國立成功大學-資工所

Data Mining 課程

**Project 3**

**LinkAnalysis**

課程教授: 高宏宇

學生: 李俊賢

學號: P96074121

1. HITS程式實作
2. **作業環境**

作業系統: Window 10 Pro

處理器: intel i7-7700 @3.60GHz

記憶體: 8GB

程式語言: python

編譯器: Spyder3.6

1. **概略程式碼**



HITS的方法是將指向同節點的hub值加總去計算authority值，將節點指出去的authority值加總為節點的hub值。每個interation都將其值標準化讓值在0~1之間，重複直到收斂。

1. **data**

Graph\_1~6

1. **運算結果**

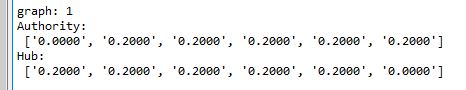
****

圖1.graph\_1結果



圖2.graph\_1示意圖

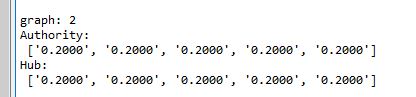
****

圖3.graph\_2結果



圖4.graph\_2示意圖

從結果來看可以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則hub值會是0，這個演算法因為要計算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。

1. PageRank程式實作
2. **作業環境**

作業系統: Window 10 Pro

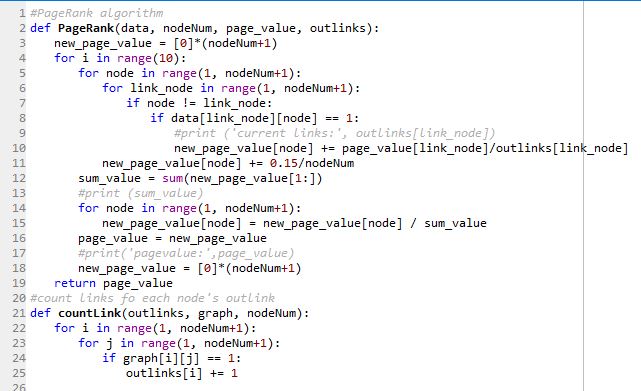
處理器: intel i7-7700 @3.60GHz

記憶體: 8GB

程式語言: python

編譯器: Spyder3.6

1. **概略程式碼**

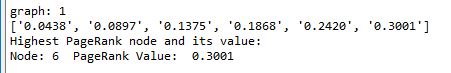
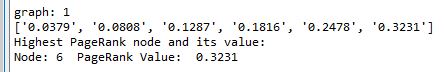
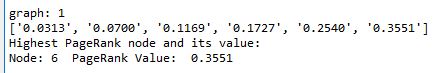


將每個節點的pagerank預設值為一，並進行計算。每個iteration將每個結點的pagerank值重新更新為連進此結點的結點的pagerank / vertex，一直重覆此步驟值到收斂。

1. **自己訂的data**

Graph\_1~6

1. **運算結果**

****

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。

1. SimRank程式實作
2. **作業環境**

作業系統: Window 10 Pro

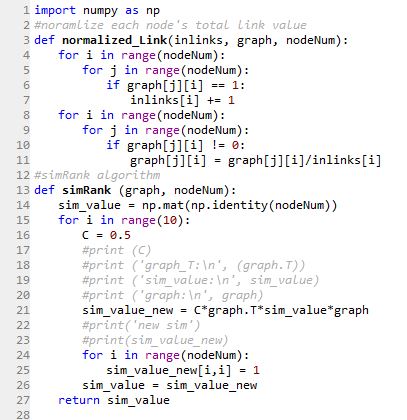
處理器: intel i7-7700 @3.60GHz

記憶體: 8GB

程式語言: python

編譯器: Spyder3.6

1. **概略程式碼**



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。

1. **自己訂的data**

Graph\_1~6

1. **運算結果**

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。

**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。

**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

**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。

1. **結論**

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所交易，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聯結所提高的，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去找出新的方法或解決方案，就是對研究領域的一種貢獻了。