Računanje PageRanka, s posebnim osvrtom na viseće čvorove

MTMAP - 1. seminarski rad

Doris Đivanović, Karlo Gjogolović, Vana Glumac

Akademska godina: 2024./2025.

1 Uvod

1998. godine Larry Page i Sergey Brin razvili su PageRank algoritam koji je preko dva desetljeća igrao ključnu ulogu u Google-ovom pretraživanju web-stranica. U ovom radu ukratko ćemo ga predstaviti, implementirati te primijeniti na manji set personaliziranih podataka. Posebno će nas zanimati pitanje: **Što se događa ako stranica nema linkova koji vode na ostale stranice?**

2 Osnovni pojmovi

Krenimo od pojma Markovljevog lanca pomoću kojeg ćemo modelirati vezu između web-stranica i i j.

Definicija. Neka je S prebrojiv skup stanja. Markovljev lanac je slučajni proces $X = (X_n : n \geq 0)$ na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, P)$ sa skupom stanja S koji zadovoljava:

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i),$$

$$\text{za svaki } i_0, \ldots, i_{n-1}, i, j \in S$$

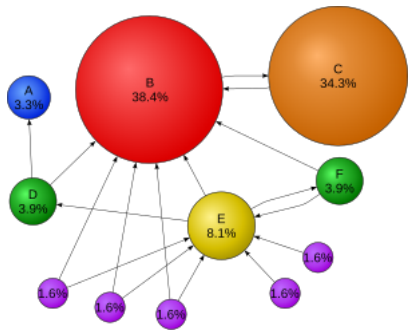
Dakle, za matricu $P = (p_{ij})$, gdje su $p_{ij} = P(X_{n+1} = j \mid X_n = i)$ za svaki $i, j \in S$ reći ćemo da stranica i sadrži vezu na stranicu j ako vrijedi $p_{ij} = 1$.

Definicija. Reći ćemo da je matrica $P = (p_{ij})$ retčano stohastička ako zadovoljava $p_{ij} \geq 0$ za svaki $i, j \in S$ te $\sum_{j=1}^{n} p_{ij} = 1$ za svaki $1 \leq i \leq n$.

Definicija. Neka je $X = (X_n : n \geq 0)$ Markovljev lanac s matricom prijelaza $P = (p_{ij})$. Distribucija π je stacionarna ako je zadovoljeno $\pi P = \pi$.

3 Definicija *Google* matrice i *PageRank* vektora

Glavni cilj ovog rada je izračunati "*vrijednost*" kojom je definirana važnost stranice: Stranica i je važnija što je veći broj na i-tom mjestu *PageRank* vektora. Prilikom implementacije algoritma bitno je imati na umu da ne nose sve stranice jednaku težinu tako da se može desiti da je neka stranica važnija od druge iako na nju pokazuje manje poveznica jer su stranice na kojima se nalaze poveznice važnije od onih sa druge stranice. Ukratko: poveznica sa važnije stranice nosi veću težinu od poveznice s manje važne stranice.



Slika 1: Rangiranje važnosti stranica. Iako na stranicu C pokazuje manje poveznica nego na stranicu E, ona je važnija jer na nju pokazuju važnije stranice.

Modelirajmo problem povezivanja stranica pomoću definicija iz prethodnog poglavlja.
Definicija. Neka je P matrica prijelaza Markovljevog lanca. Tada je problem povezivanja web-stranica moguće definirati kao:

$$p_{ij} = \begin{cases} 1 & \text{ako postoji poveznica sa stranice } i \text{ na stranicu } j, \\ 0 & \text{inače.} \end{cases}$$

U idućim koracima definirat ćemo *Google* matricu G.
Neka je A matrica susjedstva koja zadovoljava prethodnu definiciju te neka su (a_{ij}) elementi te matrice koji su jednaki 1 ukoliko postoji poveznica sa stranice i na stranicu j te 0 ako ne postoji. Ako za neki redak i vrijedi $\sum_{j=1}^{n} a_{ij} = 0$ tada ćemo reći da na stranici i ne postoje poveznice na iduću stranicu. Takve stranice nazivat ćemo viseći čvorovi (eng. *dangling nodes*). Pretpostavimo da takvih stranica ima k, pri čemu je $1 \leq k \leq n$ gdje je n broj web-stranica. Permutirajmo matricu A po retcima (i stupcima u svrhu zadržavanja originalnih veza među stranicama) s ciljem da takve stranice i dovedemo na dno matrice, tj. naša polazna matrica A sada postaje matrica $\tilde{A} = PAP^T$. Normiranjem matrice $\tilde{A}$ napokon dolazimo do matrice H:

$$H = \begin{bmatrix} H_{11} & H_{12} \\ 0 & 0 \end{bmatrix}$$

gdje su $H_{11} \geq 0$ i $H_{12} \geq 0$ redom matrice dimenzije $k \times k$ i $k \times (n-k)$ koje zadovoljavaju

$$H_{11}e + H_{12}e = e.$$

Želimo da matrica H bude stohastička i zbog toga umjesto svakog nul-retka stavljamo vektor-redak w^T gdje je $w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ za koji vrijedi $w \geq 0$ i $\|w\| = 1$, a w_1 i w_2 su redom dimenzija $k \times 1$ i $(n-k) \times 1$. Dakle, zbroj svakog retka u novoj matrici H koju ćemo označavati sa S bit će jednak 1, pa je po definiciji matrica stohastička. Formalno, veza između H i S zadana je kao

$$S \equiv H + \begin{bmatrix} 0 \\ e \end{bmatrix} \begin{bmatrix} w_1^T & w_2^T \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ ew_1^T & ew_2^T \end{bmatrix}.$$

Uzmimo proizvoljan vektor $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ takav da on zadovoljava $v \geq 0$ i $\|v\| = 1$ gdje su v_1 i v_2 redom vektori dimenzije $k \times 1$ i $(n-k) \times 1$.

Konačno, definirajmo stohastičku *Google* matricu G kao konveksnu kombinaciju dviju matrica:

$$G = \alpha S + (1-\alpha) e v^T$$

pri čemu je $0 \leq \alpha \leq 1$.

Zbog dodavanja matrice ev^T ranga jedan, matrica G ima jedinstvenu stacionarnu distribuciju π koju nazivamo *PageRank*. Dakle, zadovoljeno je $\pi \geq 0$ i $\|\pi\| = 1$ te $\pi^T G = \pi^T$. Pokazat ćemo da se distribucija π može dobiti iz matrice G *metodom potencija*.

4 Metoda potencija

Najopćenitiji način za računanje *PageRank* vektora je pomoću iterativne metode potencija. Ideja metode je izabrati proizvoljnu početnu distribuciju π_0^T pri čemu vrijedi

$$\pi_0^T \geq 0 \quad \text{i} \quad \|\pi_0^T\| = 1$$

te za k-tu distribuciju π_k^T iteracijom dobivamo

$$\pi_k^T = \pi_{k-1}^T G = \ldots = \pi_0^T G^k.$$

Prije nego što pustimo limes po k, osigurajmo da vektor π_k^T ne teži ekstremnim vrijednostima normiranjem vektora $\pi_{k-1}^T G$.

Sada imamo

$$\lim_{k\to\infty} \pi_k^T = \lim_{k\to\infty} \frac{\pi_{k-1}^T G}{\|\pi_{k-1}^T G\|} = \pi^T$$

pri čemu je π traženi *PageRank* vektor. Dakle, pronalazak *PageRank* vektora (točnije, njegove dovoljno dobre aproksimacije) svodi se na potenciranje matrice G za dovoljno velik broj k i množenje vektora π_0^T tom potenciranom matricom.

5 Pojednostavljenje *PageRank* algoritma reduciranjem dimenzije matrice

U ovom poglavlju pokazat ćemo kako se računanje *PageRank* vektora može pojednostaviti grupiranjem stranica sa kojih ne postoje poveznice (eng. *dangling nodes*) u jedan čvor. Ova metoda grupiranja u engleskoj se literaturi naziva *lumping*, a njezin cilj je grupirati sve viseće čvorove u jedan te na taj način ubrzati izračun stacionarne distribucije. Više o toj metodi moguće je pročitati u [2].

Postupkom uvedenim u *Poglavlju 3*, pomoću kojeg definiramo matricu G, dolazimo do većine pretpostavki u idućem rezultatu.

Teorem. Neka je dana matrica $G = \begin{bmatrix} G_{11} & G_{12} \\ eu_1^T & eu_2^T \end{bmatrix}$ pri čemu je $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \alpha w + (1-\alpha)v$ te $G_{11} \geq 0$ i $G_{12} \geq 0$ redom matrice dimenzije $k \times k$ i $k \times (n-k)$ koje zadovoljavaju $G_{11}e + G_{12}e = e$. Neka je $X \equiv \begin{bmatrix} I_k & 0 \\ 0 & L \end{bmatrix}$, gdje je $L \equiv I_{n-k} - \frac{1}{n-k}\hat{e}\hat{e}^T$ za $\hat{e} = e - e_1 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$.

Tada je $XGX^{-1} = \begin{bmatrix} G^{(1)} & * \\ 0 & 0 \end{bmatrix}$, gdje je $G^{(1)} = \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^T e \end{bmatrix}$

stohastička matrica reda $k+1$ s istim ne-nul svojstvenim vrijednostima kao i G.

Dokaz. Pogledati u [1].

Neka je sada σ stacionarna distribucija za manju matricu $G^{(1)}$.

Teorem. Neka je G matrica dobivena postupkom definiranim u *Poglavlju 3* te neka vrijedi $\sigma^T G^{(1)} = \sigma^T$, gdje je $G^{(1)} = \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^T e \end{bmatrix}$ te je zadovoljeno $\sigma \geq 0$ i $\|\sigma\| = 1$.

Ako particioniramo σ^T na $[\sigma_{1:k}^T \quad \sigma_{k+1}]$, tada vrijedi: $\pi^T = \left[\sigma_{1:k}^T \quad \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix}\right]$.

Dokaz. Pogledati u [1].

Prethodna dva rezultata poslužit će nam za stvaranje algoritma za izračun *PageRank* vektora π koristeći *power* metodu, ali na manjoj matrici $G^{(1)}$, te izračunavajući π, tj. podvektor vektora π koji odgovara visećim čvorovima, iz dobivene stacionarne distribucije matrice $G^{(1)}$.

Iz toga što je matrica $G^{(1)}$ puno manja od matrice G, odmah je jasno da je sama metoda potencija puno brža za matricu $G^{(1)}$ od metode potencija za matricu G. Treba još uočiti da je izračunavanje podvektora *PageRank* vektora π koji odgovara visećim čvorovima

efikasna operacija.

Nadalje, budući da je dimenzija matrice $G^{(1)}$ jednaka $k+1$, tj. za jedan (broj čvorova u koje smo spljoštili viseće čvorove) veća od k (tj. od broja nevisećih čvorova), zaključujemo da što je za istu dimenziju *Google* matrice G broj visećih čvorova veći, odnosno broj nevisećih manji, to je dimenzija male matrice sve manja od dimenzije matrice G, pa je sve efikasnije primijeniti ovaj algoritam umjesto metode potencija na cijeloj matrici G.

Algorithm 1: Algoritam za računanje *PageRank* vektora π

Input: H, v, w, α.

1: **Inicijalizacija** σ: Odaberi početni vektor $\sigma = [\sigma_{1:k}^T \quad \sigma_{k+1}]$ td. $\sigma \geq 0$ i $\|\sigma\| = 1$.
2: **while** se ne postigne konvergencija
Ažuriraj prvi blok:

$$\sigma_{1:k}^T = \alpha\sigma_{1:k}^T H_{11} + (1-\alpha)v_1^T + \alpha\sigma_{k+1}w_1^T$$

Ažuriraj završni blok:

$$\sigma_{k+1} = 1 - \sigma_{1:k}^T e$$

endwhile
3: **Rekonstrukcija vektora** π:

$$\pi^T = \left[\sigma_{1:k}^T \quad \alpha\sigma_{1:k}^T H_{12} + (1-\alpha)v_2^T + \alpha\sigma_{k+1}w_2^T\right].$$

Output: Aproksimacija PageRank vektora π.

Uvjerimo se da je ovaj algoritam izveden na temelju rezultata iz prethodnog teorema.

Ukoliko se sjetimo pravila za množenje dviju blok-matrica podijeljenih u blokove dimenzija takvih da su odgovarajuća množenja blokova definirana te ukoliko se sjetimo da, u skladu s oznakama uvedenim ranije, vrijedi

$$G_{11} = \alpha H_{11} + (1-\alpha)ev_1^T \qquad \text{i} \qquad u_1^T = \alpha w_1^T + (1-\alpha)v_1^T$$

te

$$G_{12} = \alpha H_{12} + (1-\alpha)ev_2^T \qquad \text{i} \qquad u_2^T = \alpha w_2^T + (1-\alpha)v_2^T,$$

možemo lako provjeriti da su dvije jednadžbe u pojedinoj iteraciji *while*-petlje gornjeg algoritma ekvivalentne jednadžbi

$$\left[\sigma_{1:k}^T \quad \sigma_{k+1}\right] = \left[\sigma_{1:k}^T \quad \sigma_{k+1}\right]\begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^t e \end{bmatrix}.$$

Dakle, pojedina iteracija *while*-petlje gornjeg algoritma uistinu je ekvivalentna računanju

$$\sigma^T = \sigma^T G^{(1)},$$

odnosno *while*-petlja gornjeg algoritma, tj. prvi dio gornjeg algoritma, uistinu jest metoda potencija za računanje jedinstvene stacionarne distribucije σ (točnije njezine dovoljno dobre aproksimacije) matrice $G^{(1)}$, matrice dimenzijom puno manje od matrice G.
Uočimo još da bi trebalo pisati

$$\sigma_{k+1} = \sigma_{1:k}^T G_{12} e + \sigma_{k+1} u_2^T e,$$

odnosno

$$\sigma_{k+1} = \sigma_{1:k}^T \left[\alpha H_{12} + (1-\alpha) e v_2^T\right] e + \sigma_{k+1} \left[\alpha w_2^T + (1-\alpha) v_2^T\right] e,$$

gdje su vektori e odgovarajućih dimenzija. Međutim, kako znamo da vrijedi

$$\|\sigma\| = 1,$$

za element σ_{k+1} vektora σ vrijedi

$$\sigma_{k+1} = 1 - \sum_{i=1}^{k} \sigma_i = 1 - \sigma_{1:k}^T e.$$

Uvjerimo se sada još da drugi dio gore napisanog algoritma, odnosno faza rekonstrukcije *PageRank* vektora π iz upravo dobivene stacionarne distribucije σ matrice $G^{(1)}$, odgovara drugom rezultatu iz posljednjeg teorema. Imamo

$$\sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} = \begin{bmatrix} \sigma_{1:k}^T & \sigma_{k+1} \end{bmatrix} \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix},$$

pa množenjem odgovarajućih blokova, te uvrštavanjem gore navedenih izraza za G_{12} i u_2, dobivamo

$$\sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} = \alpha \sigma_{1:k}^T H_{12} + (1-\alpha) v_2^T + \alpha \sigma_{k+1} w_2^T.$$

Konačno, budući da algoritam daje dovoljno dobru aproksimaciju stacionarne distribucije matrice $G^{(1)}$, napomenimo da je *PageRank* vektor kojeg daje algoritam zapravo dovoljno dobra aproksimacija stvarnog *PageRank*-a.

Zapitajmo se zašto dani algoritam particionira računanje rezultata.

Algoritam je, dakle, napisan na temelju rezultata iz posljednjeg teorema, ali nam upravo ovakav prikaz rezultata podijeljenih na dva bloka, odnosno podvektora, koji odgovaraju redom nevisećim i visećim čvorovima, daje izraze za eventualno zasebno izračunavanje svakog od tih podvektora, ali i jasno pokazuje od čega je sve svaki od tih dvaju podvektora nastao.

Očito, možemo zaključiti da se izračunavanje vektora $\pi_1 := \pi_{1:k}$ duljine k, odnosno podvektora *PageRank*-a π koji odgovara nevisećim čvorovima, dakle izračunavanje *PageRank*-a za skup svih nevisećih čvorova, provodi koristeći podvektore vektora w i v koji odgovaraju isključivo nevisećim čvorovima, te bloka matrice H koji odgovara vezama s nevisećih čvorova na neviseće čvorove. Dakle, *PageRank* vektor za skup svih nevisećih čvorova ni na koji način ne ovisi o parametrima vezanima za viseće čvorove naše mreže. Konačno, možemo zaključiti da *PageRank* vektor za skup svih nevisećih čvorova ne ovisi o broju visećih čvorova u mreži.

Također, iz izraza danih u gornjem algoritmu možemo zaključiti da vektor $\pi_2 := \pi_{k+1:n} = \pi_{k+1:k+(n-k)}$ duljine $n-k$, odnosno podvektor *PageRank* vektora π koji odgovara visećim čvorovima, ovisi o cijelom podvektoru π_1, odnosno podvektor π_2 zapravo nije moguće izračunati zasebno i neovisno od izračunavanja podvektora π_1 (tj, da bismo dobili vektor π_2 nužno je prvo dobiti cijeli vektor π_1).

Ponovno uočimo i navedimo prednosti ovog algoritma za računanje *Pagerank*-a u odnosu na metodu potencija za *Google* matricu.

Iako je brzina konvergencije prethodnog algoritma primijenjenog na reduciranu matricu $G^{(1)}$ ista kao za *power method* za polaznu *Google* matricu G, zbog toga što matrice G i $G^{(1)}$ imaju iste ne-nul svojstvene vrijednosti, algoritam je primijenjen na puno manju matricu $G^{(1)}$, pa je puno brži. Osim što je i dalje jednostavan za implementaciju a zahtijeva puno manje računskih operacija, ima i predvidivo ponašanje konvergencije koja mu omogućuje otpornost na promjene u strukturi matrice i velike *dataset*-ove.

Prethodni algoritam možemo proširiti na slučaj kada viseće čvorove želimo podijeliti u više od jedne klase prema nekom, proizvoljnom kriteriju.

6 Generalizacija - više klasa visećih vrhova

U prethodnim poglavljima predstavili smo tehniku koja sve nul-retke normalizirane matrice susjedstva H koja modelira veze među web-stranicama, dakle sve retke koji odgovaraju visećim čvorovima, zamjenjuje jednim istim (stohastičkim) vektorom-retkom w^T. Međutim, u složenijim primjenama, može biti potrebno ili poželjno razvrstavanje visećih čvorova u više različitih klasa. Na primjer, kod personaliziranog pretraživanja ili analize tematski povezanih stranica, može biti potrebno razlikovati različite klase. Ova potreba motivira proširenje algoritma, pa zato promatramo $m \geq 1$ klasa i uvodimo međusobno različite stohastičke vektore $w_1, w_2, \ldots, w_m$, pri čemu svaki vektor odgovara jednoj klasi. U ovom poglavlju pokušat ćemo objasniti postupak nalaženja *PageRank* vektora za *Google* matricu G kojoj su viseći čvorovi raspoređeni u više različitih klasa, proširiti metodu grupiranja, a zatim implementirati osmišljeni algoritam.

Generalizirajmo postupak kreiranja retčano stohastičke *Google* matrice s jedinstvenom stacionarnom distribucijom.

Ponovno, neka je dana matrica susjedstva A dimenzije n koja modelira veze između n web-stranica odnosno čvorova. Neka ponovno postoji k nevisećih stranica, odnosno neka postoji $n-k$ visećih stranica, dakle matrica A ima točno $n-k$ nul-redaka, koji odgovaraju visećim čvorovima. Međutim, neka sada postoji $m \geq 1$ klasa. Odredimo neki fiksan poredak klasa i označimo ih rednim brojevima $1, \ldots, m$. U ovom slučaju, najprije je potrebno rasporediti $n-k$ visećih čvorova u tih m klasa. Neka, nakon raspodjele, u klasi i ima k_i visećih čvorova, za svaki $i = 1, \ldots, m$. Dakle, $k_1 + k_2 + \cdots + k_m = n - k$.

Ponovno normiramo (po retcima) matricu A, a zatim je permutiramo i podijelimo u blokove na sljedeći način:

$$H \equiv \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1,i+1} & \cdots & H_{1,m+1} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix},$$

gdje blok H_{11} odgovara vezama s nevisećih čvorova na neviseće čvorove, dakle dimenzije je $k \times k$, a blok $H_{1,i+1}$ odgovara vezama s nevisećih čvorova na viseće čvorove iz klase i, dakle dimenzije je $k \times k_i$, za svaki $i = 1, \ldots, m$. Donjih $n-k$ redaka matrice H su nul-retci koji odgovaraju visećim čvorovima, ali tako da najgornjih k_1 nul-redaka odgovara visećim čvorovima iz klase 1, ..., najdonjih k_m redaka odgovara visećim čvorovima iz klase m.

Definirajmo sada za svaki $j = 1, \ldots, m$ vektor w_j dimenzije $n \times 1$, t.d.

$$w_j \geq 0 \qquad \text{i} \qquad \|w_j\| = 1,$$

i označimo

$$w_j^T = \begin{bmatrix} w_{j1}^T & w_{j2}^T & \cdots & w_{j,i+1}^T & \cdots & w_{j,m+1}^T \end{bmatrix},$$

gdje podvektor w_{j1} odgovara nevisećim čvorovima, dakle dimenzije je $k \times 1$, a podvektor $w_{j,i+1}$ odgovara visećim čvorovima iz klase i, dakle dimenzije je $k_i \times 1$, $\forall i = 1, \ldots, m$.

Sada, vektorom-retkom w_j^T zamijenimo sve nul-retke u matrici H koji odgovaraju visećim čvorovima iz klase j, $\forall j = 1, \ldots, m$. Dobivenu matricu označimo s S.

Odredimo sada jedan vektor v dimenzije $n \times 1$, t.d.

$$v \geq 0 \qquad \text{i} \qquad \|v\| = 1,$$

i označimo

$$v^T = \begin{bmatrix} v_1^T & v_2^T & \cdots & v_{i+1}^T & \cdots & v_{m+1}^T \end{bmatrix},$$

gdje podvektor v_1 odgovara nevisećim čvorovima, dakle dimenzije je $k \times 1$, a podvektor v_{i+1} odgovara visećim čvorovima iz klase i, dakle dimenzije je $k_i \times 1$, za svaki $i = 1, \ldots, m$.

Konačno, *Google* matrica je konveksna kombinacija matrica

$$F \equiv \alpha S + (1-\alpha)ev^T, \qquad 0 \le \alpha \le 1,$$

gdje je α fiksan. Ova je matrica po konstrukciji retčano stohastička matrica koja ima jedinstvenu stacionarnu distribuciju, *PageRank* vektor π dimenzije $n \times 1$, dakle vrijedi

$$\pi^T F = \pi^T, \qquad \pi \ge 0, \quad \|\pi\| = 1.$$

Ukoliko označimo

$$u_i = \alpha w_i + (1-\alpha)v \qquad i = 1, \ldots, m,$$

imamo

$$u_{i1} = \alpha w_{i1} + (1-\alpha)v_1, \quad \text{dimenzije } k \times 1,$$

i

$$u_{i,j+1} = \alpha w_{i,j+1} + (1-\alpha)v_{j+1}, \quad \text{dimenzije } k_j \times 1, \quad j = 1, \ldots, m.$$

Tada, *Google* matricu možemo pisati ovako:

$$F = \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1,m+1} \\ eu_{11}^T & eu_{12}^T & \cdots & eu_{1,m+1}^T \\ \vdots & \vdots & \ddots & \vdots \\ eu_{m1}^T & eu_{m2}^T & \cdots & eu_{m,m+1}^T \end{bmatrix},$$

gdje blok F_{11} odgovara vezama s nevisećih čvorova na neviseće čvorove, dakle dimenzije je $k \times k$, a blok $F_{1,i+1}$ odgovara vezama s nevisećih čvorova na viseće čvorove iz klase i, dakle dimenzije je $k \times k_i$, za svaki $i = 1, \ldots, m$. Nadalje, za svaki $i = 1, \ldots, m$, podredak u_{i1}^T odgovara retcima visećih čvorova iz klase i i nevisećim čvorovima, dakle dimenzije je $1 \times k$, a podredak $u_{i,j+1}^T$ odgovara odgovara retcima visećih čvorova iz klase i i visećim čvorovima iz klase j, dakle dimenzije je $1 \times k_j$, za svaki $j = 1, \ldots, m$.

Generalizirajmo grupiranje visećih čvorova grupiranjem svake od $m \ge 1$ klasa u jedan čvor.

Grupiranje provodimo primjenom niza transformacija sličnosti. Svaka transformacija postupno spaja (ili grupira) klasu po klasu, smanjujući dimenziju matrice počevši od posljednje klase prema prvoj.
Tako npr. za posljednju klasu koristimo matricu:

$$X_1 = \begin{bmatrix} I_{n-k_m} & 0 \\ 0 & L_1 \end{bmatrix},$$

pri čemu je

$$L_1 = I_{k_m} - \frac{1}{k_m}\hat{e}\hat{e}^T,$$

gdje je k_m broj čvorova u posljednjoj klasi.

Nakon primjene transformacije, nova matrica poprima oblik:

$$X_1 F X_1^{-1} = \begin{bmatrix} F_{11}^{(1)} & * \\ 0 & 0 \end{bmatrix},$$

gdje je $F_{11}^{(1)}$ stohastička matrica sa istim ne-nul svojstvenim vrijednostima kao i polazna F. Dakle, $F_{11}^{(1)}$ uključuje sve prethodne klase i sažetu informaciju o zadnjoj klasi. Analogan postupak se ponavlja za svaku klasu $m-1, m-2, \ldots, 1$ gdje se za i-tu klasu koristi transformacijska matrica

$$X_i = \begin{bmatrix} I_{k+(m-i)} & 0 & 0 \\ 0 & L_i & 0 \\ 0 & 0 & I_{m-i} \end{bmatrix},$$

pri čemu je

$$L_i = I_{k_i} - \frac{1}{k_i} \hat{e}\hat{e}^T,$$

gdje k_i broj čvorova u trenutnoj klasi i.

Nakon m ponavljanja, dobiva se konačna matrica:

$$\begin{bmatrix} F^{(1)} & * \\ 0 & 0 \end{bmatrix} \qquad \text{gdje je } F^{(1)} = \begin{bmatrix} F_{11} & F_{12}e & \cdots & F_{1,m+1}e \\ u_{11}^T & u_{12}^T e & \cdots & u_{1,m+1}^T e \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1}^T & u_{m2}^T e & \cdots & u_{m,m+1}^T e \end{bmatrix}.$$

Ova matrica je stohastička, dimenzije $k+m$, te ima iste ne-nul svojstvene vrijednosti kao i prvobitna matrica F.

Generalizirajmo teorijske rezultate za računanje *PageRank*-a.

Nakon višestrukog grupiranja, izračun PageRank-a postaje učinkovitiji jer radimo s manjom matricom te vrijede svi rezultati koje smo prethodno opisali za matricu G. PageRank π izračunava se pomoću stacionarne distribucije ρ matrice $F^{(1)}$ power metodom opisanom u *Poglavlju 4*:

$$\rho^T F^{(1)} = \rho^T, \quad \rho \geq 0, \quad \|\rho\| = 1.$$

Particioniranjem $\rho^T = [\rho_{1:k}^T \quad \rho_{k+1:+k+m}^T]$ gdje je $\rho_{k+1:k+m}$ dimenzije $m \times 1$, *PageRank* π dobivamo kao:

$$\pi^T = \begin{bmatrix} \rho_{1:k}^T & \rho^T \begin{pmatrix} F_{12} & \cdots & F_{1,m+1} \\ u_{12}^T & \cdots & u_{1,m+1}^T \\ \vdots & \ddots & \vdots \\ u_{m2}^T & \cdots & u_{m,m+1}^T \end{pmatrix} \end{bmatrix}.$$

Očito, za dani $m \geq 1$ dimenzija matrice $F^{(1)}$ jednaka je $k + m$, tj. za m (broj klasa visećih čvorova koje smo spljoštili u po jedan čvor) veća od k, jer smo transformacijama sličnosti, za svaki $i = 1, ..., m$, odgovarajućih k_i redaka matrice G "spljoštili" u jedan redak.

Konačno, pokušajmo dobiti generalizaciju algoritma kojeg smo dali ranije, onog koji particionira rezultate.

Dakle, teorijski rezultat ponovno kaže da je za računanje *PageRank*-a najprije potrebno izračunati stacionarnu distribuciju matrice $F^{(1)}$ dimenzije $k + m$ metodom potencija. Dakle, potrebno je iterativno računati umnoške

$$\rho^T = \rho^T F^{(1)}.$$

Sada, analogno kao u slučaju $m = 1$, sjetimo se da ako matrice koje sudjeluju u umnošku podijelimo u blokove na sljedeći način:

$$\begin{bmatrix}\rho_{1:k}^T & \rho_{k+1} & \cdots & \rho_{k+m}\end{bmatrix} = \begin{bmatrix}\rho_{1:k}^T & \rho_{k+1} & \cdots & \rho_{k+m}\end{bmatrix} \begin{bmatrix} F_{11} & F_{12}e & \cdots & F_{1,m+1}e \\ u_{11}^T & u_{12}^T e & \cdots & u_{1,m+1}^T e \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1}^T & u_{m2}^T e & \cdots & u_{m,m+1}^T e \end{bmatrix},$$

zbog pravila za množenje dviju blok-matrica podijeljenih u blokove dimenzija takvih da su odgovarajuća množenja blokova definirana, toj je jednakosti ekvivalentno sljedećih $1 + m$ jednadžbi:

$$\begin{aligned} \rho_{1:k}^T &= \rho_{1:k}^T F_{11} + \sum_{i=1}^{m} \rho_{k+i} u_{i1}^T \\ \rho_{k+i} &= \rho_{1:k}^T F_{1,i+1} e + \sum_{j=1}^{m} \rho_{k+j} u_{j,i+1}^T e, \qquad i = 1, \ldots, m. \end{aligned}$$

Nadalje, sjetimo se da vrijedi

$$F_{11} = \alpha H_{11} + (1-\alpha) e v_1^T \quad \text{i} \quad u_{i1}^T = \alpha w_{i1}^T + (1-\alpha) v_1^T \quad \text{za svaki} \quad i = 1, \ldots, m,$$

pa prvu jednadžbu možemo pisati ovako:

$$\boxed{\rho_{1:k}^T = \alpha \rho_{1:k}^T H_{11} + (1-\alpha) v_1^T + \alpha \sum_{i=1}^{m} \rho_{k+i} w_{i1}^T.}$$

Također, budući da za svaki $i = 1, \ldots, m$ vrijedi

$$F_{1,i+1} = \alpha H_{1,i+1} + (1-\alpha) e v_{i+1}^T$$

i

$$u_{j,i+1}^T = \alpha w_{j,i+1}^T + (1-\alpha) v_{i+1}^T \quad \text{za svaki} \quad j = 1, \ldots, m,$$

ostalih m jednadžbi izgleda ovako:

$$\boxed{\rho_{k+i} = \rho_{1:k}^T \left[\alpha H_{1,i+1} + (1-\alpha) e v_{i+1}^T\right] e + \sum_{j=1}^{m} \left[\rho_{k+j} (\alpha w_{j,i+1}^T + (1-\alpha) v_{i+1}^T) e\right]}$$

$$\text{za svaki} \qquad i = 1, \ldots, m,$$

gdje su vektori e odgovarajućih dimenzija.

Iz ovih $1 + m$ jednadžbi možemo rekonstruirati cijeli ρ^T jer vrijedi sljedeća particija:

$$\boxed{\rho^T = \begin{bmatrix} \rho_{1:k}^T & \rho_{k+1} & \cdots & \rho_{k+m} \end{bmatrix}}$$

Podredak $\rho_{1:k}^T$ dimenzije $1 \times k$ odgovara k nevisećih čvorova, a preostalih m skalara odgovara pojedinim čvorovima u koje smo grupirali svaku od m klasa visećih čvorova.

Iz ovakvih jednadžbi možemo uočiti da su one stvarno generalizacija odgovarajućih jednadžbi u algoritmu danom za $m = 1$, dakle pokazali smo da je prva faza algoritma za računanje *Pagerank*-a za $m \geq 1$ stvarno generalizacija prve faze algoritma danog za računanje *Pagerank*-a za $m = 1$.

Pokušajmo sada dobiti analogno za drugu fazu algoritma, odnosno particiju *PageRank* vektora π.

Teorijski rezultat i u generalnom slučaju kaže da vrijedi:

$$\pi^T = \begin{bmatrix} \rho_{1:k}^T & \rho^T \begin{pmatrix} F_{12} & \cdots & F_{1,m+1} \\ u_{12}^T & \cdots & u_{1,m+1}^T \\ \vdots & \ddots & \vdots \\ u_{m2}^T & \cdots & u_{m,m+1}^T \end{pmatrix} \end{bmatrix}.$$

Particionirajmo *PageRank* vektor π dimenzije $n \times 1$ na sljedeće blokove:

$$\boxed{\pi^T = \begin{bmatrix} \pi_{1:k}^T & \pi_{k+1:k+k_1}^T & \pi_{k+k_1+1:k+k_1+k_2}^T & \cdots \end{bmatrix},}$$

gdje podvektor $\pi_{1:k}$, dimenzije $k \times 1$, odgovara nevisećim čvorovima, a podvektor

$$\Pi_{\left(k + \sum_{j=1}^{i-1} k_j\right) + 1 \,:\, \left(k + \sum_{j=1}^{i-1} k_j\right) + k_i} \qquad \text{dimenzije} \quad k_i \times 1,$$

odgovara visećim čvorovima iz klase i, za svaki $i = 1, \ldots, m$.

Sada, očito vrijedi

$$\begin{bmatrix} \pi^T_{k+1:k+k_1} & \pi^T_{k+k_1+1:k+k_1+k_2} & \cdots \end{bmatrix} = \rho^T \begin{pmatrix} F_{12} & \cdots & F_{1,m+1} \\ u^T_{12} & \cdots & u^T_{1,m+1} \\ \vdots & \ddots & \vdots \\ u^T_{m2} & \cdots & u^T_{m,m+1} \end{pmatrix},$$

a to je

$$\begin{bmatrix} \pi^T_{k+1:k+k_1} & \pi^T_{k+k_1+1:k+k_1+k_2} & \cdots \end{bmatrix} = \begin{bmatrix} \rho^T_{1:k} & \rho_{k+1} & \cdots & \rho_{k+m} \end{bmatrix} \begin{pmatrix} F_{12} & \cdots & F_{1,m+1} \\ u^T_{12} & \cdots & u^T_{1,m+1} \\ \vdots & \ddots & \vdots \\ u^T_{m2} & \cdots & u^T_{m,m+1} \end{pmatrix}.$$

Konačno, množenjem odgovarajućih blokova, dobivamo izraze za računanje svakog pojedinog podvektora *PageRank* vektora π iz gornje particije:

$$\boxed{\Pi^T_{\left(k+\sum_{j=1}^{i-1} k_j\right)+1 \,:\, \left(k+\sum_{j=1}^{i-1} k_j\right)+k_i} = \alpha\rho^T_{1:k}H_{1,i+1} + (1-\alpha)v^T_{i+1} + \alpha\sum_{j=1}^{m}\rho_{k+j}w^T_{j,i+1}}$$

$$\text{za svaki} \qquad i = 1, \ldots, m.$$

Iz ovog izraza možemo uočiti da je on stvarno generalizacija odgovarajućeg izraza u rekonstrukciji *PageRank*-a π iz stacionarne distribucije σ, tj. u drugoj fazi, algoritma danog za $m = 1$. Ondje je podvektor *PageRank*-a π koji odgovara visećim čvorovima bio cijeli u jednom bloku, jer je postojala samo jedna klasa visećih čvorova, a ovdje je podijeljen u m blokova.

Mi ćemo, stoga, implementirati upravo ovakav algoritam za računanje *Pagerank*-a, koji je za proizvoljan $m \geq 1$, u što spada i $m = 1$, u stanju "izbaciti" rezultate za sve podvektore u particiji traženog *PageRank* vektora π. Ulazni podaci za algoritam su matrica H, vektori w_i, $i = 1, \ldots, m$, vektor v i faktor α. Algoritmu ćemo dati i kriterij zaustavljanja iterativne metode potencija te maksimalni broj iteracija.

Zanima nas usporedba vremena izvršavanja ovog algoritma na transformiranoj i algoritma metode potencija provedene na početnoj *Google* matrici dimenzije n. Promatrat ćemo što se događa s razlikom u vremenima izvršavanja kako raste dimenzija matrice, tj. broj web-stranica.

Nadalje, promatrat ćemo povećava li se stvarno razlika u vremenima izvršavanja, naravno u korist ovog algoritma, kako za neku fiksnu dimenziju matrice, tj. za fiksan broj web-stranica, povećavamo broj visećih čvorova.

Proširenje modela na više klasa korisno je za:

- Pretraživanje ovisno o namjenama visećeg čvora, npr. to može biti slika, videozapis ili neki drugi medijski sadržaj
- Preciznije upravljanje neželjenim poveznicama, poput spam stranica.
- Personalizaciju pretraživanja na temelju interesa korisnika.
- Pretraživanje prema jeziku ili geografskim specifičnostima.

7 Literatura

[1] Ipsen I.C.,Selee T.M. (2008) PageRank computation, with special attention to dangling nodes. SIAM Journal on Matrix Analysis and Applications. 29(4)128: 1–1296. doi:10.1137/060664331.
[2] J. G. Kemeny and J. L. Snell, Finite Markov Chains, Van Nostrand Reinhold Company, 1960
[3]Langville, A. and Meyer, C. (2004) 'Deeper inside PageRank', Internet Mathematics, 1(3), pp. 335–380. doi:10.1080/15427951.2004.10129091.
[4]Bryan, K. and Leise, T. (2006) 'The 25,000,000,000 eigenvector: The linear algebra behind Google', SIAM Review, 48(3), pp. 569–581. doi:10.1137/050623280.