Implementation Detail

1. HITS的實作過程及細節如下：
2. 建立一個矩陣A，將有指向關係的元素設為1。

例如：(1, 2)則將A[1][2]設為1。

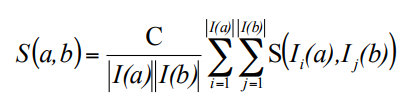
1. 將所有page的hub值以及auth值都先設定為1。
2. 計算新的hub值時，利用矩陣乘法將A的轉置矩陣及原本的hub相乘，再做normalize。
3. 計算新的auth值時，利用矩陣乘法將A及原本的auth相乘，再做normalize。
4. 持續這樣的計算直到新的hub與舊的hub誤差小於特定值，或是新的auth與舊的auth誤差小於特定值，就結束運算。
5. PageRank的實作過程及細節如下：
6. 建立一個矩陣A，將有指向關係的元素設為1。

例如：(1, 2)則將A[1][2]設為1。

1. 在pagerank中對於每個node都先給定值1/n。
2. 給定alpha=0.85，也就是damping factor=0.15。
3. 根據矩陣A，建立其轉移矩陣，利用矩陣乘法將A及原本的pagerank相乘，得到的矩陣\*alpha再加上一個每個元素值皆為

(1-alpha)/n的n為向量，得到新的pagerank。

1. 持續這樣的計算直到新的pagerank與舊的pagerank誤差小於特定值，就結束運算。
2. SimRank：



依照上圖公式計算每個pair(u->v)的sim值。

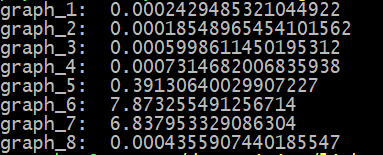
I(a): a的所有parents

Result analysis and discussion

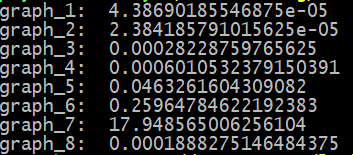
1. HITS演算法及PageRank演算法實作code在hits\_pagerank.py中，執行這支程式，會對data/ 中的8個graph操作，並將結果存在result/hits/graph\_{i}.txt以及result/pagerank/graph\_{i}.txt中。
2. SimRank演算法實作code在simrank.py中，執行程式後會對data/ 中的前5個graph操作，並將結果存在result/simrank/graph\_{i}.txt中。
3. 根據不同的圖可以觀察到幾件事：
4. 如果graph是一個cycle，例如graph\_2.txt，則所有page的hub值跟auth值都會一樣。而每個page的pagerank也都是一樣的。
5. 如果graph是雙向的，例如graph\_3.txt，則每個page對應的hub值跟auth值是一樣的。互相指向的兩個page的pagerank值也會一樣。
6. 所有pagerank的總合為1。(轉移矩陣相乘後依舊是轉移矩陣)

Computation performance analysis

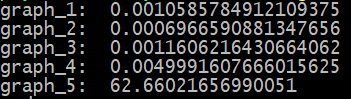
1. HITS運算時間



1. PageRank運算時間



1. SimRank運算時間



1. 從運算時間可以發現，node數及edge數越多，所需運算時間越長。

Find way to increase auth, hub, and pagerank

藉由增加1->1的edge，能夠有效增加node 1的hub值及auth值

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | prev\_hub | new\_hub | prev\_auth | new\_auth |
| graph\_1 | 0.447 | 0.999 | 0 | 0.707 |
| graph\_2 | 0.447 | 0.851 | 0.447 | 0.851 |
| graph\_3 | 0.372 | 0.657 | 0.372 | 0.657 |

藉由增加1->1, 1->3的edge，能夠有效增加node 1的pagerank值

|  |  |  |
| --- | --- | --- |
|  | prev\_pagerank | new\_pagerank |
| graph\_1 | 0.167 | 0.308 |
| graph\_2 | 0.2 | 0.261 |
| graph\_3 | 0.175 | 0.231 |

Questions & Discussion

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

以上實做的演算法，只考慮了page之間的link，並沒有將網頁的內容考慮進去，因此，如果一個網頁中都是不重要的內容，但卻有很多指向外部網站的連結的話，此網頁的hub值會高，但卻是一個沒有用的網頁。

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

在現實中，使用者期待的是搜尋出來的結果是真正有用的內容，且搜尋的速度要快，所以在實作時，應該在反應時間以及搜尋質量之間抓到一個平衡點。

Disadvantage of PageRank

Pagerank對主題較沒有相關性，不能區分頁面內的不同連結，容易發生對廣告頁面有過高的評價。另外，舊的page rank會比新的page高。因為即使是非常好的new page也不會有很多連結，除非它是某個站點的子站點。

Disadvantage of HITS

當一個網站上的多個網頁指向同一個相同的連結，容易引起評分的不正常增加。

Compare between HITS and PageRank

PageRank算法和HITS算法都是基于鍵接分析的搜索引擎排序算法，並且在算法中兩者都利用了特征向量作为理論基礎和收斂性依據。雖然兩種算法均為鍵接分析算法，但還是有明顯區別。HITS算法計算的authority值只是相對於某個檢索主题的權重，因此HITS算法也常被稱為Query-dependent算法；而PageRank算法是獨立於檢索主题，因此也常被稱為Query-independent算法。

HITS與使用者輸入的查詢有相關，而PageRank與查詢無關。所以，HITS可以單獨作為相似性計算標準，而PageRank必須結合其他內容相似性計算才可以進行搜尋的比較。

HITS因為與使用者查詢有關，所以必須在接收到使用者查詢後才能進行計算，計算效率較低；而PageRank則可以直接計算結果，計算效率較高。

HITS的計算數量較少，只需特定page之間的link；而PageRank則須對所有page進行計算。