線代啟示錄
I seek not to know the answers, but to understand the questions.

Google 搜尋引擎使用的矩陣運算

Posted on 05/02/2009 by ccjou

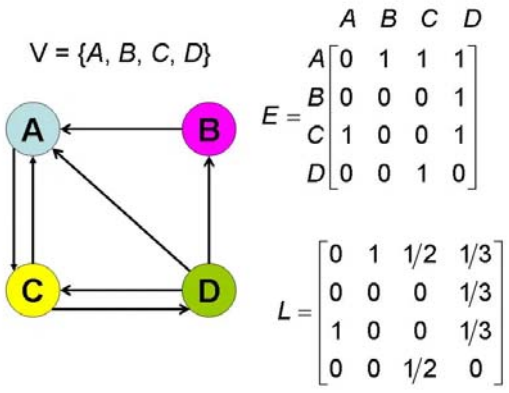
本文的閱讀等級：中級

Google 的技術總覽開宗明義說：

> Google 出類拔萃的地方在於專注開發「完美的搜尋引擎」，創辦人 Larry Page 將其定義為「能做到確實瞭解使用者想要的東西，並確實提供對應的資訊」...Google 使用 PageRank™ 檢查網路上的整個連結架構，藉此決定網頁的重要性。 接著會執行超文字符合分析，來判定哪些網頁與所執行的特定搜尋相關。結合所有的重要性和特定查詢的相關性後，Google 才會將最相關和最可靠的結果放在搜尋結果最前方。

PageRank 也稱為網頁排名，是 Google 創辦人 Larry Page 和 Sergey Brin 在史丹佛大學就讀研究所時所發展出的一項專利技術。與傳統的搜尋引擎倚重某字詞出現在網頁上的頻率不同，PageRank 完全由網路的超連結架構決定，基本的想法是「將網頁 A 連至網頁 B 的超連結，解譯為網頁 B 得到網頁 A 的一張選票，然後某網頁的 PageRank 再由收到的選票數量來評估其重要性。」

PageRank 體現了計算機科學、數學，以及資訊科技在現實應用的完美結合。我們從網路模型開始談起，以圖論中的有向圖 (directed graph) 表示網頁之間的超連結關係，設 $G = (V, E)$，其中 V 為包含 n 個網頁的頂點，E 為包含所有方向性的超連結。



一個小型互聯網

上圖顯示一個小型的互連網路，V 包含四個網頁 A，B，C，D，網頁之間的超連結以方陣 E 表示，若網頁 j 連至網頁 i，則 (i, j) 元為 1，否則為 0 (見“線性代數在圖論的應用 (一)：鄰接矩陣”)。矩陣 E 每一行 (column) 的所有元的總和表示所對應網頁的外部總連結數。考慮每一個網頁僅有一票，所以投出的每票要經過歸一化，矩陣 L 表示網頁 j 對於網頁 i 投出的分配票。每一個網頁的 PageRank 即為其所收到的總加權票數：

$$PR(A) = \frac{PR(B)}{1} + \frac{PR(C)}{2} + \frac{PR(D)}{3}$$
$$PR(B) = \frac{PR(D)}{3}$$
$$PR(C) = \frac{PR(A)}{1} + \frac{PR(D)}{3}$$
$$PR(D) = \frac{PR(C)}{2}$$

或寫成如下的矩陣形式：

$$\mathbf{x} = \begin{bmatrix} PR(A) \\ PR(B) \\ PR(C) \\ PR(D) \end{bmatrix} = L\mathbf{x}$$

現實上，某些網頁未被其他網頁連結，因此該網頁的 PageRank 值為 0。為了避免此問題發生，我們在每個網頁都加上一點「基本票」，以網頁 A 為例：

$$PR(A) = (1-p) + p\left(\frac{PR(B)}{1} + \frac{PR(C)}{2} + \frac{PR(D)}{3}\right)$$

數值 p 介於 0 與 1，稱為阻尼因數 (damping factor)，Google 將 p 值設為 0.85。原方程式因此改為

$$\mathbf{x} = (1-p)\mathbf{e} + pL\mathbf{x}$$

其中向量 $\mathbf{e}$ 的每一個元都是 1。

若限制所有網頁的 PageRank 總和為 n，便有 $\mathbf{e}^T\mathbf{x} = n$。將此條件置入上式，可得到等價方程式

$$\mathbf{x} = \left(\frac{1-p}{n}\mathbf{e}\mathbf{e}^T + pL\right)\mathbf{x} = M\mathbf{x}$$

矩陣 M 的每一元皆不為負，且每行之元的總和為 1，這種矩陣稱做 Markov 矩陣，可以解釋為 Markov 模型裡的轉移機率矩陣 (見“馬可夫過程”)。Perron-Frobenius 定理說：Markov 矩陣的最大特徵值等於 1，對應的特徵空間維數等於 1 (這保證特徵向量各元的比例是唯一的)，且該特徵向量的所有元皆為正數 (見“特殊矩陣 (21)：非負矩陣”)。矩陣 M 對應特徵值 1 的特徵向量 $\mathbf{x}$ 的元就是網頁的 PageRank。

上面小網路例子的矩陣 M 為 4 階方陣，很容易利用矩陣計算軟體得到特徵向量，為容易辨識，我將 PageRank 總和設為 1：

$$\mathbf{x} = \begin{bmatrix} 0.3330 \\ 0.0936 \\ 0.3762 \\ 0.1972 \end{bmatrix}$$

某網頁如果被許多 PageRank 高的網頁連結上，該網頁的 PageRank 便會上升 (如網頁 A 和 C)，這解釋了為什麼我們經常發現維基百科的網頁總是於 Google 搜尋結果的前方，因為維基百科內部有大量相互超連結。(這算是 PageRank 的一個缺點嗎？)

回到現實，根據 Google 於2008年七月公布的資料，全球網頁總數已達 10^{12}。理論上，PageRank 演算法必須解出

(見“Power 迭代法”)。Power 法是一種相當簡易的迭代法，將猜測的初始特徵向量與方陣 M 相乘再予以正規化，如下：

$$\mathbf{x}(k+1) = \frac{M\mathbf{x}(k)}{\|M\mathbf{x}(k)\|}$$

此法最後會收斂至對應最大特徵值的特徵向量。當然，Google 必定還要考慮網際網路的實際結構，將方陣 M 解構為若干優化的分塊矩陣，再派工給多個伺服器完成計算。

Google 是一個值得後人學習的成功典範。從開始先有網頁搜尋的創新觀念，再搭配資訊科技的領先技術，這兩個致勝因素──創新和技術，穩固了 Google 今天在網頁搜尋領域屹立不搖的龍頭地位。

Google 的 PageRank™ 的技術報告可於以下網址下載：
http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf

This entry was posted in 線性代數專欄, 圖論, 應用之道 and tagged 特徵值, 特徵向量, 馬可夫矩陣, Perron-Frobenius 定理, Power 迭代法, 分塊矩陣, 圖論. Bookmark the permalink.

7 Responses to *Google* 搜尋引擎使用的矩陣運算

ALeaf *says:*
05/03/2009 at 4:52 pm

What is the purpose of the damping factor in the page rank?
What effects of the damping factor on the whole system?

Reply

大俠 *says:*
05/03/2009 at 6:09 pm

meaning of df>>>
p indicates the fraction of time that the imaginary random click follows a link and 1-p is the fraction of time that an an arbitrary page is chosen, so u can think of 1-p as “seed”.

effect of df>>>
quoted from http://www2002.org/CDROM/poster/173.pdf
“as the damping factor (p) is reduced from 1 the gap between the principal and second eigenvalue will increase, and power iteration will converge more quickly.”

btw, it happens to me that a child of one of my colleagues is also named “a leaf.” how strange?

Reply

您好，
请问”考慮每個網頁僅有一票，所以投出的每票要經過正規化，”,这个是什么道理呢？为什么要正规化？为什么要考虑每个网页仅有一票？
谢谢！

Reply

turtle *says:*
04/17/2012 at 8:37 pm

一個網頁上有一個連結=>該連結得1分
一個網頁上有10個連結=>每個連結得0.1分
一個網頁上有100個連結=>每個連結得0.01分
應該是類似此方式的處理方式
我猜的0..0

Reply

ccjou *says:*
10/26/2010 at 7:38 pm

Page rake 既是一個演算法也是一個網路模型，模型的良窳由解釋能力，運算效率，應用範圍，強健性(假設條件未必成立)等因素決定。

Page rank 假定我們在網路上隨機瀏覽，當進入某個網站後便隨機點選該網站的連結，也就是說 page rake 將網路瀏覽行為看成 Markov chain，而且點選各連結的機率都相同，所以投出的每票(連結)才予以正規化。維基百科有張圖說明了這個概念
http://en.wikipedia.org/wiki/PageRank
根據這個模型，如果特徵向量各元總和為1，如本文的小型網路例子，我們停留在網站 C 的機率為 37.62%。

如果不將票數正規化，我們依然可以得到對應最大特徵值的特徵向量，但此特徵向量的意義就不是那麼具體。

Reply

question *says:*
10/26/2010 at 7:59 pm

哇，太棒了。多谢老师的讲解，
1：我看了这个博文，才知道pagerank的原理，一目了然，原来如此“简单”啊，是可以很顺的退出来的。而我看到别的介绍pagerank的网站，包括wikipedia，几乎都是避重就轻，直接在为什么要算1特征值和对应的特征向量的时候，一笔带过，看来他们也不是真明白。而你的这个很容易看懂，包括另外一篇的类似AHP，降低排名误差的文章。
Great work！
2：“Page rank 假定我們在網路上隨機瀏覽，當進入某個網站後便隨機點選該網站的連結，也就是說 page rake 將網路瀏覽行為看成 Markov chain，而且點選各連結的機率都相同，所以投出的每票(連結)才予以正規化。”，看来pagerank这个算法还有很大的改进余地啊。毕竟这个假设和真实情况完全不一样的，比如大部分人，进入一个网站，肯定对有些链接是一辈子都没有点过，对一些链接是每次进入都要点击。甚至，如果引入进入链接后，停留时间长短，统计后，结果是不是会更加精确？

thanks

Reply

wonderlandtommy *says:*
03/28/2016 at 10:45 pm

Google 2013年的Hummingbird演算法有重大革新! 可以辨識句子的主格、受格作出最佳回應！

Reply