CALCULUL UNOR MĂSURI NECESARE CĂUTARII PE WEB

MARIN POPA
Universitatea Spiru Haret, Facultatea de Matematică-Informatică,
marpopa2002@yahoo.com

MARIANA POPA
Universitatea Spiru Haret, Facultatea de Matematică-Informatică,

Rezumat: În articol se definesc câteva măsuri necesare pentru a uşura căutarea paginilor pe WEB şi se prezintă modalităţi eficiente de calcul al acestor măsuri.

Cuvinte cheie: gradul sau importanţa unei pagini, index, autoritate, grad de indexare, grad de autoritate, grad relativ al unui bloc

1. INTRODUCERE

Definim câteva măsuri [4] printre care: importanţa unei pagini de pe Web, gradul relativ al unui bloc corespunzător unui domeniu de pe Web, gradul de indexare al unei pagini, gradul de autoritate al unei pagini şi altele, măsuri care uşurează căutarea paginilor de pe Web. Pentru aceasta se consideră Web-ul ca un digraf [1,5] de dimensiuni uriaşe, în care nodurile corespund paginilor de pe Web iar arcele corespund link-urilor dintre pagini. De asemenea, se consideră Web-ul ca un digraf [5] în care nodurile corespund domeniilor de pe Web iar arcele sunt date de legăturile existente între pagini din aceste domenii. Se calculează apoi măsurile definite, local pe subgrafurile corespunzătoare domeniilor şi apoi după calculul acestora pe graful Web în care acum noduri sunt domeniile, se concatenează rezultatele găsite pentru a obţine valorile globale ale acestor măsuri.

2. CONSIDERAŢII GENERALE

Fie X o mulţime finită de elemente şi F o familie de submulţimi ale lui X. $F = \{S_1, S_2, \ldots, S_p\}$ unde $S_i \subset X$, $i=1,\ldots,p$. Construim $Y \subset X$ formată din elementele lui X care se regăsesc în submulţimile lui F, adică :

$$Y = \{ x \in X / \exists i=1,\ldots,p \text{ a.i. } x \in S_i \}$$

Definiţia 1. *Fie* $A \subset Y$, $A \neq \emptyset$. *Numim suport al mulţimii* A *şi notăm* sup A *mulţimea*

Analele Universităţii SPIRU HARET, Seria Matematică-Informatică, nr. 1 , 2005, ISSN 1841 - 7833

sup A = card $\{ i \mid i=1,...,p$ şi $A \subset S_i\}$.

Neformalizat sup A este de fapt numărul aparițiilor lui A în submulțimile familiei F.

Spunem că $A \subset Y$ este mulțime q frecventă în F dacă sup $A \geq q$, unde q este un prag minim dat.

Observația 1. *Dacă $A \subset Y$ este mulțime q frecventă în F atunci orice submulțime a sa este de asemenea frecventă în F.*

Spunem că $A \subset Y$ este mulțime maximal q frecventă în F dacă nu există o supramulțime a sa în Y care să fie q frecventă în F .

Pornind de la observația de mai sus , pentru a determina mulțimile q-frecvente în F, pentru un q dat , se procedează din aproape în aproape , găsind mai întâi mulțimile q-frecvente cu un singur element, apoi mulțimile q-frecvente cu două elemente şi aşa mai departe, până se găsesc mulțimile q-frecvente cu un număr n de elemente, unde $n \leq$ card Y.

Algoritmul de găsire a mulțimilor frecvente [2] se poate formaliza astfel:
Intrare

Pragul q, mulțimile $S_1, S_2, \ldots, S_p$

P1. Pentru fiecare $x \in X$ se determină sup $\{x\}$. Fie

$L_1 = \{ x \in X \mid \sup\{x\} \geq q\}$

şi $Y = \{ x \in X \mid \sup\{x\} \neq \emptyset\}$

P2. Pentru fiecare (x,y) Ţ L_1 x L_1 se determină sup {x,y}. Fie

$L_2 = \{(x,y) \in L_1 \times L_1 \mid sup\{x,y\} \geq q \}$

P3. Pentru fiecare i=3,…, card Y se construieste

$C_i = \{ (x_1, x_2, \ldots, x_i) \in X^i \mid (x_{j_1}, x_{j_2}, ..., x_{j_{i-1}}) \in L_i$, unde

$j_1, j_2, ..., j_{i-1} \in \{1, 2, ..., i\}\}$

Se construieşte $L_i = \{(x_1, ..., x_i) \in C_i \mid \sup\{x_1, ..., x_i\} \geq q\}$.

Ieşire: $L_1, L_2, ..., L_{\text{card } Y}$.

Exemplul 1. Pentru $X = \{A, B, C, D, E, F, G, H, I, J, K\}$, $q = 2$

şi $F = \{\{A, C, D\}, \{B, C, E\}, \{A, B, C, E\}, \{B, E\}\}$ avem:

P1. $\sup\{A\} = 2$, $\sup\{B\} = 3$, $\sup\{C\} = 3$, $\sup\{D\} = 1$, $\sup\{E\} = 3$,
$\sup\{F\} = \sup\{G\} = \sup\{H\} = \sup\{I\} = \sup\{J\} = \sup\{K\} = 0$.

Astfel $Y = \{A, B, C, D, E\}$ şi $L_1 = \{A, B, C, E\}$.

P2. Deoarece $L_1 \times L_1 = \{(A, B), (A, C), (A, E), (B, C), (B, E), (C, E)\}$ avem:
$\sup\{A, B\} = 1, \sup\{A, C\} = 2, \sup\{A, E\} = 1,$
$\sup\{B, C\} = 2, \sup\{B, E\} = 3, \sup\{C, E\} = 2.$

Rezultă $L_2 = \{(A, C), (B, C), (B, E), (C, E)\}$.

Deoarece $|Y| = 5$ avem de analizat cazurile $i = 3, i = 4, i = 5.$

P3. Pentru $i = 3$.

$C_3 = \{(x_1, x_{2,} x_3) \in Y^3 \mid (x_{i_1}, x_{i_2}) \in L_2\} = \{(B, C, E)\}$

$\sup\{B, C, E\} = 2$

Rezultă $L_3 = \{(B, C, E)\}$.

Pentru $i = 4$

$C_4 = \{(x_1, x_{2,} x_3, x_4) \in Y^4 \mid (x_{i_1}, x_{i_2}, x_{i_3}) \in L_3, \ \ i_1, i_2, i_3 \in \{1, 2, 3, 4\}\} = \varnothing,$

$\Rightarrow L_5 = \varnothing$.

Pentru $i = 5$, evident $C_5 = \varnothing$ şi deci $L_5 = \varnothing$.

Ieşire: avem următoarele mulțimi 2-frecvente:

$L_1 = \{A, B, C, E\}, \ \ L_2 = \{(A, C), (B, C), (B, E), (C, E)\},$

$L_3 = \{(B, C, E)\}$

Se observă că $\{B, C, E\}$ este o mulțime maximal 2-frecventă.

Exemplul 2. Pentru $X = \{A, B, C, ..., H\}, \ \ q = 3$ şi $F = \{\{A, C, D\}, \{B, C, E\}, \{A, B, C, E\}, \{B, E\}\}$ folosind calculele din exemplul 1 avem:

$L_1 = \{B, C, E\}$

$L_2 = \{B, E\}$

$L_3 = L_4 = L_5 = \varnothing$

Rezultă că mulțimi 3-frecvente sunt doar: $\{B\}, \{C\}, \{E\}$ si $\{B, E\}$.

În plus se observă că {B,E} este mulțime maximal 3-frecventă.

Observația 2. *Dacă* $\exists i = 2, ..., \text{card } Y$ *astfel încât* $L_i = \varnothing$ *atunci pentru orice* $j \geq i, \ \ L_j = \varnothing$.

Rezultă din această observație că algoritmul se poate opri cu mult înainte ca *i* să ajungă la card Y. Cu toate acestea complexitatea temporală şi sparțială a algoritmului este exponențială deoarece trebuie analizate submulțimi ale lui Y care sunt în număr de $2^{\text{card}Y}$.

3. GRADUL (IMPORTANȚA) UNEI PAGINI PE WEB

Problema căutării unui topic pe WEB a fost rezolvată de **Kleinberg [4]** folosind **algoritmul PageRang** (rangul unei pagini) care atribuie măsuri paginilor WEB pentru a găsi cele mai importante pagini şi **algoritmul Hubs and authorities** (indecşi şi autorități) care face o evaluare detaliată a importanței paginilor WEB parcurgând următorii 3 paşi:

P1. Crearea unei mulțimi inițiale de pagini (root set). În această fază se face o căutare bazată pe un text, rezultând o mulțime de pagini, folosită pentru pornirea procesului de căutare. Aceste pagini conțin textul folosit în căutare, dar nu conțin neapărat informațiile cele mai autorizate aşteptate de utilizator, deoarece sursele cele mai autorizate pe subiectul cerut nu folosesc frecvent topicul respectiv.
Exemplu, pentru cuvântul harvard ne aşteptăm ca printre paginile selectate să găsim şi pagina autorizată **www.harvard.edu**, ceea ce nu se întâmplă în realitate. Este totuşi posibil ca unele pagini din cele selectate să aibă linku-uri spre pagina autorizată **www.harvard.edu**.
P2. Se extinde setul de pagini selectate anterior (root set) prin pagini la care avem link-uri de la cele din root set sau de la care avem link-uri spre cele din root set. Se obține o mulțime candidat de pagini.
P3. Se ordonează mulțimea candidat de pagini după gradul de autorizare privind topicul căutat şi în funcție de legăturile pe care le conține către alte pagini autorizate.

Kleinberg foloseşte cei doi algoritmi mai sus numiți la pasul doi pentru a găsi mulțimea candidat de pagini.

Algoritmul PageRank [2] determină importanța fiecărei pagini. Pentru aceasta se construieşte matricea pătratică WEB de dimensiune egală cu numărul paginilor WEB. În această matrice fiecărei pagini îi corespunde câte o coloană.
Dacă pagina j are n link-uri către alte pagini, atunci

$$WEB(i,j)=\begin{cases}\dfrac{1}{n}, & \text{pagina i este succesorul paginii j}\\ 0, & \text{altfel}\end{cases}$$

Definiția 2. *Se numeşte importanță a paginii i, notată PR(i) suma importanțelor paginilor care referă pagina i, scalată eventual cu o constantă nenulă.*

Fie x vectorul importanțelor paginilor de pe WEB. Din definiție rezultă relația:

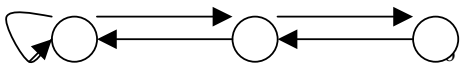
$x = c \cdot WEB \cdot x$, unde c este factor de scală.

Dacă punem această relație sub forma:

$$WEB \cdot x = \lambda x, \text{ unde } \lambda = \frac{1}{c},$$

rezultă că x este vector propriu corespunzător unei valori proprii λ a matricei WEB.
De aici rezultă că importanța unei pagini *i* este componenta de ordin *i* a vectorului propriu corespunzător valorii proprii principale a matricei WEB.
Importanța unei pagini *i* se poate interpreta ca probabilitatea ca un navigator de pe WEB, pornind de la o pagină arbitrară şi urmând linkuri alese aleator din fiecare pagină accesată, să ajungă după o serie de link-uri la pagina *i*.

Exemplul 3. Presupunem 3 pagini WEB *a,b,c* legate între ele ca în figura următoare, unde $x \rightarrow y$, indică un link de la pagina *x* la pagina *y*.



Matricea WEB este:

$$WEB = \begin{array}{c|ccc} & a & b & c \\ a & \frac{1}{2} & 0 & \frac{1}{2} \\ b & 0 & 0 & \frac{1}{2} \\ c & \frac{1}{2} & 1 & 0 \end{array}$$

Prima coloană din WEB spune că pagina *a* îşi împarte importanța cu pagina *c*, a treia coloană spune că pagina *c* transferă importanța ei în mod egal paginilor *a* şi *b* iar coloana a doua spune că pagina *b* transmite toată importanța ei paginii *c*.

Rezolvând ecuația: $\det(WEB - \lambda I_3) = 0$ găsim valorile proprii 1 şi $\frac{-1 \pm \sqrt{5}}{4}$.

Vectorul propriu principal se obține rezolvând ecuația omogenă: $WEB \cdot x = \lambda x$ pentru $\lambda = 1$, unde $x = (a,b,c)^T$. Rezolvând sistemul în necunoscute *a,b,c* găsim rezultatul $a = c = 2b$. Acest rezultat spune că paginile *a* şi *c* au aceiaşi importanță, care este de două ori mai mare decât importanța paginii *b*.

Algoritmul PageRank se poate formaliza astfel:

P1. Construieşte matricea patratică WEB de ordin *n*, unde *n* este numărul paginilor de pe WEB.

P2. Rezolvă ecuația $\det(WEB - \lambda I_3) = 0$, şi determină valorile proprii $\lambda_1 \lambda_2, ..., \lambda_n$.

P3. Rezolvă ecuația vectorială $WEB \cdot x = \lambda x$, unde λ este valoare proprie reală maximă.

Vectorul propriu x conține pe poziția *i, i* = 1,…,*n*, importanța paginii de ordin *i* , adică *PR(i)* = x_i .

Analizând acest algoritm vom observa că se poate exprima importanța unei pagini i în funcție de importanțele paginilor care au link-uri spre pagina i sub forma:

$PR(i) = \sum_{j \in B(i)} \frac{PR(j)}{N_j}$, unde $B(i)$ este mulțimea paginilor ce au link-uri spre pagina *i*, iar N_j este numărul link-urilor paginii *j* spre alte pagini de pe WEB.

Exemplul 4. Reluăm exemplul 3 în care am găsit $PR(a) = PR(c) = 2PR(b)$.

Pentru pagina *c* avem $B(c) = \{a,b\}$, $N_a = 2$, $N_b = 1$.

Astfel $\sum_{j \in B(c)} \frac{PR(j)}{N_j} = \frac{PR(a)}{N_a} + \frac{PR(b)}{N_b} = \frac{2PR(b)}{2} + \frac{PR(b)}{1} = 2PR(b) = PR(c)$

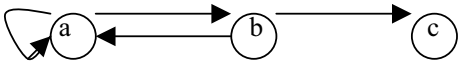
şi *a* astfel formula se verifică pentru pagina *c*.

Prin modelarea WEB-ului ca un graf apar două probleme limită:

1. O *problemă* numită *dead-end* care corespunde situației în care o pagină care nu are succesori nu are către cine să-şi transmită importanța şi astfel această importanță se scurge în Internet.
2. O *problemă* numită *spider traps* care corespunde situației în care un grup de una sau mai multe pagini care nu au legaturi către pagini din afara grupului vor acumula eventual toata importanța din Internet.

Vom exemplifica aceste situații limită:

Exemplul 5. (*dead end*). Reluăm exemplul de WEB prezentat în exemplul 3 în care presupunem că pagina b elimină toate legăturile cu celelalte pagini. Noul WEB are forma din figura următoare:



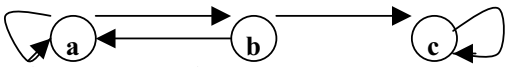
În acest caz matricea WEB este:

$$WEB=\begin{array}{c|ccc} & a & b & c \\ a & \frac{1}{2} & 0 & \frac{1}{2} \\ b & 0 & 0 & \frac{1}{2} \\ c & \frac{1}{2} & 0 & 0 \end{array}$$

Ecuația $\det(WEB-\lambda I_3)=0$ ne dă valorile proprii $\lambda_1=\frac{1}{2},\lambda_2=0,\lambda_3=-\frac{1}{2}$.

Rezolvând sistemul omogen $WEB\cdot(a,\ b,\ c)^T=\frac{1}{2}(a,\ b,\ c)^T$ găsim soluția $a=b=c=0$ şi deci toate paginile au importanța nulă, adică toată importanța se scurge în Internet.

Exemplul 6. (*spider traps*) Reluăm exemplul 3 în care presupunem că pagina b hotărăşte să aibă legături doar către ea însăşi de acum înainte, devenind astfel o capcană. Noul WEB are forma din figura următoare:



Matricea WEB are forma:

$$WEB=\begin{array}{c|ccc} & a & b & c \\ a & \frac{1}{2} & 0 & \frac{1}{2} \\ b & 0 & 1 & \frac{1}{2} \\ c & \frac{1}{2} & 0 & 0 \end{array}$$

Ecuația $\det(WEB-\lambda I_3)=0$ ne dă valorile proprii $\lambda_1=1,\lambda_{2,3}=\frac{1\pm\sqrt{5}}{4}$.

Rezolvând sistemul omogen $WEB \cdot (a,\ b,\ c)^T = (a,\ b,\ c)^T$ găsim soluția $a = c = 0$ şi b arbitrar.

Astfel paginile *a* şi *c* au importanța nulă iar *b* acumulează toată importanța din Internet.

Pentru a combate efectele *dead end* şi *spider trap* GOOGLE nu foloseşte direct matricea WEB ci taxează fiecare pagină cu o fracțiune $q \in (0,1)$ din importanța sa curentă şi distribuie importanța taxată în mod egal tuturor paginilor.

Cu aceasta ecuația vectorială a importanței paginilor pentru exemplul anterior se scrie sub forma: $(a,\ b,\ c)^T = q \cdot WEB \cdot (a,\ b,\ c)^T + (1-q,\ 1-q,\ 1-q)^T$.

Pornind de la $a = b = c = 1$ şi calculând iterativ vectorul $(a,\ b,\ c)^T$ pentru *q*=0.8 găsim după 4 iterații soluția: $a = \frac{7}{11}, b = \frac{21}{11}, c = \frac{5}{11}$, ceea ce reprezintă o distribuție mult mai rezonabilă a importanței paginilor.

Chiar dacă algoritmul PageRank este folosit de multe motoare de căutare printre care şi GOOGLE, rămâne totuşi o problemă de rezolvat şi anume: cu cât o pagină este mai importantă cu atât mai mult celelalte pagini tind să aibă legaturi cu aceasta. Astfel importanța unei pagini se propagă în mod uniform la toate paginile spre care pagina respectivă are link-uri.

Este evident însă că pentru o pagină WEB anumite link-uri sunt mai importante decât altele şi astfel importanța unei pagini nu poate fi distribuită uniform spre toate paginile cu care ea are legături.

Se rezolvă această problemă prin algoritmul PageRank ponderat.

Pentru pagina *i*, numim *outlink* o pagină către care *i* are o legătură şi *inlink* o pagină de la care există un link către pagina *i*.

Notăm cu $W^{in}(i, j)$ importanța legăturii de la inlink-ul *j* la pagina *i* şi cu $W^{out}(i, j)$ importanța legăturii de la pagina i la outlink-ul său j .

Cu aceste notații avem: $W^{in}(j,i) = \frac{I_i}{\sum_{p \in R(j)} I_p}$ şi $W^{out}(i,j) = \frac{O_i}{\sum_{p \in R(j)} O_p}$ unde:

I_l reprezintă numărul inlink-urilor paginii *l*

O_l reprezintă numărul outlink-urilor paginii *l*

R(j) reprezintă mulțimea paginilor care au link-uri spre pagina *j*.

Folosind PageRank, importanța inlink-urilor şi a outlink-urilor, se poate stabili formula următoare:

$$PR(i) = q \cdot \sum_{j \in B(i)} PR(j) \cdot W^{in}(j,i) \cdot W^{out}(i,j) + (1-q)$$

unde $q \in (0,1)$ este taxa folosită de GOOGLE pentru a combate efectele de *dead end* şi *spider trap* (de obicei *q*=0.8).

4. **INDECŞI ŞI AUTORITĂȚI PE WEB**

Se definesc aceşti termeni într-un mod mutual recursiv [3].

Definiția 3. *Indexul este o pagină care nu furnizează informații ci spune unde se găsesc informații , iar autoritatea este o pagină care oferă informații despre un subiect.*
Astfel, un index conține legături către diverse autorități, iar o autoritate este referită de anumiți indecşi.

Pentru a defini gradul de indexare şi respectiv gradul de autoritate al unei pagini atribuim fiecărui site o matrice pătrată A de dimensiune n (n este numărul paginilor de pe site), definită astfel:

$$A_{i,j} = \begin{cases} 1, & \text{există cel puțin un link de la pagina i la pagina j} \\ 0, & \text{altfel} \end{cases}$$

Din această definiție a lui A rezultă:
- Linia i a lui A este vector de selecție pentru paginile pe care le referă i
- Coloana i a lui A este vector de selecție pentru paginile care referă pagina i.

Fie a şi h doi vectori de dimensiune n în care a_i reprezintă gradul de autoritate al paginii i iar h_i reprezintă gradul de indexare al paginii i.

Definiția 4. *Definim gradul de indexare al unei pagini i ca fiind suma gradelor de autoritate ale tuturor paginilor referite de i, scalată cu ponderea* λ.
Astfel $h = \lambda \cdot A \cdot a$ (1)

Definiția 5. *Definim gradul de autoritate al paginii i ca fiind suma gradelor de indexare ale tuturor paginilor care referă pagina i, scalată cu ponderea* μ.

Astfel $a = \mu \cdot A^T \cdot h$ (2)

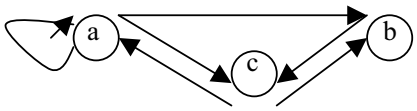
Combinând relațiile (1) şi (2) obținem:

$$a = \lambda\mu \cdot A^T A a \quad (3)$$

$$h = \lambda\mu \cdot A A^T \cdot h \quad (4)$$

Rezultă că a şi h sunt vectori proprii corespunzători valorilor proprii principale ale matricilor $A^T A$ şi respectiv AA^T.

Exemplul 8. [..] Considerăm un site compus din 3 pagini *a,b,c* legate între ele ca în figura următoare:



Matricea asociată legăturilor dintre pagini este:

$A=$

	a	b	c
a	1	1	1
b	0	0	1
c	1	1	0

$A^T=$

	a	b	c
a	1	1	1
b	1	0	1
c	1	1	0

=>

$AA^T=$

	a	b	c
a	3	1	2
b	1	1	0
c	2	0	2

şi $A^TA=$

	a	b	c
a	2	2	1
b	2	2	1
c	1	1	2

Dacă luăm factorii de scală unitari: $\lambda = \mu = 1$ şi vectorii $a = (a_a, a_b, a_c)^T$, $h = (h_a, h_b, h_c)^T$ avem:

Ecuația $\det(AA^T - \lambda i_3) = 0$ dă valorile proprii $\lambda_1 = 3+\sqrt{3}, \lambda_2 = 3-\sqrt{3}, \lambda_3 = 0$.

Rezolvând sistemul omogen: $AA^T \cdot (h_a, h_b, h_c)^T = (3+\sqrt{3})(h_a, h_b, h_c)^T$ găsim soluția: $h = h_a \cdot (1, 2-\sqrt{3}, \sqrt{3}-1)^T$. Se observă că pagina *a* are gradul de indexare mai mare decât gradul de indexare al celorlalte pagini. Mai concret gradul de indexare al paginii *a* este de $2+\sqrt{3} \approx 3,73$ ori mai mare decât gradul de indexare al lui *b* şi de $\frac{\sqrt{3}+1}{2} \approx 1.36$ ori mai mare decât gradul de indexare al paginii *c*.

Ecuația $\det(A^TA - \lambda i_3) = 0$ ne conduce la aceleaşi valori proprii $\lambda_1 = 3+\sqrt{3}, \lambda_2 = 3-\sqrt{3}, \lambda_3 = 0$.

Rezolvând sistemul omogen: $A^TA \cdot (a_a, a_b, a_c)^T = (3+\sqrt{3})(a_a, a_b, a_c)^T$ găsim soluțiile: $a_a = a_b$ şi $a_c = a_a(\sqrt{3}-1)$, adică $a = a_a(1, 1, \sqrt{3}-1)$.

Se observă că paginile a şi b au acelaşi grad de autoritate, grad care este mai mare decât gradul de autoritate al paginii c, fiind în raportul de $\frac{1}{\sqrt{3}-1} = \frac{\sqrt{3}+1}{2} \approx 1.36$.

Astfel gradul de autoritate al paginilor a şi b este de 1.36 ori mai mare decât gradul de autoritate al paginii c.

În concluzie, utilizând măsurile pentru importanța paginilor, gradul de autoritate şi gradul de indexare , căutarile pe WEB se pot face mai simplu.

5. ALGORITM PENTRU CALCULUL GRADELOR: PAGINILOR DINTR-UN BLOC, A BLOCURILOR ŞI A PAGINILOR DE PE WEB

5.1 Preliminarii [6]

Am văzut în cele precedente că un link de la pagina *i* la pagina *j* de pe WEB indică faptul că pagina *j* este importantă pentru *i*.

Dacă presupunem WEB-ul ca un digraf *G* în care nodurile grafului sunt paginile de pe WEB, iar arcele indică legăturile între pagini, atunci gradul unei

pagini *i* (importanţa paginii *i*) poate fi definit ca fiind probabilitatea ca pagina *i* să fie vizitată (să fie referită) din alte pagini. Mai exact, gradul unei pagini referită din pagina *i* este $\frac{1}{d^+(i)}$, unde $d^+(i)$ este gradul exterior al paginii *i*. Considerăm lanţul Markov indus de parcurgerea aleatoare a grafului *G*, lanţ ale cărui stări sunt nodurile grafului , iar matricea trecerilor directe dintre stări, *P* dată prin $p_{ij} = \frac{1}{d^+(i)}$.

Pentru ca *P* să fie o matrice de probabilităţi de trecere este necesar să nu aibă linii nule, ceea ce înseamnă că pe WEB să nu existe pagini fără link-uri în exterior. Cum acest lucru nu este adevărat se poate transforma *P* într-o matrice de probabilităţi de tranzitie *P'* prin adăugarea unei mulţimi de link-uri din nodurile lui *G* cu gradul exterior nul către alte noduri din *G*. Astfel fiecare pagină de pe WEB va avea cel puţin un link către o altă pagină.

Pentru a defini riguros matricea *P'* considerăm *v* un vector n-dimensional (*n* este numărul tuturor nodurilor grafului *G*) vector ce reprezintă o probabilitate uniform distribuită pe toate nodurile, adică $v_i = \frac{1}{n}$, $\cdot i = 1,...,n$.

Fie *d* un vector n-dimensional ce identifică nodurile cu gradul exterior nul, adică $d_i = \begin{cases} 1, & d^+(i) = 0 \\ 0, & \text{altfel} \end{cases}$

Construim $D = dv^t$ şi apoi $P' = P + D$. Efectul lui *P* asupra lui *D* este de a modifica probabilităţile de tranziţie astfel încât un utilizator care vizitează o pagină fără link-uri în exterior să ajungă în mod aleator la o pagină oarecare, folosind distribuţia dată de *v*.

Se ştie că lanţul Markov considerat are o distribuţie de probabilitate unică dacă *P'* nu este periodică şi este ireductibilă.

Evident, pentru graful *G* asociat WEB-ului, matricea *P'* nu este periodică.

In plus *P'* este ireductibilă dacă şi numai dacă *G* este tare conex, ceea ce pentru WEB nu se întâmplă. Pentru a face *P'* ireductibilă vom adăuga arce cu probabilităţi de trecere asociate extrem de mici astfel încât de la fiecare nod să se poată trece la orice alt nod al grafului. Evident noul digraf este tare conex.

Pentru a construi matricea ireductibilă Markov *P''* asociată acestui digraf luăm e un vector n-dimensional cu toate componentele egale cu 1 şi construim:

$E = e \cdot v^t$ şi apoi $P'' = cP + (1-c)E$ unde $c \in (0,1)$.

Efectul matricei *E* este următorul: în orice moment un utilizator care vizitează o pagină oarecare cu probabilitatea $1-c$ va putea trece la o pagină aleatoare de pe WEB, pagină aleasă în funcţie de distribuţia probabilităţii dată de *v*.

Aceste salturi artificiale datorate lui E se numesc *teleportări*.

In experimente valoarea lui *c* se alege 0.85.

Vectorul *v* se numeşte *vector personalizator* deoarece putem alege *v* neuniform încât *D* şi *E* să adauge tranziţii artificiile cu probabilităţi neuniforme şi în concluzie vectorul gradelor paginilor (importanţelor paginilor) să fie reglat astfel încât să fie preferate anumite tipuri de pagini.

Din cele de mai sus, se observă că dacă un utilizator se află pe pagina *i* de pe WEB, distribuţia de probabilitate a unei tranziţii din nodul *i* este data de linia *i* a lui *P''*.

De asemenea, se observă că arcele suplimentare introduse de către *D* şi *E* nu trebuie efectiv materializate şi astfel matricele $A=P''^{t}$ şi *P''* rămân tot matrice rare.

Fie $x^{(0)}$ vectorul distribuţiei de probabilitate ca un utilizator să se afle pe WEB la momentul 0 (x_i^0 este probabilitatea ca un utilizator să se afle la momentul 0 la pagina *i*). Atunci distribuţia probabilităţii ca un utilizator să se afle pe WEB la momentul k este:

$$x^{(k)} = A^k x^{(0)}.$$

Distribuţia staţionară unică a lanţului Markov este:

$$\lim_{k\to\infty} x^{(k)} = \lim_{k\to 0} A^k x^{(0)},$$

care este independentă de distribuţia iniţială $x^{(0)}$.

Această distribuţie este vectorul propriu principal al matricei $A = P''^{t}$, care ne va da gradul (importanţa) paginilor WEB-ului.

Algoritmul de calcul al gradelor paginilor de pe WEB, calculează vectorul propriu principal al matricei *A* pornind cu o distribuţie iniţială uniformă $x^{(0)} = v$ şi calculând succesiv $x^{(k+1)} = A \cdot x^{(k)}$ până la convergenţă.

Pseudocodul acestui algoritm este următorul:

funcţia pageRank (G, $x^{(0)}, v$)

{

calculează P prin: $p_{ij} = \dfrac{1}{d^+(i)}$

repeat

$x^{(k+1)} = cP^t x^{(k)}$

$w = \left\| x^{(k)} \right\|_1 - \left\| x^{(k+1)} \right\|_1$

$x^{(k+1)} = x^{(k+1)} + wv$

$\delta = \left\| x^{(k+1)} - x^{(k)} \right\|_1$

until $\delta < \varepsilon$

return $x^{(k+1)}$

}

In algoritm graful WEB-ului este dat prin matricea sa de adiacenţă *G*. Structura de blocuri a WEB-ului sugerează un algoritm rapid pentru calculul gradelor

blocurilor. Se calculează mai întâi pentru fiecare host un vector local al gradelor paginilor din interiorul host-ului. Vectorii locali ai host-urilor sunt folosiți pentru a aproxima vectorul global al blocurilor care va fi vector de start pentru calculul gradelor blocurilor de pe WEB.
Paşii algoritmului sunt următorii:
P1. Se separă WEB-ul în blocuri corespunzătoare diferitelor domenii
P2. Se calculează vectorul local al gradelor paginilor fiecărui bloc folosind funcția pageRank
P3. Se estimează importanța relativă a fiecărui bloc
P4. Se împarte vectorul local al gradelor paginilor la gradul întregului bloc
P5. Se concatenează vectorii locali obținuți la P4. obținând o aproximare a vectorului global *z* al gradelor blocurilor.
P6. Se foloseşte *z* ca vector de start pentru calculul gradelor blocurilor.

Pentru a formaliza acest algoritm folosim următoarele notații:
i,j,k, … reprezintă site-uri individuale
I,J,K,… reprezintă blocuri de pe WEB
n_J este numărul de pagini al blocului *J*.
Graful unui bloc *J* este dat de matricea G_{JJ} pătratică de dimensiune n_j , submatrice a matricei WEB-ului G.
$i \in J$ spune că pagina i se afla în blocul *J*.
Vectorul local al gradelor paginilor din blocul *J* se notează I_J şi se obține aplicând algoritmul PageRank blocului *J* ca şi cum acesta ar fi întregul WEB, neexistând legături cu alte blocuri.

Formal $I_J = PageRank(G_{JJ}, s_J, v_J)$,

unde vectorul de start s_J este vector de probabilitate uniformă peste paginile din blocul *J*, adică $(s_J)_i = \frac{1}{n_J}, \forall i = 1,...,n_J$, iar vectorul de personalizare v_J este vector n_J dimensional cu toate elementele nule, mai puțin elementul ce corespunde nodului rădăcină al blocului *J*, care este 1.

Au fost realizate o serie de experimente care au evidențiat faptul că o concatenare a vectorilor locali ai gradelor paginilor conduce la un vector global de start pentru algoritmul PageRank, mai bun decât un vector global uniform şi de asemenea că ordinea relativă a paginilor din interiorul unui host, dată de valorile locale ale gradelor paginilor, este aceiaşi cu ordinea intra-host-uri dată de valorile globale pentru gradul paginilor.

In acest scop s-au calculat vectorii locali I_J ai gradelor paginilor dintr-un domeniu J pentru fiecare host din J. S-a calculat şi vectorul global al gradelor paginilor *x* folosind algoritmul PageRank cu vectorul personalizator v care este o distribuție uniformă peste nodurile rădăcină . In final s-au comparat valorile locale obținute pentru gradul paginilor dintr-un host dat cu valorile globale ale paginilor din acel host.

In mod concret s-au luat elementele ce corespund paginilor din host-ul J din vectorul global x formând vectorul g_J care s-a normalizat încât suma elementelor sale să fie 1. Astfel $g_J = \frac{x(j \in J)}{\|x(j \in J)\|_1}$.

Vectorul g_J se numeşte segment global normalizat.

Eroarea pentru vectorul local al gradelor I_J este $\|I_J - g_J\|_1$. Aceasta se compară cu eroarea pentru un vector uniform :

$(v_J)_i = \frac{1}{n_J}, (i = 1, 2, ..., n_J)$ care este $\|v_J - g_J\|_1$.

In experiment s-a obținut relația: $\|I_J - g_J\|_1 < \|v_J - g_J\|_1$, relația spune că vectorii locali ai gradelor paginilor sunt mult mai apropiați de segmentele globale normalizate ale paginilor decât sunt vectorii uniformi.

Pentru a compara ordinea relativă a paginilor din interiorul unui host cu ordinea intra-host-uri se măsoară distanța medie între vectorii I_J si g_J, notată Dist, astfel:

- se consideră două liste de lungime p, σ_1 si σ_2 parțial ordonate;
- se notează U intersecția lui σ_1 cu σ_2 şi δ_1=U-σ_1, δ_2=U-σ_2;
- se consideră $\sigma_1^{'}$ extensia lui σ_1 concatenată în continuare cu δ_1 şi $\sigma_2^{'}$ extensia lui σ_2 prin concatenarea acesteia în continuare cu δ_2;
- se defineşte măsura:

$$\text{Dist}(\sigma_1, \sigma_2) = \frac{\left|\{(u,v) \in U \times U / d \neq v \text{ si } \sigma_1^{'} \text{ difera de } \sigma_2^{'} \text{ pe } (u,v)\}\right|}{|U|(|U|-1)}$$

Această măsură ar reprezenta probabilitatea ca $\sigma_1^{'}$ şi $\sigma_2^{'}$ să difere pentru o ordine relativă a unei perechi arbitrare de noduri distincte din U.

Dacă Dist(σ_1, σ_2) are o valoare mică, atunci ordinea dată de gradul local al paginilor este corectă. Aceasta sugerează că majoritatea erorilor din comparația gradelor locale şi globale ale gradelor paginilor provin din calculul greşit al importanței unor pagini pentru fiecare host. Aceste erori pot afecta paginile importante de pe WEB.

In plus, gradul relativ al paginilor din hosturi diferite nu este cunoscut şi de aceea nu se recomandă folosirea gradului local al paginilor pentru a calcula gradul global al paginilor. El se foloseşte doar ca instrument de calcul mai rapid pentru valorile globale ale gradelor paginilor.

Din cele prezentate rezultă o condiție de oprire a calculului gradului local al paginilor. Pentru aceasta la fiecare etapă a algoritmului se calculează Dist între

iterația curentă $I_J^{(k)}$ şi cea precedentă $I_J^{(k-1)}$. Când această valoare a lui Dist se apropie suficient de mult de zero înseamnă că ordinea este corectă şi calculul gradului local al paginilor se poate opri.

5.2 **Calculul importanței relative a blocurilor [6]**

Presupunem că există k blocuri care formează WEB-ul. Pentru a calcula gradul blocurilor formăm graful blocurilor *B* în care fiecare vârf corespunde unui bloc de pe WEB. Ducem un arc între două blocuri dacă fiecare conține câte cel puțin o pagină şi între acestea avem un arc.

Importanța unui arc (*I,J*) dintre blocurile *I* şi *J* este dată de suma importanțelor arcelor de la paginile din blocul *I* la paginile din blocul *J*, arce existente în graful *G* al WEB-ului.

Astfel importanța $B_{I,J}$ a arcului (*I,J*) este dată de formula: $B_{I,J} = \sum_{i \in I, j \in J} A_{i,j} l_i$,

unde *A* este matricea de adiacență a grafului *G*, iar l_i este gradul local al paginii *i* din blocul *I*.

Considerăm *L* o matrice $n \times k$ -dimensională (*n* este numărul paginilor de pe WEB) a valorilor locale pentru gradul paginilor. Coloanele lui *L* sunt vectorii l_j ai gradelor locale. *L* are următoarea formă:

$$L = \begin{matrix} l_1 & 0 & \dots & 0 \\ 0 & l_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & l_k \end{matrix}$$

Considerăm *S* o matrice $n \times k$ -dimensională cu aceiaşi structura ca şi *L* în care toate valorile nenule se înlocuie cu 1.

Atunci matricea *B* a blocurilor are forma: $B = L^T AS$. Se observă că matricea pătratică de ordin *k, B* este o matrice de tranziție în care elementul $B_{I,J}$ reprezintă probabilitatea trecerii din blocul *I* în blocul *J*, $B_{I,J} = \Pr(I \to J)$. Având la îndemână matricea *k*-dimensională *B* putem aplica algoritmul PageRank pe această matrice redusă pentru a calcula vectorul ce dă gradul blocurilor b prin formula:

$$b = \text{pageRank}(B, v_k, v_k),$$

unde v_k este un vector *k*-dimensional uniform, adică $(v_k)_i = \frac{1}{k}$, pentru orice *i*=1,2,…,*k*.

Notăm cu *b* vectorul gradelor blocurilor în care elementul b_J reprezintă gradul blocului *J* şi măsoară importanța relativă a acestui bloc.

In interiorul fiecărui bloc *J* avem gradul local l_J pentru paginile din acest bloc.

Cu acestea putem aproxima gradul global al paginii $j \in J$ ca fiind gradul local l_j al paginii j ponderat cu gradul b_J al blocului din care *j* face parte: $x_j^{(0)} = l_j b_J$.

Matriceal această relație devine: $x^{(0)} = Lb$.

Pentru a arăta că se poate lua $x^{(0)}$ ca vector de start pentru algoritmul PageRank, va trebui să arătăm că suma componentelor sale este 1. Pentru aceasta folosim că suma gradelor locale pentru fiecare domeniu *J* este 1, adică $\sum_{j\in J} l_j = 1$ şi că suma gradelor blocurilor este 1, adică $\sum_j b_J = 1$. Astfel avem:

$$sum(x_j^{(0)}) = \sum_j x_j = \sum_j \sum_{j\in J} x_j = \sum_j \sum_{j\in J} l_j b_J = \sum_j b_J \sum_{j\in J} l_j = 1$$

Pentru a calcula adevăratul vector global al gradelor paginilor de pe WEB, $\bar{x}$ din vectorul ce aproximează gradele paginilor $x^{(0)}$, vom folosi în algoritmul PageRank vectorul $x^{(0)}$ ca vector de start. Astfel x = pageRank(G,$x^{(0)}$,v) unde *G* este matricea de adiacență a grafului WEB iar *v* este o distribuție uniformă peste nodurile rădăcină.

Algoritmul pentru calculul gradului paginilor de pe WEB se poate acum sintetiza sub următoarea formă:

P1. Descompune WEB-ul în *k* blocuri pe domenii şi ordonează lexicographic blocurile după numele drumurilor

P2. Calculează vectorul gradelor locale l_J pentru fiecare bloc *J*

for *J=1,k* do

l_J= *pageRank*(G_{JJ}, s_J v_J)

end

P3. Calculează matricea de trecere între blocuri *B* şi vectorul gradelor blocurilor *b*

$$B = L^T AS$$

$$b = \text{pageRank}(B, v_k, v_k),$$

P4. Găseşte aproximarea $x^{(0)}$ a vectorului *x* ce conține gradele globale ale paginilor prin ponderarea gradelor locale ale paginilor din blocul *J* cu gradul blocului *J*

$$x^{(0)} = Lb$$

P5. Calculeaza *x* folosind $x^{(0)}$ ca vector de start pentru algoritmul PageRank prin relația: *x = pageRank(G,*$x^{(0)}$*,v)*

Vom face acum o scurtă analiză a acestui algoritm.

Algoritmul are o performanță bună deoarece toate blocurile de pe WEB sunt de dimensiuni relativ mici încât graful fiecărui bloc încape în memorie iar vectorul gradelor blocului curent încape în memoria cache a procesorului.

Deoarece întregul graf WEB nu încape în memoria principală, iterațiile pentru calculul gradelor locale necesită mai puțin spațiu pe disc pentru operațiile de intrare/ieşire decât ar fi necesar pentru un calcul global.

Vectorii ce reține gradele locale pentru multe blocuri converg rapid şi astfel calculele pentru acele blocuri se încheie după câteva iterații. Aceasta permite algoritmului să se ocupe mai mult de blocurile care converg încet.

Pasul 2 al algoritmului permite paralelizarea calculului, prin calcularea pe procesoare separate a gradelor locale pentru fiecare bloc. Este necesară o singură sincronizare la sfârşitul pasului 2 când fiecare procesor trimite vectorul I_J către un

procesor care controlează rezultatele şi calculează vectorul global al gradelor paginilor.

Rezultatul final al algoritmului poate fi refolosit pentru o următoare aplicare a acestui algoritm, atunci când se consideră necesară o astfel de operație.

BIBLIOGRAFIE

1. Broder A., Kumar R., Maghoul F., *Graph Structure in the Web*, In Proceedings of the Ninth International World Wide Web Conference, 2000
2. Ullman J., *Data Mining* Lecture Notes (2000-2003)
3. Page L.,Brin S., Motwani R., Winograd T., *The PageRank citation ranking: Bringing order the web*, Stanford Digital Libraries Working Paper, 1998
4. Raghavan S., Garcia-Molina H. , *Representing web graphs*. In Proceedings of the IEEE Intl. Conference on Data Engineering, March 2003
5. Arasu A., *PageRank computation and the structure of the Web: experiments and algorithms*, In Proceedings of the Eleventh International World Wide Web Conference, Poster Track, 2002
6. Kanvar S., Aveliwala T., Manning C., Golub G. , *Exploiting the Block Structure of the WEB for Computing PageRank,* , Stanford University2003

Abstract: In this article we define a couple of measures that facilitate WEB page searching, including: the importance (degree) of a page, the relative degree of a block, the indexing and authority degree of a page, etc. For this we consider the WEB as being a great digraph in which nodes correspond to pages and arcs correspond to links. Also, we consider the WEB as a digraph where nodes are WEB domains and arcs are links between pages on those domains. We compute the defined measures first on a domain level, then intra-domain and then globally, by concatenation, on a WEB level.