Link Analysis Practice

Q56061040 湯立婷

Implementation detail - PageRank

damping factor 設 0.85：表示在任意時刻，用戶存取到某頁面後繼續存取下一個頁面的機率為0.85，而有0.15的機率會停止繼續存取新頁面

max_iteration = 100：我們假設最大迭代次數為100，超過100即停止迭代

頁面A的PR值為連結到頁面A的所有頁面的PR值的總和，若只有A, B, C三個頁面，B, C連到A，B也連到C，那麼PR(A) = PR(B)/2 + PR(C)

而為了符合馬可夫鏈收斂性，若A頁面沒有連到任何頁面，我們將指定A頁面必須連到所有頁面(包括A頁面自己)

首先，需給每個頁面一個初始的PR值為(1/所有頁面數)，

Implementation detail - PageRank (Cont.)

再設定damping value = (1 - damping factor) / (總頁數)

接著遍歷所有頁面，再將每一頁面的(PageRank值*damping factor) / 其頁面出射的頁數，再加上damping value

再將t時間點的PageRank值減去t-1時間點的PageRank值接著取絕對值

一直迭代著做，直到t時間點的PageRank值減去t-1時間點的PageRank值取絕對值後小於預先設定好的epsilon=0.00001，則停止

Implementation detail - HITS algorithm

首先初始化設定所有頁面的hub與authority值為1

進入迭代(最大次數100次)，再去遍歷每一頁面計算authority值，authority值為所有指向此頁面的hub值加總

接著對所有頁面的authority值取平方加總再開根號得出一個值

接著將每一個頁面的authority值都除以這個值，當作標準化

再將t+1時間點的authority值減去t時間點的authority值再取絕對值，這個值我們稱為ChangeAuthority

Implementation detail - HITS algorithm (Cont.)

接著跟計算authority值的流程差不多

去遍歷每一頁面計hub值，hub值為此頁面所指向的頁面的authority值加總

接著對所有頁面的hub值取平方加總再開根號得出一個值

接著將每一個頁面的hub值都除以這個值，當作標準化

再將t+1時間點的hub值減去t時間點的hub值再取絕對值，這個值我們稱為Changehub

最後將ChangeAuthority+Changehub若小於0.0001則迭代停止

Result analysis (PageRank value)

Graph1: finished in 6 iterations! The final page rank is
{'1': 0.06071525091827945, '2': 0.11232321419881698, '3': 0.15618998298727388, '4': 0.19347673645746225, '5': 0.22517047690712236, '6': 0.2521101562893335}

Graph2: finished in 5 iterations! The final page rank is
{'1': 0.2, '2': 0.2, '3': 0.2, '4': 0.2, '5': 0.2}

Graph3: finished in 4 iterations! The final page rank is
{'1': 0.037500000000000006, '2': 0.069375, '3': 0.4624876264454514, '4': 0.43061448247863365}

Graph4: finished in 7 iterations! The final page rank is
{'1': 0.05512500000000001, '2': 0.039642857142857146, '3': 0.021428571428571432, '4': 0.021428571428571432, '5': 0.4176663486083002, '6': 0.3764449677456266, '7': 0.06828482142857144}

Graph5: finished in 99 iterations! The final page rank is (Top 10)
[('282', 0.028837475882923985), ('348', 0.01909945042134983), ('440', 0.017836506584558622), ('457', 0.013244303899756874), ('331', 0.012631460427932267), ('468', 0.011133681956018012),('461', 0.0110655315989257),('415', 0.009189882199283225), ('429', 0.008449736238859312), ('433', 0.00841034139246494)]

Graph6: finished in 999 iterations! The final page rank is (Top 10)
[('1134', 0.05688331725203913),('1084', 0.051620262245154425),('1227', 0.030390558762010175),('1151', 0.026454782598913287),('1199', 0.013852804571925349),('1213', 0.005685810078390483),('1174', 0.004088926508520093),('1192', 0.003797842684995903),('1156', 0.003030491962949644),('1070', 0.002739432378039708)]

Result analysis (hub algo.)

Graph1: finished in 2 iterations!
The best authority page: ('2', 0.447213595499958)
The best hub page: ('1', 0.447213595499958)

Graph2: finished in 2 iterations!
The best authority page: ('1', 0.447213595499958)
The best hub page: ('1', 0.447213595499958)

Graph3: finished in 16 iterations!
The best authority page: ('3', 0.9999999997671692)
The best hub page: ('2', 0.7071067811042294)

Graph4:finished in 11 iterations!
The best authority page: ('5', 0.9999999999362672)
The best hub page: ('4', 0.5773502691773604)

Graph5: finished in 19 iterations!
The best authority page: ('348', 0.9999999993598458)
The best hub page: ('115', 0.2886751344871025)

Graph6: finished in 30 iterations!
The best authority page: ('1227', 0.9999999991463098)
The best hub page: ('1003', 0.16903085084263236)

Find a way (e.g., add/delete some links) to increase PageRank of Node 1 in first 3 graphs respectively.

Add links from node1 to all nodes
Graph1: finished in 6 iterations! The final page rank is
{'1': 0.07508644601818276}

Add links from node1 to all nodes
Graph2: finished in 5 iterations! The final page rank is
{'1': 0.3010734926986356

Add links from node1 to all nodes
Graph3: finished in 4 iterations! The final page rank is
{'1': 0.047619047619047763}

Below is the original PageRank

Graph1: finished in 6 iterations! The final page rank is
{'1': 0.060715250918279 45}

Graph2: finished in 5 iterations! The final page rank is
{'1': 0.2}

Graph3: finished in 4 iterations! The final page rank is
{'1': 0.037500000000000006}

Find a way (e.g., add/delete some links) to increase hub, authority of Node 1 in first 3 graphs respectively.

Add links from node1 to all nodes
Graph1: finished in 5 iterations!
authority {'1': 0.3645124459491862}
hub {'1': 0.9436282629706528}

Below is the original hub

Graph1: finished in 2 iterations!
authority {'1': 0.0}
hub {'1': 0.447213595499958}

Add links from node1 to all nodes
Graph2: finished in 5 iterations!
authority {'1': 0.461939787635968 93}
hub {'1': 0.9238795251747178}

Graph2: finished in 2 iterations!
authority {'1': 0.447213595499958}
hub {'1': 0.447213595499958}

Add links from node1 to all nodes
Graph3: finished in 9 iterations!
authority {'1': 0.3952932200825086}
hub {'1': 0.8781806212040688}

Graph3: finished in 16 iterations!
authority {'1': 0.0}
hub {'1': 1.0789593217532798e-05}

Computation performance analysis

Iteration number	**PageRank algo.**	**Hub algo.**
Graph1	6	2
Graph2	5	2
Graph3	4	16
Graph4	7	11
Graph5	99	19
Graph6	999	30

整體來說, hub演算法花費的迭代次數比較少

Discussion

PageRank :

1. 需要迭代的次數蠻多
2. 對新的網頁比較不友善，新網頁被連結到的次數本來就會蠻低，這樣要花很久的時間才能讓新網頁的PageRank value上升，而舊網頁的PageRank value就依然很穩定的持續高

hub :

1. 假設我寫一個網頁連結到很多high authority value的頁面，那麼我就可以成為一個high hub value的網頁，接著我再寫另一個頁面，將剛剛那個high hub value的網頁指過來，那我就可以成為一個high authority value的頁面，感覺有點容易作弊？