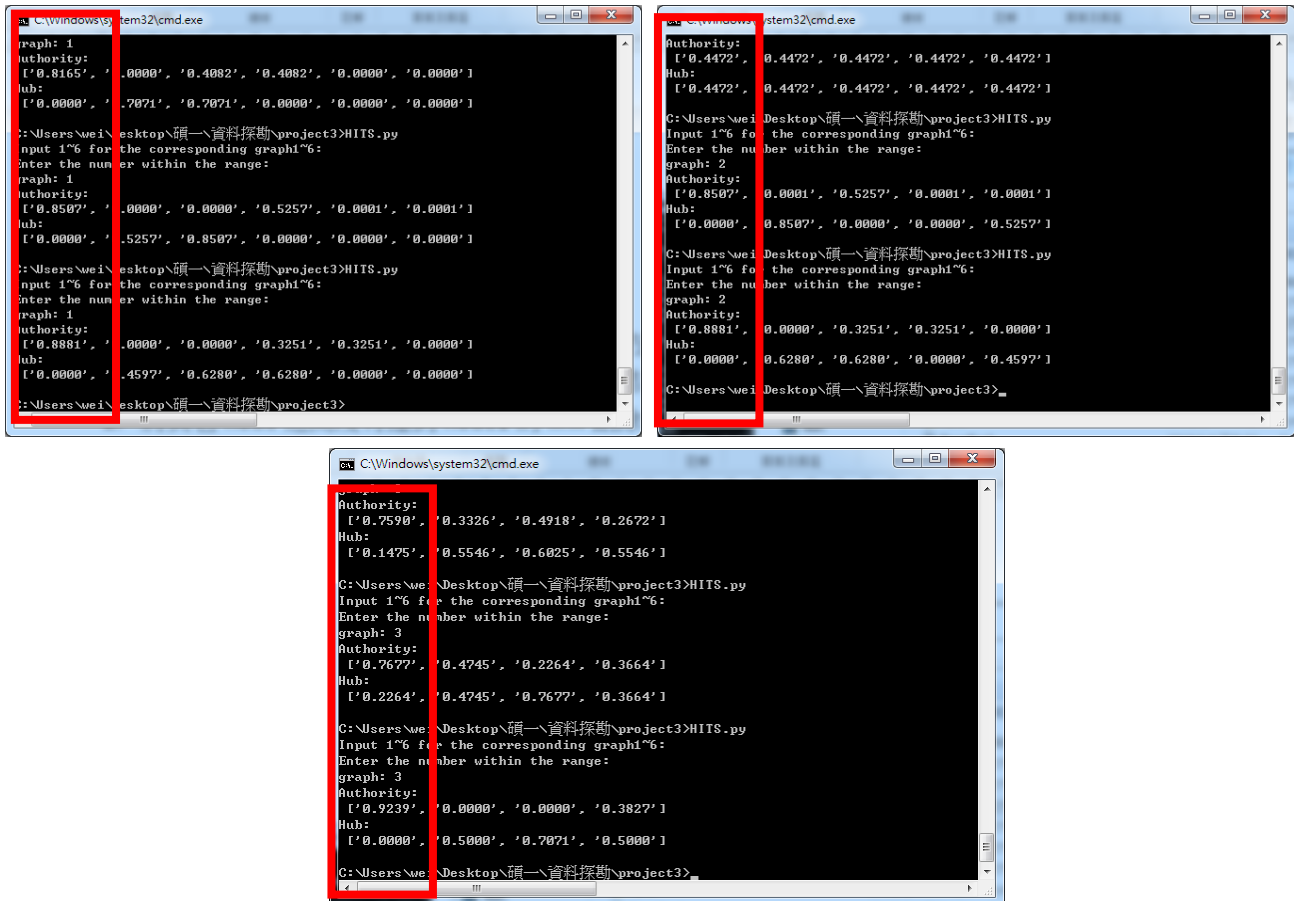
C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 3
Authority:
['0.0017', '0.4597', '0.6280', '0.6280']
Hub:
['0.8881', '0.0009', '0.3251', '0.3251']

C:\Users\wei\Desktop\碩一\資料探勘\project3>
```

■ Authority 值：

實驗的結果發現，要讓 authority 值提升有兩種方法。

1. 將 node 1 的 inlink 變多，多被其它 node 連結
2. 將其它 node 之間沒有連到 node1 的 link 刪除



```
graph: 1
Authority:
['0.8165', '0.0000', '0.4082', '0.4082', '0.0000', '0.0000']
Hub:
['0.0000', '0.7071', '0.7071', '0.0000', '0.0000', '0.0000']

C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1 for the corresponding graph: 1
Enter the number within the range:
graph: 1
Authority:
['0.8507', '0.0000', '0.0000', '0.5257', '0.0001', '0.0001']
Hub:
['0.0000', '0.5257', '0.8507', '0.0000', '0.0000', '0.0000']

C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1 for the corresponding graph: 1
Enter the number within the range:
graph: 1
Authority:
['0.8881', '0.0000', '0.0000', '0.3251', '0.3251', '0.0000']
Hub:
['0.0000', '0.4597', '0.6280', '0.6280', '0.0000', '0.0000']

C:\Users\wei\Desktop\碩一\資料探勘\project3>

graph: 2
Authority:
['0.4472', '0.4472', '0.4472', '0.4472', '0.4472', '0.4472']
Hub:
['0.4472', '0.4472', '0.4472', '0.4472', '0.4472', '0.4472']

C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1 for the corresponding graph: 1
Enter the number within the range:
graph: 2
Authority:
['0.8507', '0.0001', '0.5257', '0.0001', '0.0001', '0.0001']
Hub:
['0.0000', '0.8507', '0.0000', '0.0000', '0.5257', '0.5257']

C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1 for the corresponding graph: 1
Enter the number within the range:
graph: 2
Authority:
['0.8881', '0.0000', '0.3251', '0.3251', '0.0000', '0.0000']
Hub:
['0.0000', '0.6280', '0.6280', '0.0000', '0.4597', '0.4597']

C:\Users\wei\Desktop\碩一\資料探勘\project3>

graph: 3
Authority:
['0.7590', '0.3326', '0.4918', '0.2672']
Hub:
['0.1475', '0.5546', '0.6025', '0.5546']

C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1 for the corresponding graph: 1
Enter the number within the range:
graph: 3
Authority:
['0.7677', '0.4745', '0.2264', '0.3664']
Hub:
['0.2264', '0.4745', '0.7677', '0.3664']

C:\Users\wei\Desktop\碩一\資料探勘\project3>HITS.py
Input 1 for the corresponding graph: 1
Enter the number within the range:
graph: 3
Authority:
['0.9239', '0.0000', '0.0000', '0.3827']
Hub:
['0.0000', '0.5000', '0.7071', '0.5000']

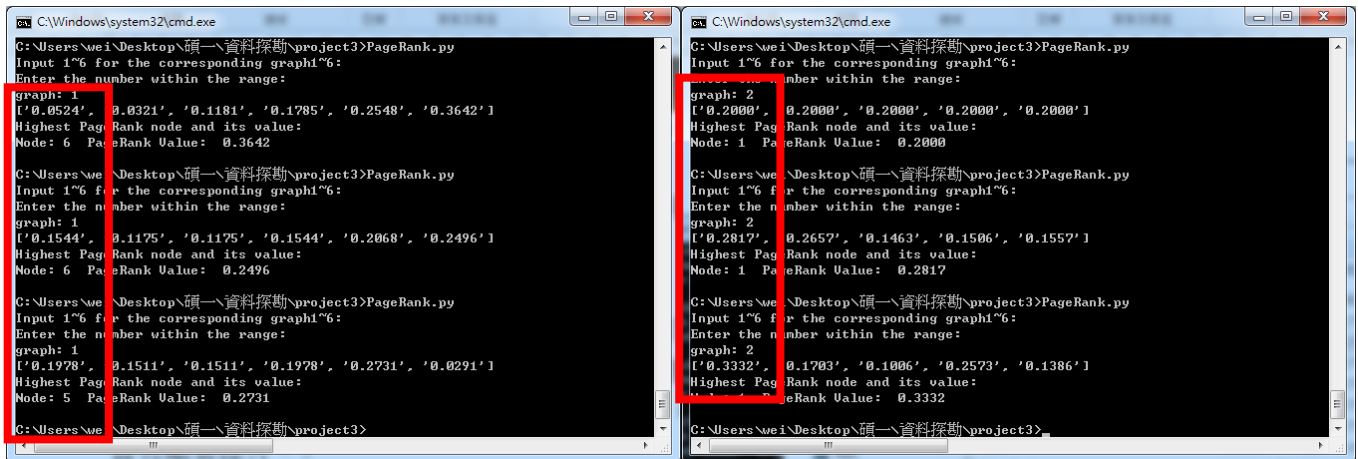
C:\Users\wei\Desktop\碩一\資料探勘\project3>
```

■ PageRank 值：

實驗的結果發現，要讓 pagerank 值提升的方法有：

1. 讓 node1 連到的 node 和 node1 互相連結
2. 刪除沒有跟 node1 連結的 link

測試的過程中發現光是有多條 link 出去或是只有多條 link 進來雖然會提升 Pagerank，但是並沒有顯著提升，在相互連結之後就有顯著的提升。而只要刪除和 node1 沒有相連的 link 之後，node1 的 Pagerank 也會有顯著提升。



```
C:\Windows\system32\cmd.exe
C:\Users\wei\Desktop\碩一\資料探勘\project3>PageRank.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 1
['0.0524', '0.0321', '0.1181', '0.1785', '0.2548', '0.3642']
Highest PageRank node and its value:
Node: 6 PageRank Value: 0.3642

C:\Users\wei\Desktop\碩一\資料探勘\project3>PageRank.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 1
['0.1544', '0.1175', '0.1175', '0.1544', '0.2068', '0.2496']
Highest PageRank node and its value:
Node: 6 PageRank Value: 0.2496

C:\Users\wei\Desktop\碩一\資料探勘\project3>PageRank.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 1
['0.1978', '0.1511', '0.1511', '0.1978', '0.2731', '0.0291']
Highest PageRank node and its value:
Node: 5 PageRank Value: 0.2731

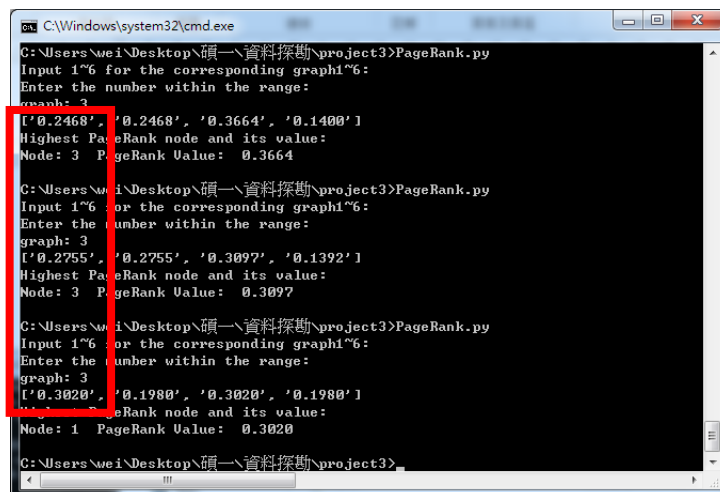
C:\Users\wei\Desktop\碩一\資料探勘\project3>
```

```
C:\Windows\system32\cmd.exe
C:\Users\wei\Desktop\碩一\資料探勘\project3>PageRank.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 2
['0.2000', '0.2000', '0.2000', '0.2000', '0.2000', '0.2000']
Highest PageRank node and its value:
Node: 1 PageRank Value: 0.2000

C:\Users\wei\Desktop\碩一\資料探勘\project3>PageRank.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 2
['0.2817', '0.2657', '0.1463', '0.1506', '0.1557', '0.1557']
Highest PageRank node and its value:
Node: 1 PageRank Value: 0.2817

C:\Users\wei\Desktop\碩一\資料探勘\project3>PageRank.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 2
['0.3332', '0.1703', '0.1006', '0.2573', '0.1386', '0.1386']
Highest PageRank node and its value:
Node: 1 PageRank Value: 0.3332

C:\Users\wei\Desktop\碩一\資料探勘\project3>
```



```
C:\Windows\system32\cmd.exe
C:\Users\wei\Desktop\碩一\資料探勘\project3>PageRank.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 3
['0.2468', '0.2468', '0.3664', '0.1400']
Highest PageRank node and its value:
Node: 3 PageRank Value: 0.3664

C:\Users\wei\Desktop\碩一\資料探勘\project3>PageRank.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 3
['0.2755', '0.2755', '0.3097', '0.1392']
Highest PageRank node and its value:
Node: 3 PageRank Value: 0.3097

C:\Users\wei\Desktop\碩一\資料探勘\project3>PageRank.py
Input 1~6 for the corresponding graph1~6:
Enter the number within the range:
graph: 3
['0.3020', '0.1980', '0.3020', '0.1980']
Highest PageRank node and its value:
Node: 1 PageRank Value: 0.3020

C:\Users\wei\Desktop\碩一\資料探勘\project3>
```

● 結論與相關討論：

我認為在這幾個演算法中，可以最直接反應出一個網頁的重要性目前就屬 PageRank。因為 PageRank 很直接的把每個網頁的分數根據使用者瀏覽到下一個網頁的機率加起來，變成下一個網頁的分數。近年來一些衝高 PageRank 的農場網站開始變多，而使得買賣點擊的連結變成一筆交易，讓 PageRank 不再是那麼具有網頁的代表性，因為可能有些網站的分數是被相互連結所提高的，也迫使 Google 聲稱已經不會再更新 PageRank，但是 PageRank 實為一個網頁重要性的指標。

SimRank 中的 C 參數越大的話最後每個節點的 SimRank 值會越大，相當於在計算時的分數權重。在計算中它是在分子，所以當 C 值越大時，每個節點之間的分數就會被估算得越大。

這些演算法的限制在於必須要知道每個網站的對外連結，還有決定要觀察的目標結點數。因為現在網頁的數量實在多如牛毛，大大小小的網頁沒辦法全部都納入計算，因此要謹慎決定要研究觀察的目標。還有一些農場網頁的問題造成 PageRank 的分數也會被買賣而上升，因此其分數表現出來的可靠度變得有待商榷。

我認為現在科技發展速度太快，許多創新、前所未見的科技日新月異，再加上規模的成長也比以往快上許多，許多舊的技術或是演算法不可能完全延用到新的技術上，因此必須做出改變。舊的技術必須要適應新的環境，或者是根據新的環境直接發展新技術我認為都是必要的，尤其是資訊相關產業的人都要自己保持警覺，隨時更新知識讓自己能夠不被淘汰。

在 Google 停止更新 PageRank 之後，我認為他們早就在著手研究網頁相關的新演算法才會下這樣的決定。在研究中我認為能越早洞悉到未來可能的發展趨勢並著手去做越有利，就算是一個簡單的概念，在還沒有人對這主題做出這種應用之前，都算是創新。就算是沒辦法想到新點子，只要能夠在現有的問題之下找出新的方法或解決方案，就是對研究領域的一種貢獻了。