PageRank

6030478021 นายเมวิน มาคารานัส

ภาควิชาวิศวกรรมคอมพิวเตอร์, คณะวิศวกรรมศาสตร์, จุฬาลงกรณ์มหาวิทยาลัย

PageRank หรือ PR อัลกอริทึมที่ใช้โดย Google Search เพื่อจัดอันดับเว็บเพจที่แสดงในผลลัพธ์ของการค้นหา โดยตั้งชื่อตาม Larry Page หนึ่งในผู้ก่อตั้ง Google และร่วมคิดค้นอัลกอริทึมนี้สมัยเรียนอยู่มหาวิทยาลัยสแตนฟอร์ด โดยมีสมมติฐานพื้นฐานคือเว็บไซต์ที่สำคัญมากมักจะถูกโยงลิงก์มาจากเว็บอื่นมากตามกันไป ซึ่งนอกจากจะเป็นอัลกอริทึมที่ใช้โดยเสิร์ชเอนจินชื่อดังอย่าง Google แล้ว มันยังเป็นอัลกอริทึมแรกที่ถูกใช้โดยบริษัทเอกชน และเป็นที่รู้จักอย่างกว้างขวางมากที่สุดอีกด้วย โดยเพจแรงก์จะแสดงเป็นค่าตัวเลขบ่งบอกถึงความสำคัญของข้อมูลในกลุ่มของชุดข้อมูลตัวเลขของเพจแรงก์ของกูเกิ้ลในปัจจุบัน จะมีค่าระหว่าง 0 ถึง 10 ซึ่งถูกคำนวณค่าในลักษณะลอการิทึม เพื่อแสดงถึงความสำคัญของหน้านั้นบนตัวค้นหาของ Google

1. หลักการทำงาน

PageRank Algorithm ให้ผลลัพธ์ออกมาเป็นการแจกแจงความน่าจะเป็นที่แสดงถึงความเป็นไปได้ที่คนหนึ่งจะคลิกลิงก์แบบสุ่มแล้วจะไปโผล่ที่หน้าใดหน้าหนึ่ง

จำลองค่าเว็บเพจสี่หน้า A B C และ D ค่าเริ่มต้นของ PageRank (PR) ในแต่ละหน้าจะมีค่าเท่ากันคือ 0.25 (เอาความน่าจะเป็นทั้งหมดคือ 1 หารด้วยจำนวนเว็บทั้งหมดคือ 4 ได้หน้าละ 0.25 ในตอนเริ่มต้น) ถ้าหน้า B C และ D ลิงก์ไปยังหน้า A จะเป็นการให้คะแนน 0.25 PR ต่อหน้า A ซึ่งค่า PR เขียนว่า PR( ) ในระบบจะกลายเป็น

PR(A) = PR(B) + PR(C) + PR(D) ซึ่งมีค่าเป็น 0.75 และถ้าหน้า B ยังคงลิงก์ไปยังหน้า C ขณะที่หน้า D ลิงก์ไปยังทุกหน้า ทำให้คะแนนจากหน้า B ถูกแบ่งออกสำหรับ A และ C เหลือเพียง 0.125 ขณะที่คะแนนจาก D จะเหลือให้แต่ละหน้าเป็นหนึ่งในสาม (ประมาณ 0.083)

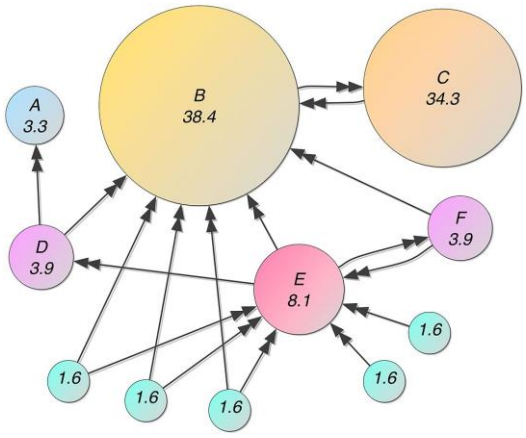
$$\mathbf{PR(A) = (1\text{-}D) + D * (\frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3})}$$

ซึ่งสามารถเขียนเป็นสมการได้ว่า เพจแรงก์ที่ให้คะแนนต่อหน้าอื่นนับตามลิงก์ที่ชี้ไปยังหน้าอื่น **L( )** มีค่าเท่ากับคะแนนเพจแรงก์ของหน้านั้นหารด้วยจำนวนลิงก์ที่ชี้ออกไป

$$\mathbf{PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}}$$

และสมการในลักษณะทั่วไปสำหรับหน้าใดๆ คือ

$$\boldsymbol{PR(u) = \sum \frac{PR(v)}{L(v)}}$$



รูปที่ 1 : ตัวอย่างเพจแรงก์สำหรับเครือข่ายอย่างง่าย โดย B มีค่าเพจแรงก์สูงสุดเพราะมีจำนวนหน้าที่ลิงก์เข้าหามากสุด และแม้ว่าจะมีหน้าที่ลิงก์มาหา E มากกว่า C แต่น้ำหนักของหน้าที่ลิงค์มาหา C สูงกว่าจึงทำให้ C มีค่าเพจแรงก์สูงกว่า E

2. โปรแกรม

ในส่วนของโปรแกรมนั้น ตัวอย่างจะขอใช้ภาษา python ในการเขียน

```
import numpy as np

def pagerank(M, eps=1.0e-8, d=0.85):
    N = M.shape[1]
    v = np.random.rand(N, 1)
    v = v / np.linalg.norm(v, 1)
    last_v = np.ones((N, 1), dtype=np.float32) * 100
    while np.linalg.norm(v - last_v, 2) > eps:
        last_v = v
        v = d * np.matmul(M, v) + (1 - d) / N
    return v


M = np.array([[0, 0, 0, 0, 1],
              [0.5, 0, 0, 0, 0],
              [0.5, 0, 0, 0, 0],
              [0, 1, 0.5, 0, 0],
              [0, 0, 0.5, 1, 0]])
v = pagerank(M, 0.001, 0.85)
```

รหัสที่ 1 ตัวอย่างฟังก์ชั่น PageRank ในภาษา python สำหรับหน้า 5 หน้า โดยใช้ numpy

ฟังก์ชั่นเพจแรงก์จะรับค่าทั้งหมด 3 ตัวคือ

-เมทริกซ์ M ซึ่งเป็น adjacency matrix โดย M(I,j) แสดงถึงลิงค์จาก j ไป I โดยที่ผลรวมของ I และ M(I,j) จะเท่ากับ 1 เสมอ

-eps quadratic error สำหรับ v โดยมีค่ามาตรฐานคือ 1.0e-8

-damping factor โดยมีค่ามาตรฐานคือ 0.85

โดยฟังก์ชั่นนี้จะคืนค่าเป็นเวกเตอร์ v ที่เก็บค่าเพจแรงก์ของแต่ละหน้า ซึ่งจะมีค่าตั้งแต่ 0 ถึง 1

3. ประสิทธิภาพการทำงาน

มีการเดินแบบสุ่มที่ง่ายและเร็วสำหรับคำนวณค่าเพจแรงก์ของแต่ละปมในเครือข่าย โดยอัลกอริทึมแบบง่ายนั้นทำงาน $O(log\ n/\epsilon)$ รอบพร้อมความน่าจะเป็นที่สูงสำหรับทุกกราฟ (ทั้งแบบ directed และ undirected) โดย n คือขนาดของเครือข่าย และ ϵ คือความน่าจะเป็นสำหรับเริ่มต้นใหม่ (1-ϵ อาจเรียกว่า damping factor ก็ได้) ที่ใช้ในการคำนวณค่าเพจแรงก์ นอกจากนั้นมันยังสามารถทำงานในแบบที่เร็วขึ้นได้ซึ่งทำงานทั้งหมด $O(\sqrt{log\ n}/\epsilon)$ รอบในกราฟแบบ undirected ทั้งสองอัลกอริทึมที่กล่าวมานั้นสามารถวัดค่าได้ โดยที่แต่ละปมนั้นประมวลผลและส่งเพียงไม่กี่บิทสำหรับการทำงานแต่ละรอบ

4. แหล่งอ้างอิง

[1] Thai Wikipedia “เพจแรงก์”

https://th.wikipedia.org/wiki/เพจแรงก์

[2] Wikipedia “PageRank”

https://en.wikipedia.org/wiki/PageRank

[3] PEECHA Khunphonaiam. Krieger (1 December 2005). "Stanford Earns $336 Million Off Google Stock". San Jose Mercury News, cited by redOrbit.

[4] Richard Brandt. "Starting Up. How Google got its groove". Stanford magazine.