UNIVERZITET U NOVOM SADU
PRIRODNO - MATEMATIČKI FAKULTET
DEPARTMAN ZA MATEMATIKU
I INFORMATIKU

Milica Višekruna

Primena teorije Markovljevih lanaca na analizu i sintezu algoritma „PageRank“

MASTER RAD

Mentor:

dr Dušan Jakovetić

Novi Sad, 2019.

Sadržaj

Veliku zahvalnost dugujem svojom mentoru dr Dušanu Jakovetić koji me je svojim predavanjima zainteresovao za matematičko modeliranje i motivisao da i sama istražujem tu oblast. Hvala mu na ideji za master rad i na razumevanju, strpljenju i trudu koji je uložio u njegovu realizaciju. Zahvaljujem se i ostalim članovima komisije, kao i svim profesorima koji su se trudili da prenesu znanje i što više približe apstraktne pojmove matematike, posebno dr Mileni Kresoja za svu pomoć tokom master studija.

Studiranje ne bi prošlo isto bez mojih koleginica koje su bile uvek tu da učimo zajedno, bodrimo jedna drugu i zajedno slavimo sve uspehe kao svoje - hvala im na tome.

Hvala Iliji i mojim prijateljima što su verovali u mene i pružali mi bezgranično razumevanje i ohrabrenje.

Na kraju, najveću zahvalnost dugujem svojoj porodici koja je bila podrška tokom celog studiranja.

Novi Sad, 2019. Milica Višekruna

1.Uvod

„Matematičke strukture spadaju među najdivnija otkrića ljudskog uma.

Njihova veličina je u njihovoj velikoj moći metafore i objašnjavanja realnosti.“

Douglas Hofstadter (Meta Mathematical Themas, 1985)

U odabiru teme za rad presudno je bilo da bude nešto zanimljivo, aktuelno i da mogu kroz rad da prikažem primenu matematike u svakodnevnom životu.

Pretraživanje interneta svi vršimo svakodnevno. Informacije su dostupnije nego ikada i njihova količina se širi ekspanzivno. Odavno se ne postavlja pitanje da li ćemo naći ono što tražimo, već koja u moru stranica koje sadrže tražene informacije je najpribližnija onome što tražimo. Kada nam pretraživač koji koristimo izbaci rezultate pretrage, uglavnom očekujemo da ćemo na prvih par stranica dobiti tražene informacije. Ali kako pretraživač zna šta da nam ponudi i kojim redom? Šta je to što neke stranice čini boljim od ostalih? Na ova pitanja ćemo delimično odgovoriti u ovom radu (delimično jer PageRank jeste važan faktor u ovom određivanju, ali ne i jedini).

Pre nego što počnem priču o samom algoritmu, u odeljku matematičke osnove ću navesti relevantne definicije i teoreme iz oblasti: stohastički procesi, lanci Markova i teorija grafova i navešću primere koji ih bliže objašnjavanju. Za prve dve oblasti su od literature korišćene stavke [8] I [9] navedene u literaturi, dok je za teoriju grafova korišćena stavka [10].

U narednom odeljku je objašnjeno šta je veb (eng. web, www), a zatim opisano kako je izgledalo pretraživanje veba nekad i šta se promenilo od tad, a u ovom odeljku poslužila je literatura [3] I [4].

Potom kreće priča o samom algoritmu. Prvih 7 stavki navedenih u literaturi je korišćeno u ovom delu. Krenućemo od osnovnih karakteristika i istorijata. U poglavlju Računanje biće predstavljena prvobitna ideje, a zatim i naredne koje su bile poboljšane verzija PageRanka. Računanje je ilustrovano primerima, predstavljen je matematički model u vidu formiranja sistema jednačina i metoda za njihovo rešavanje i ukazano je na probleme do kojih može doći, kao i kako se oni mogu prevazići. Nakon toga biće objašnjeno šta je personalizovani PageRank i kako se dolazi do njega, a on će biti korićen u odeljku sa primenom. U odeljku Aproksimacija i ažuriranje predstavljeno je kako su naučnici pokušali da ubrzaju algoritam i povećaju njegovu efikasnost. Ova tema je detaljno objašnjena u [5] gde su sumirani rezultati različitih istraživanja i pristupa. I na kraju dolazimo do problema skladištenja podataka i mogućeg rešenja tog problema koji se mogu pronaći u literaturi [1].

SALSA i HITS su algoritmi slični PageRank-u i naredno poglavlje je posvećeno njima - prvo ću ih objasniti pojedinačno, a zatim ću ih uporediti sa Pagerank algoritmom. Ovde je ponovo korišćena stavka [5].

Naredno poglavlje je posvećeno primeni globalnog i personalizovanog PageRanka na sisteme preporuke. Konkretnije, na primerima sistema za preporuku filmova i primeru socijalne mreže pokazano je kako se ideja o preporučivanju iz PageRanka može preneti na druge sisteme. Literatura koja je uglavnom korišćena u ovom poglavlju se nalazi pod brojem [2].

Za kraj ću je dat još jedan osvrt na temu i mišljenje o pravcu daljih istaživanja na ovu temu.

2. Matematičke osnove

2.1.Stohastički procesi

Stanje nekih sistema moguće je opisati pomoću jedne ili više veličina, u zavisnosti od parametra koji odgovara, a da se zavisnost između odgovarajućeg parametra i postojećih veličina ne može tačno odrediti. U velikom broju slučajeva ta zavisnost se potčinjava statističkim zakonima koji omogućavaju da se odrede verovatnoće realizacija posmatranih veličina. Odnosno možemo reći da vrednosti posmatrane veličine ili veličina nisu unapred određene, već predstavljaju slučajne (stohastičke) veličine u zavisnosti od odgovarajućih parametara. Skup realizacija određene slučajne veličine možemo posmatrati kao slučajnu veličinu koja se menja u vremenu tj. kao slučajni proces.

Zamislimo da u svakom vremenskom trenutku t vremenskog intervala I posmatramo neku karakteristiku X određenog fizičkog sistema koja je slučajna veličina. Dakle, $X(t)$ je neka slučajna promenljiva za svako $t \in I$. Tada na skup svih slučajnih promenljivih $X(t)$, $t \in I$ možemo gledati kao na slučajnu veličinu koja se menja u vremenu, odnosno dobijamo jednu slučajnu funkciju vremena. Ako je interval koji posmatramo $I = \mathbb{Z}$ ili $I = \mathbb{N}$ tada se posmatra stohastički proces sa diskretnim vremenom, a kada je $I = \mathbb{R}$ ili $I = \mathbb{R}^+$ tada se radi o stohastičkom procesu sa neprekidnim vremenom.

Definicija 1. Stohastički proces $\{X(t), t \in I\}$ je familija slučajnih promenljivih definisanih na istom prostoru verovatnoća (Ω, $\mathcal{F}$, P) gde je I tzv. parametarski skup stohastičkog procesa. Kako su slučajne promenljive iz definicije 4. realne ($X : \Omega \to \mathbb{R}^d$) onda je i stohastički proces koji one čine realan. $\mathbb{R}^d$ se može nazvati i skupom stanja stohastičkog procesa.

Kao što smo napomenuli, parametar koji nas interesuje za teoriju redova čekanja u opštem slučaju zapisujemo kao$\{X(t), t \in [t_0, T]\}$ pri čemu je dozvoljen izbor $t_0 = -\infty$, $T = \infty$.
Svaki stohastički proces je funkcija dve promenljive, ω i t, ali se u zapisu umesto $\{X(t, \omega), \omega \in \Omega,\ t \in [t_0, T]\}$ najčešće koristi samo $\{X(t), t \in [t_0, T]\}$. Najčešće se koristi oznaka $X(t)$ ili X_t.

Kada posmatramo stohastički proces :

- Za fiksirano $t \in [t_0, T]$, dobijamo slučajnu promenljivu na prostoru (Ω, $\mathcal{F}$, P)
- Za fiksirano $\omega \in \Omega$ dobijamo funkciju vremena koju nazivamo staza(realizacija ili trajektorija) stohastičkog procesa.

Definicija 2. Stohastički procesi $\{X_t\}_t$ i $\{Y_t\}_t$ koje uzimaju vrednosti iz istog procesa stanja su stohastički ekvivalentni ako $P(X_t \neq Y_t) = 0$ za svako $t \in T$. Ako su $\{X_t\}_t$ i $\{Y_t\}_t$ stohastički ekvivalentni tada se može reći da je $\{X_t\}_t$ verzija $\{Y_t\}_t$ (i obratno).

Konačno-dimenzionalne raspodele stohastički ekvivalentnih procesa se poklapaju. Međutim, ekvivalentni procesi mogu imati potpuno drugačija analitička svojstva.

Uzmimo primer procesa $X_t(\omega) = 0$ i $Y_t(\omega) = \begin{cases} 0, & \omega \neq t \\ 1, & \omega = t \end{cases}$ koji su ekvivalentni jer se poklapaju svuda osim u jednoj tački, a njihove trajektorije imaju drugačija svojstva neprekidnosti tj. trajektorije procesa X_t su svuda neprekidne dok trajektorije procesa Y_t imaju prekid u jednoj tački.

U opštem slučaju nije dovoljno da poznajemo samo zakon jednodimenzionalne raspodele, da bi smo poznavali ceo proces neophodno je poznavati konačno-dimenzionalne raspodele stohastičkog procesa.

Definicija 3. Konačno-dimenzionalne raspodele stohastičkog procesa $\{X(t), t \in [t_0, T]\}$ su date sa:

$$F_t(x) = F_1(x) = P\{X(t) < x\}$$

$$F_{t_1,t_2}(x_1, x_2) = F_2(x_1, x_2) = P\{X(t_1) < x_1, X(t_2) < x_2\}$$

$$F_{t_1,t_2,t_3}(x_1, x_2, x_3) = F_3(x_1, x_2, x_3) = P\{X(t_1) < x_1, X(t_2) < x_2, X(t_3) < x_3\}$$

$$\vdots$$

$$F_{t_1,t_2\ldots t_n}(x_1, x_2, .., x_n) = F_n(x_1, x_2, .., x_n) = P\{X(t_1) < x_1, X(t_2) < x_2, \ldots, X(t_n) < x_n\}$$

$$\vdots$$

gde su $t, t_1, t_2, \ldots, t_n, \ldots \in [t_0, T]$ i $x, x_1, x_2, .., x_n, \ldots \in \mathbb{R}^d$.

Konačno-dimenzionalne raspodele zadovoljavaju dva uslova:

i. Uslov simetrije: za svaku permutaciju $\{i_1, \ldots, i_n\}$ skupa brojeva od $\{1, \ldots, n\}$ važi:
$F_{t_{i_1}\ldots t_{i_n}}(x_1, x_2, .., x_n) = F_{t_1,t_2\ldots t_n}(x_1, x_2, .., x_n)$

ii. Uslov saglasnosti: za $m < n$ i proizvoljne $t_1, t_2, \ldots, t_m, t_{m+1}, \ldots\ t_n \in [t_0, T]$ važi:
$F_{t_1,\ldots,t_m,t_{m+1}\ldots t_n}(x_1, .., x_m, \infty, \infty, \ldots \infty) = F_{t_1,\ldots,t_m}(x_1, .., x_m)$

Uglavnom u praksi imamo slučaj da nemamo familiju slučajnih promenljivih na nekom prostoru verovatnoća već njihove konačno-dimenzionalne raspodele koje zadovoljavaju dva navedena uslova.

Teorema1.(*Fundamentalna teorema Kolmogorova)*

Za svaku familiju raspodela koja zadovoljava uslove i. i ii. postoji prostor verovatnoća i na njemu definisan stohastički proces X_t čije su to konačno-dimenzionalne raspodele.

Svojstva stohastičkih procesa**:**

Definicija 4. Srednja vrednost procesa X_t je:

$$m_x(t) = m(t) = E(X_t).$$

Definicija 5. Autokovarijansna (korelaciona) funkcija stohastičkog procesa X_t je:

$$K_x(t,s) = K(t,s) = E(X_t, X_s) - m(t)m(s), \qquad t,s \in [t_0, T].$$

Definicija 6. Disperzija stohastičkog procesa X_t je:

$$D_x(t) = D(t) = K_x(t,t) = E(X_t^2) - \big(m(t)\big)^2.$$

Definicija 7. Koeficijent korelacije stohastičkog procesa X_t je:

$$\rho_x(t,s) = \rho(t,s) = \frac{K_x(t,s)}{\sqrt{D_x(t)D_x(s)}}.$$

2.2. Lanci Markova

Lanac Markova je diskretni Markovljev slučajni proces. Najkraće objašnjeno – imati svojstvo Markova znaci da buduće stanje zavisi samo od trenutnog, a ne i od prošlih. U svakom trenuntku (na osnovu date raspodele verovatnoća) sistem moze zadržati trenutno stanje ili ga promeniti. Promenu stanja nazivamo prelazom, a verovatnoću promene stanja nazivamo verovatnoćom prelaza.

Definicija 8. Niz slučajnih promenljivih na istom prostoru verovatnoća (Ω, *F* , P) i sa istim skupom stanja $S = \{x_1, x_2, \dots\}$ zove se ***Lanac Markova***, ako za proizvoljne $r \in N$ i $n > k_1 > k_2 > \cdots > k_r$ *važi :*

$$P\{X_n = x_n \mid X_{k_1} = x_{k_1}, \dots, X_{k_r} = x_{k_r}\} = P\{X_n = x_n \mid X_{k_1} = x_{k_1}\}.$$

Ovo svojstvo se zove Markovsko svojstvo i ono kaže da je verovatnoća da se sistem nađe u stanju x_n u nekom budućem trenutku n zavisi samo od sadašnjeg trenutka k_1, a ne od prošlosti $(k_2, \dots, k_r)$.

Definicija 9. Verovatnoća prelaza za jedan korak iz stanja i u stanje j ako je sistem u trenutku n bio u stanju i definiše se kao:

$$p_{i,j}^{n,n+1} = P\{X_{n+1} = x_j \mid X_n = x_i\}.$$

Ako $p_{i,j}^{n,n+1}$ ne zavisi od vremenskog trenutka *n* kažemo da je lanac homogen i verovatnoća prelaza u jednom koraku se označava sa $p_{i,j}$ i računa se: $p_{i,j} = P\{X_{n+1} = x_j \mid X_n = x_i\}$.

Za verovatnoću prelaza iz i-tog stanja u j-to stanje u *n*-koraka (za homogeni lanac) je $p_{i,j}(n) = P\{X_{n+m} = x_j \mid X_m = x_i\}$ za neko m.

Sada možemo da uvedemo i pojam matrice verovatnoće prelaza homogenog lanca Markova za n koraka, $n \in N$:

$$P_n = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

Dimenzije matrice verovatnoće prelaska zavisi od broja svih mogućih stanja u kojem sistem može da se nađe. Kada je *n* = 1, matricu prelaza obeležavamo sa P i zove se matrica prelaza za jedan korak. Matrica P ima svojstva:

- $p_{ij} \geq 0, \forall i, j \in N$
- $\sum_j p_{ij} = 1, \forall i, j \in S$

U prethodnom delu smo definisali verovatnoće prelaza iz jednog stanja u drugo posle jednog i više koraka kao i matrice verovatnoća prelaska posle jednog i više koraka. Sada ćemo pokazati njihovu vezu i na koji način se može da definiše lanac Markova.

Definicija 10. Markovljevi lanci kod kojih matrica prelaznih verovatnoća $P(n)$ ne zavisi o koraku n se zovu *homogeni Markovljevi lanci*. Tu konstantnu matricu prelaznih verovatnoća označavamo sa P.

Od velikog značaja za izračunavanje verovatnoća prelaza u n koraka su jednačine Čepmen-Kolmogorova:

$$p_{i,j}(n+m) = \sum_k p_{i,k}(n) p_{k,j}(m).$$

Ovaj zapis je u nematričnom obliku.

Čepmen-Kolmogorova proces u matričnom obliku glasi:

$$P_{m+n} = P_n * P_m$$

Ili

$$\begin{aligned} & P_m = P_n * P_{m-n}\,, && m > n \\ & m = 1 && P_1 = P \\ & m = 2 && P_2 = P_1 * P_1 = P * P = P^2 \\ & m = 3 && P_3 = P_2 * P_1 = P^2 * P = P^3 \\ & \vdots \\ & m = n && P_n = P^n \ . \end{aligned}$$

Pretpostavljamo da lanac Markova ima konačno mnogo stanja tj. $S = \{x_1, x_2, \dots, x_m\}$.

Sa $p_i(n)$ označavamo verovatnoću da u trenutku n sistem bude u i-tom stanju: $p_i(n) = P\{X_n = x_i\}$.

Za fiksirano n dobijamo $p_i(0)$ tzv. početnu verovatnoću, i pomoću nje određujemo gde je sistem bio u početnom trenutku.

Početni vektor je:

$$p(0) = [\, p_1(0)\ \ p_2(0) \dots\ p_m(0)].$$

Analogno, k-ti vektor je:

$$p(k) = [\, p_1(k)\ \ p_2(k) \dots\ p_m(k)].$$

Pa je Čepmen-Kolmogorova jednačina za $p(k)$:

$$p(k) = p(0) * P^k.$$

Definicija 11. Ako $p(k)$ ne zavisi od k, kažemo da je lanac *stacionaran.*

Definicija 12. Za Markovljev lanac kažemo da je *ergodičan* ako postoji $n \in \mathbb{N}$ tako da matrica $P_n = P^n$ ima sve pozitivne elemente.

Definicija 13. Za slučajni proces Markova kažemo da je *ergodičan*, ako je definisan na diskretnom skupu i ako po isteku dovoljno velikog intervala vremena, verovatnoće stanja sistema ne zavise od početnih uslova, početnog trenutka, ni vremena koje je prošlo.

Definicija 14. Za proces Markova kažemo da je *nesvodljiv* ukoliko se u svako stanje procesa može doći iz drugog stanja procesa.

Definicija 15. Ako je lanac *nesvodljiv* tada se sva stanja ponavljaju ili se iz svakog stanja može preći u drugo stanje.

Definicija 16. Za svaki *ergodičan* lanac i za svako i postoje verovatnoće $p_j^* = \log_{n\to\infty} p_{i,j}(n)$ koje se nazivaju granične verovatnoće. Drugim rečima, posle dovoljno dugo vremena iz bilo kog stanja sistem završava u stanju j sa verovatnoćom p_j^*.

Definicija 17. Stanje x_j je *povratno* ako postoji $n \in \mathbb{N}$ tako da $p_{jj}(n) > 0$.

Definicija 18. *Finalne (granične) verovatnoće su definisane sa*

$$p_j^* = \lim_{n\to\infty} \mathrm{p_{ij}(n)}\,, \forall i \in N, j = 1, \dots, \infty.$$

Predstavljaju udeo vremenskog perioda koji sistem provede u stanju j.
Za ergodične lance, finalne verovatnoće računamo rešavanjem sistema:

$$p^* = p^* P;\ \textstyle\sum_{j=1}^{\infty} p_j^* = 1.$$

Definicija 19. *Neka je* $\{X(n), n \in N_0\}$ *Markovljev lanac. Stanje* $\mathrm{x_i}$ *je povratno ako P{* X_n *=* $\mathrm{x_i}$*, za neko n ≥ 1 |* $X_0 = \mathrm{x_i}$ *} = 1 tj. ako je verovatnoća da se sistem vrati u stanje* $\mathrm{x_i}$ *pri uslovu da je u početnom trenutku u tom stanju, jednaka 1. U suprotnom, ako je P{* X_n *=* $\mathrm{x_i}$*, za neko n ≥ 1 |* $X_0 = \mathrm{x_i}$ *} < 1, stanje* $\mathrm{x_i}$ *je prolazno.*

Definicija 20. *Unutar lanca Markova, kažemo da je stanje* $\mathrm{x_j}$ *moguće dostići iz stanja* $\mathrm{x_i}$ *ako* $\exists n \in N$ *takvo da je* $\mathrm{p_{jj}(n)} > 0$.

Definicija 21. *Stanje* $\mathrm{x_j}$ *je apsorbujuće ako je* $\mathrm{p_{jj}} = 1$. *Jednom kad proces uđe u to stanje, ostaje u njemu.*

Primer 1. Lanac Markova

Pretpostavimo da to da li će sutra padati kiša ili ne, zavisi od prethodnih uslova tako što zavisi samo od toga da li danas pada kiša ili ne, a ne od prethodnih dana. Takođe, pretpostavimo da ako danas padne kiša, padaće i sutra sa verovatnoćom α, a ako danas ne padne onda će sutra padati sa verovatnoćom β.

Imamo Markovski lanac sa dva stanja: 0-pada kiša i 1-ne pada kiša.

Matrica prelaza izgleda ovako: $P = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix} = \begin{bmatrix} \alpha & 1-\alpha \\ \beta & 1-\beta \end{bmatrix}$

Sada uzmimo npr. $\alpha = 0.7$ i $\beta = 0.4$ .Verovatnoća da npr. padne kiša 4.dana je $p_{00}(4)$ i da bismo je dobili izračunaćemo matricu P_4 u čijem će se gornjem levom uglu nalaziti upravo tražena verovatnoća.

$$P^2 = P \cdot P = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} = \begin{bmatrix} 0.61 & 0.39 \\ 0.52 & 0.48 \end{bmatrix}$$

$$P_4 = P^4 = P^2 \cdot P^2 = \begin{bmatrix} 0.61 & 0.39 \\ 0.52 & 0.48 \end{bmatrix} \begin{bmatrix} 0.61 & 0.39 \\ 0.52 & 0.48 \end{bmatrix} = \begin{bmatrix} \mathbf{0.5749} & 0.4254 \\ 0.5668 & 0.4332 \end{bmatrix}$$

$p_{00}(4) = 0.5749$ je verovatnoća da padne kiša 4.dana.

2.3. Teorija grafova

Grafovi su matematički objekti koje često srećemo u svakodnevnom životu : geografska mapa sa gradovima povezanim putevima, skup ljudi sa relacijom poznanstva, strukturna formula nekog molekula ili jedinjenja, šema električnog kola...

Zbog osobine grafova da prikazuju entitete u paru i veze između njih, teorija grafova je vrlo zastupljena za opis modela i strukture podataka.

Zbog velikog spektra primena, kao i izuzetno jednostavne veze definicije i osnovnih svojstava, grafovi su našli veliku primenu ne samo u drugim matematičkim oblastima poput kombinatorike, kombinatorne optimizacije, operacionih istraživanja, linearne algebre, kompleksne analize, nego i u drugim (nematematičkim) naukama kao što su elektrotehnika, računarstvo, hemija, fizika, biologija, sociologija, vojne nauke...

Jedna od značajnih primena grafova u oblasti informatike (ali i nekim drugim oblastima) je upravo ona kojom ćemo se služiti u ovom radu – analiza mreže.

Definicija 22. *Graf G je uređen par $(V(G), E(G))$, gde je $V(G)$ konačan neprazan skup elemenata koji se zovu čvorovi, a $E(G)$ je konačan skup različitih neuređenih parova različitih elemenata skupa $V(G)$ koji se zovu grane.*

Graf G se može geometrijski predstaviti crtežom u ravni. Čvorovi grafa se predstavljaju tačkama ravni, a grane grafa linijama koje povezuju odgovarajuće čvorove.

Definicija 23. *Graf G je povezan ako se svaka dva njegova čvora mogu povezati putem. Ako postoje čvorovi koji se ne mogu povezati putem, graf je nepovezan.*

Definicija 24. *Bipartitan graf je graf čiji se skup čvorova moze razbiti na dva disjunktna skupa (partitivni skupovi) na takav način da svaka grana spaja čvor prvog skupa sa čvorom drugog skupa.*

Definicija 25. *Za granu e = (u, v) orijentisanog grafa (V, ρ) kažemo da vodi iz čvora u u čvor v (e izlazi iz čvora u, a ulazi u čvor v).*

Ulazni stepen indeg(v) čvora v je broj grana koje ulaze u v.

Izlazni stepen outdeg(v) čvora v je broj grana koje izlaze iz v.

Ulazni skup I(v) čvora v je skup čvorova iz kojih vodi grana u v, I(v) = {x | (x, v) ∈ ρ }. Izlazni skup O(v) čvora v je skup čvorova u koje vodi grana iz v, O(v) = {x | (v, x) ∈ ρ}. Petlja je grana koja i ulazi i izlazi iz čvora.

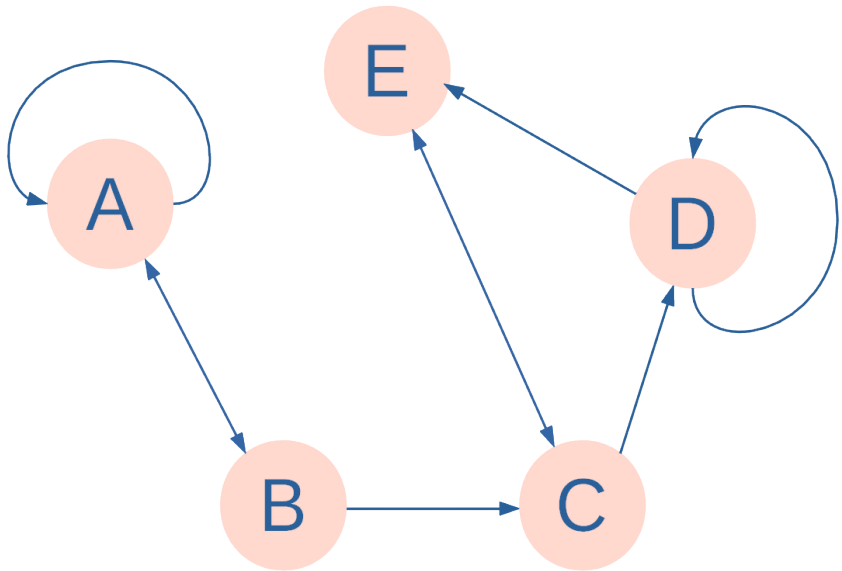
Definicija 26. *Broj grana koje se stiču u čvoru v zove se stepen čvora v (eng. degree) i označava se sa d(v).*

Definicija 27. *Čvor v koji nema susednih čvorova, tj. za koji je d(v) = 0, nazivamo izolovan (viseći) čvor.*

Definicija 28. *Graf G je regularan ako su stepeni svih njegovih čvorova jednaki.*

Primer 2. Graf i njegove osnovne komponente

Za dati graf ispisati skup čvorova, skup grana, a zatim za svaki čvor izlazni i ulazni skup i izlazni i ulazni stepen.



Slika1 - primer grafa

$V = \{A, B, C, D, E\}$

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

$I(A) = \{A, B\} \Rightarrow indeg(A) = 2$

$O(A) = \{A, B\} \Rightarrow outdeg(A) = 2$

$I(B) = \{B\} \quad \Rightarrow indeg(B) = 1$

$O(B) = \{A, C\} \Rightarrow outdeg(B) = 2$

$I(C) = \{B, E\} \Rightarrow indeg(C) = 2$

$O(C) = \{D, E\} \Rightarrow outdeg(C) = 2$

$I(D) = \{C, D\} \Rightarrow indeg(D) = 2$

$O(D) = \{D, E\} \Rightarrow outdeg(D) = 2$

$I(E) = \{C, D\} \Rightarrow indeg(E) = 2$

$O(E) = \{C\} \quad \Rightarrow outdeg(E) = 1$

3. Pretraživanje Veba

Veb ili ***svetska mreža*** *(engl.* ***World Wide Web, WWW****)* je sistem međusobno povezanih, hipertekstualnih dokumenata koji se nalaze na internetu, a koje nazivamo web stranicama. Stvorili su ga Englez Tim Berners-Li i Belgijanac Robert Kajo, 1990. godine, radeći u CERN-u u Ženevi. Ovaj pojam se često pogrešno koristi kao sinonim za internet, a u stvari označava samo jednu od usluga koje omogućava internet. Uz pomoć internet pregledača, korisnici mogu da gledaju veb stranice koje obično sadrže tekst, slike, zvučni i video-zapis. Primarni je alat za interakciju na internetu.

Veb stranica (eng. web page) se sastoji od niza tih dokumenata, a ***veb sajt (web site)*** je skup povezanih web-stranica (uglavnom sa istog web servera). Veze između dokumenata se nazivaju ***hipervezama (eng. hiperlink)***, a odomaćena je i reč link.

Web pretraživač je internet servis čija je svrha pretraživanje weba zadavanjem kljucnih reči ili ređe biranjem između ponuđenih izbora.

Biranje putem ponuđenih izbora je starija metoda i danas se retko koristi. Odnosi se na manje nepovezane kolekcije dokumenata kakve su postojale i pre weba. Te kolekcije su organovane i kategorizovane od strane stručnjaka i predstavljaju kataloge web stranica, klasifikovane po tematici . Ovakav pristup se brzo pokazao kao neefikasan, prvenstveno usled ogromne brzine ekspanzije Web-a, koja uzrokuje novim sadržajima, čije je pronalaženje i klasifikacija izuzetno zahtevan posao. Dodatno, obzirom na ekspanzivan rast količine informacija, klasifikacija se pokazuje kao nedovoljno efikasan pristup za rešavanje problema preopterećenosti informacijama (information overload). Modeli za ovako pretraživanje su se razvili ´60ih i predstaljaju preteču današnjih pretraživača,ca najpoznatiji su:

- **Booleov model** koji je dobio ime po Bulovoj algebri jer koristi logičke veznike I, ILI i NE koji logički spajaju riječi i oblikuju upit (eng. Query). Ako tekst sadrži reč ili više reči koje odgovaraju logičkom izrazu iz upita, onda je relevantan,a u suprotnom se smatra nerelevantnim. Pored toga što je potrebno poznavanje Booleovih operatora da bi je upit dobro uneo, dve najvece mane ovog modela su ujedno dva standardna problema pretraživanja:
  - *Polisemija* – dovodi do toga da rezultati upita budu nerelevantni dokumenti koji sadrže neko drugo značenje reči iz zadatog upita
  - *Sinonimija* - ovakav pretraživač ne može semantički povezati reč iz upita s dokumentima koji sadrže njen sinonim

  Koristi se kao baza za današnje pretraživače u vidu polja u naprednim opcijama (I - sve reči, ILI – bar jedna od ovih reči, NE – nijedna od ovih reči).
- **Model vektorskog prostora** je matematički model za prikaz tekstualnih dokumenata kao vektora njihovih identifikatora
- **Probabilisticki model** procenjuje da li će dokument biti značajan za korisnika i pri tome koristi rekurziju – postavi početne verovatnoće da je dokument

relevantan,a zatim ih pokušava poboljšati kroz iteracije. Za početne vrednosti koristi podakte iz prethodnih upita tog korisnika. Mana je što su izuzetno kompleksni i teški za implementaciju.

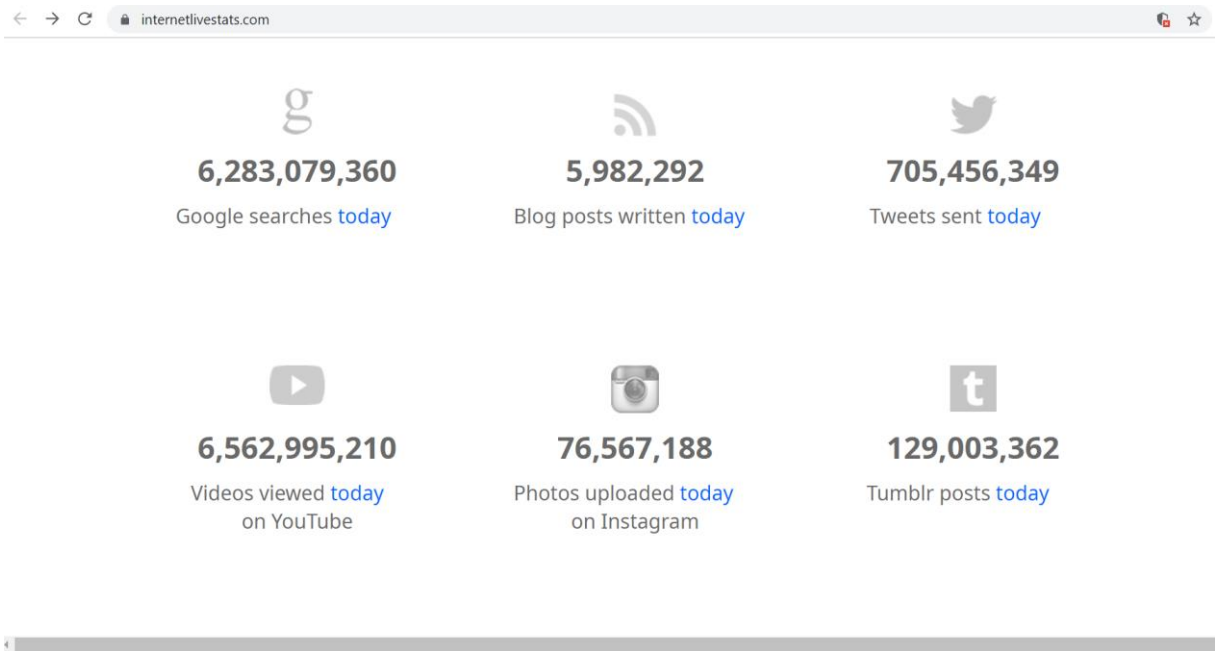
Loša strana ovog pristupa je što rangiranje stranica zavisi od sadržaja koji se nalazi na njoj. Možemo npr. dodati na stranicu neku reč (koja ni ne mora imati veze sa sadržajem) u istoj boji pozadine stranice (tako da nije uočljiva) i ponoviti je veliki broj puta. Rang bi joj se povećao i korisnici koji su pretraživali tu reč bi došli na stranicu koja im nije relevantna.

Najznačajnije mere uspešnosti nekog pretraživača su:

- Preciznost - broj relevantnih dokumenata/ broj dokumenata
- Odziv - broj pronađenih relevantnih dokumenata/broj relevantnih dokumenata

Mere koje su takođe značajne su optimizacija prostora i vremena - brži i memorijski optimizovani pretraživači imaju prednost.

Na sajtu https://www.internetlivestats.com/ može se pratiti statistika na internetu u svakom trenutnku. Da bismo stvorili ideju o količini podataka koji svakosnevno pristignu pogledajmo podatke sa ove stranice u nekom trenutunku početkom septembra :



Slika2 – snimak ekrana, stranica internetlivestats.com

Ekspanzija sadržaja na webu je vrlo očigledna i predstavlja težak izazov za njegovo pretraživanje. Jasno je da tradicionalne tehnike više nisu dovoljne. Pri svakom pretraživanju kao rezultat ćemo dobiti veliki broj strana. Ono što nas zanima je koji su najrelevantnije među njima.

4. PageRank

1998. Studenti postdiplomskih studija na Univerzitetu Stenford Sergej Brin i Lari Pejdž uvideli su potencijal grafova kao prezentaciju najveće svetske mreže te su kao projekat na fakultetu razvili algoritam PageRank. Napustili su fakultet i projekat su integrisali u Gugl (eng. Google).

Ideju za algoritam su predstavili kao vrstu preporučivanja među sajtovima. Među najpoznatijim primerima je „Preporuka od Donalda Trampa“ – ako vas preporuči neko kao što je Donald Tramp, ta preporuka sigurno ima veću vrednost nego preporuka od strane neke nepoznate osobe. Sa druge strane, Donald Tramp je napisao veliki broj preporuka u svom životu pa su one gubile na značaju.

Ovaj algoritam se ne računa prilikom upita pa nije zavisan od njih, već se računa u odrađenim vremenskim periodima. Dužina tih perioda često zavisi od važnosti sajta, a postoji mogućnost i da vlasnik sajta zahteva da se ponovo pregleda sajt.

Primena ovog algoritama je vrlo raznovrsna. Može biti primenjen na svim mrežama u kojim je potrebno rangiranje čvorova (mreža rutera, mreža email-ova, mreža prevoza, društvene mreže, baza podataka, ekonomske mreže, mreža modela širenja zarazne bolesti, mreža napajanja...).

4.1. Računanje PageRank-a

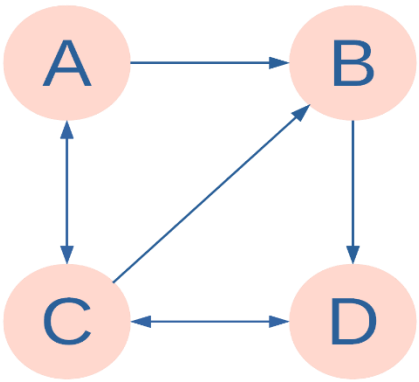
Zamislimo povezan usmeren graf u kom su čvorovi web stranice, a grane su linkovi među njima. Pagerank $\pi(x)$ je funkcija koja svakom čvoru, tj. stranici dodeljuje neki pozitivan broj ($\pi(x) \geq 0, \forall x$). Što čvor ima veći broj to je stranica važnija.

4.1.1 Razvoj

Ideja1: Stranica je visoko rangirana ako visoko rangirane stranice imaju link ka njoj.

Problem: Ova definicija je ciklicna.

Primer 3. Ako je PageRank neke stranice jednak zbiru PageRank-ova svih stranica koje pokazuju na nju to možemo napisati na sledeći način:



Slika3 - Jednostavan web graf

$$\pi(D) = \pi(C) + \pi(B)$$

$$\pi(B) = \pi(A) + \pi(C)$$

$$\pi(C) = \pi(A) + \pi(D)$$

$$\pi(A) = \pi(C)$$

Kada sumiramo prethodne jednačine dobijamo:

$$2\pi(A) + \pi(C) = 0,$$

iz čega sledi :

$$\pi(A) = \pi(B) = \pi(C) = \pi(D) = 0.$$

Sta je pogrešno? Ne možemo tretirati sve stranice isto - npr B i A imaju linkove ka D, ali za B je to jedini link koji ima, dok A ima link ka jos dve stranice.

Ideja2: Podeliti rang stranice sa njenim brojem linkova ka drugim stranicama

Ako se vratimo na graf sa Slike1 i primenimo ovu ideju dobijamo :

$$\pi(D) = \frac{1}{3}\pi(C) + \pi(B)$$

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

$$\pi(C) = \frac{1}{2}\pi(A) + \pi(D)$$

$$\pi(A) = \frac{1}{3}\pi(C).$$

Ako sada sumiramo prethodne jednačine dobijamo:

$$\pi(A) + \pi(B) + \pi(C) + \pi(D) = \pi(A) + \pi(B) + \pi(C) + \pi(D).$$

Dakle, u pitanju je neodređen sistem (ima beskonačno mnogo rešenja).

Ovo mozemo popraviti tako sto ćemo „normalizovati“ rangove:

$$\pi(A) + \pi(B) + \pi(C) + \pi(D) = 1.$$

Primetimo: $\pi(x)$ sada mozemo posmatrati kao verovatnoće.

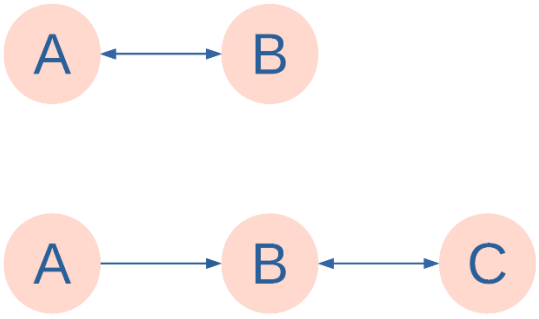
PageRank izračunat na ovaj način se naziva „naivni“ PageRank. *Sta je ovde naivno?*

1) Lako se moze manipulisati dodavanjem dodatnih stranica

Primer4: Manipulacija dodavanjem stranica

Imamo dva čvora A i B koji imaju linkove jedan ka drugom:

$$\pi(A) = \pi(B) = \frac{1}{2}$$

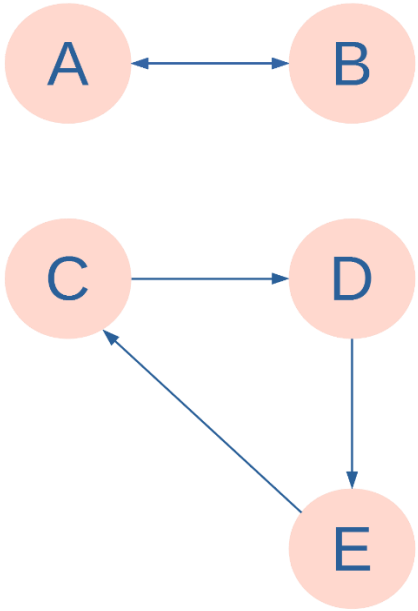


Slika4 - Manipulacija dodavanjem stanica

Ukoliko ubacimo dodatnu stranicu C koja ima link ka čvoru B i B ima link ka njoj, a više nema ka A rang stranica se menja na sledeći način: $\pi(A) = 0$, $\pi(B) = \pi(C) = \frac{1}{2}$.

2) Ukoliko imamo dva disjunktna grafa ponovo imamo beskonačno mnogo rešenja

Primer 5: Disjunktni grafovi



Slika5 – disjunktni grafovi

$$\pi(A) = \pi(B), \;\; \pi(C) = \pi(D) = \pi(E)$$

Ideja3: rang meren vremenom provedenim na sajtu – svodi se na „naivan“PageRank

Zamislimo sada nasumičnog surfera (eng.random surfer) koji nasumično bira linkove i „skače“ sa stranice na stranicu ka kojoj nema link.

Ukoliko se nalazi na nekoj strainici X, neka je ε verovatnoća da klikne na link na stranici na kojoj se nalazi i time se prebaci na sledeću, a $1 - \varepsilon$ verovatnoća da pređe na neku drugu nasumičnu stranicu.

Neka je $\pi(i)$ rang i-te stranice. Prema naivnom PageRank-u tada je:

$$\pi(i) = \sum_{j:\, j\ ima\ link\ ka\ i} \frac{\pi(j)}{|outdeg(j)|}.$$

gde je outdeg(j) broj linkova koji idu od stranice j. Ako uključimo nasumičnu šetnju u ovu priču dobijamo :

$$\pi(i) = \frac{\varepsilon}{N} + (1 - \varepsilon) \sum_{j:\, j\ ima\ link\ ka\ i} \frac{\pi(j)}{|indeg(j)|}.$$

gde je N broj stranica u mreži (čvorova u grafu).

Ideja4: Kako cemo izračunati $\pi(i)$?

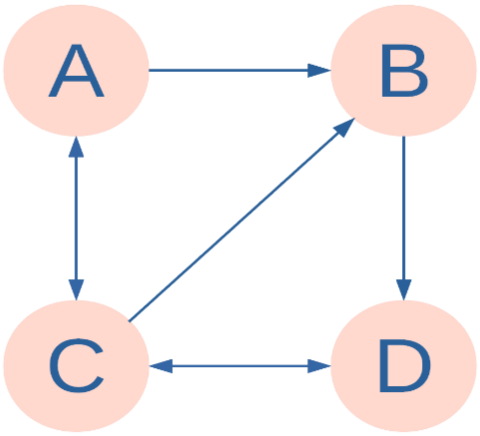
Koristićemo stepeni iterativni metod (eng.power iteration method) i pokazaćemo da za dovoljno velik broj iteracija $\pi(i)$ postaje verovatnoća poseta stranice i.

4.1.2. Stepeni metod i matrica prelaza

Stepeni metod (power metod) predstavlja funkcionalnu vezu izmedju dve veličine gde promena jedne veličine izaziva proporcionalnu promenu druge veličine, nezavisno od početne vrednosti.

Kao i kod svakog iterativnog metoda, cilj nam je da poboljšavamo procenu svakom narednom iteracijom.

Da bismo približili kako to izgleda vratimo se na primer iz **Ideje2**:



Slika3 - Jednostavan web graf

	Iteracija 0	Iteracija 1	Iteracija 2	...	PageRank
$\pi(A)$	1/4	2/24	3/24	...	1
$\pi(B)$	1/4	5/24	4/24	...	2
$\pi(C)$	1/4	9/24	9/24	...	4
$\pi(D)$	1/4	8/24	4/24	...	3

Prepostavimo da je $\varepsilon = 0$.

-Iteracija 0: Postavimo inicijalne vrednosti na $\frac{1}{N}$, $N = 4 \Rightarrow \frac{1}{4}$.

-Iteracija 1:

$$\pi(A) = \frac{1}{3}\pi(C) = \frac{1}{3}\cdot\frac{1}{4} = \frac{2}{24}$$

$$\pi(B) = \frac{1}{3}\pi(C) + \frac{1}{2}\pi(A) = \frac{1}{3}\cdot\frac{1}{4} + \frac{1}{2}\cdot\frac{1}{4} = \frac{5}{24}$$

$$\pi(C) = \frac{1}{2}\pi(A) + \pi(D) = \frac{1}{2}\cdot\frac{1}{4} + \frac{1}{4} = \frac{9}{24}$$

$$\pi(D) = \frac{1}{3}\pi(C) + \pi(B) = \frac{1}{3}\cdot\frac{1}{4} + \frac{1}{4} = \frac{8}{24}$$

-Iteracija 2:

$$\pi(A) = \frac{1}{3}\pi(C) = \frac{1}{3}\cdot\frac{9}{24} = \frac{3}{24}$$

$$\pi(B) = \frac{1}{3}\pi(C) + \frac{1}{2}\pi(A) = \frac{1}{3}\cdot\frac{9}{24} + \frac{1}{2}\cdot\frac{2}{24} = \frac{4}{24}$$

$$\pi(C) = \frac{1}{2}\pi(A) + \pi(D) = \frac{1}{2}\cdot\frac{2}{24} + \frac{8}{24} = \frac{9}{24}$$

$$\pi(D) = \frac{1}{3}\pi(C) + \pi(B) = \frac{1}{3}\cdot\frac{9}{24} + \frac{5}{24} = \frac{8}{24}$$

... do koje iteracije idemo? Najčešće se unapred odredi prag tolerancije kao razlika između dve iteracije koju treba zadovoljiti. Tačne vrednosti PageRanka nas ionako ne zanimaju, ono što je bitno je poređenje njihovih vrednosti. Time dolazimo do zaključka da je dovoljno iterirati do konvergencije.

Formirajmo sada matricu dimenzija $n \times n$, gde je n broj stranica sa elementima M_{ij} na sledeći način :

$$Mij = \begin{cases} 0, & ako\ ne\ postoji\ nijedan\ link\ na\ toj\ stranici \\ \dfrac{1}{outdeg(i)}, & inače \end{cases}$$

Matricu M određuje usmeren graf G i može se pretpostaviti da je graf G jako povezan. Možemo zaključiti da to znači da ne postoji nijedan viseći čvor, tj. ne postoji stranica koja nema link ka nekoj drugoj stranici. U narednom odeljku će ovaj problem biti detaljnije objašnjen.

Ukoliko pogledamo osobine matrice prelaza za lanac Markova (2.2.) možemo zaključiti da ih navedena matrica ima, dakle ona je matrica prelaza. Takođe je i stohastička matrica jer joj je zbir po kolonama jednak 1.

Pokažimo sada kako ova matrica izgleda na konkretnom primeru grafa sa Slike1:

$$P = \begin{bmatrix} 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & 0 & 1 \\ 0 & 1 & \frac{1}{3} & 0 \end{bmatrix}$$

Kako ćemo sad iskoristiti ovu maticu da dobijemo PageRank?

Koristićemo jednačinu koja opisuje iterativni stepeni metod: $\pi_{i+1} = P\pi_i$.

Početni vektor nam je $\pi_0 = \begin{bmatrix} 1/4 \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}$ pa ćemo π_1 dobiti na sledeći način :

$$\pi_1 = \begin{bmatrix} 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & 0 & 1 \\ 0 & 1 & \frac{1}{3} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix} = \begin{bmatrix} \frac{2}{24} \\ \frac{5}{24} \\ \frac{9}{24} \\ \frac{8}{24} \end{bmatrix}$$

Dalje možemo nastaviti :

$$\pi_2 = P\pi_1 = P^2\pi_0$$

$$\pi_3 = P\pi_2 = P^3\pi_0$$

...

$$\pi_{r+1} = P\pi_r = P^r\pi_0$$

Ako u ovu jednačinu dodamo i nasumičnog surfera (**Ideja3**), dolazimo do formule:

$$\pi_{r+1}(i) = \frac{\varepsilon}{N} + (1-\varepsilon)\sum_{j:\, j\ ima\ link\ ka\ i} \frac{\pi_r(j)}{|van(j)|}.$$

Pitanje: Da li $\pi_r(i)$ konvergira ka $\pi(i)$ kada $r \to \infty$?

Definišimo rastojanje između iteracije i stvarne vrednosti: $\Delta_r = \sum_i |\pi_r(i) - \pi(i)|$.

Cilj nam je da pokažemo da je $\lim_{r\to\infty} \Delta_r = 0$.

Za sada znamo:

$$\pi_{r+1}(i) = \frac{\varepsilon}{N} + (1-\varepsilon)\sum_{j:\, j\ ima\ link\ ka\ i} \frac{\pi_r(j)}{|van(j)|},$$

$$\pi(i) = \frac{\varepsilon}{N} + (1-\varepsilon)\sum_{j:\, j\ ima\ link\ ka\ i} \frac{\pi(j)}{|van(j)|}.$$

Ako oduzmemo donju jednačinu od gornje dobijamo:

$$\pi_{r+1}(i) - \pi(i) = (1-\varepsilon)\sum_{j:\, j\ ima\ link\ ka\ i} \frac{\pi_r(j)-\pi(j)}{|van(j)|},$$

a zatim na dobijenu jednačinu primenimo nejednakost trougla:

$$|\pi_{r+1}(i) - \pi(i)| \leq (1-\varepsilon)\sum_{j:\, j\ ima\ link\ ka\ i} \frac{|\pi_r(j)-\pi(j)|}{|van(j)|}\ .$$

I na kraju, sumiranjem po i dobijamo:

$$\begin{aligned}\Delta_{r+1} = \sum_i |\pi_{r+1}(i) - \pi(i)| &\leq (1-\varepsilon)\sum_i \sum_{j:\, j\ ima\ link\ ka\ i} \frac{|\pi_r(j)-\pi(j)|}{|van(j)|} \\ &= (1-\varepsilon)\sum_j \sum_{i:\, i\ ima\ link\ ka\ j} \frac{|\pi_r(j)-\pi(j)|}{|van(j)|} \\ &= (1-\varepsilon)\sum_j |\pi_r(j)-\pi(j)| = (1-\varepsilon)\Delta_r.\end{aligned}$$

Dakle,

$\Delta_{r+1} \leq (1-\varepsilon)\Delta_r$ iz čega dobijamo:

$\Delta_r \leq (1-\varepsilon)^2\Delta_{r-1} \leq (1-\varepsilon)^3\Delta_{r-2} \ldots \leq (1-\varepsilon)^r\Delta_0.$

Kako je $\Delta_r \leq (1-\varepsilon)^r\Delta_0$, sledi da $\Delta_r \to 0$ kada $r \to \infty$.

Zaključak: $\forall i,\ \boldsymbol{\pi_r}(i) \underset{r\to\infty}{\longrightarrow} \boldsymbol{\pi}(i)$ **.**

4.1.3. Problemi

Problem 1: viseći čvorovi

Kao što je navedeno ranije, viseći čvor je svaka stranica koja ne poseduje link ka nekoj drugoj stranici.

Navedimo jednostavan primer od tri čvora u kom je C viseći:

Primer 6: Viseći čvor

Slika6 – Viseći čvor

Kako je N jednako 3, uzmimo za početni vektor: $v_0 = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$. Tada matrica prelaza izgleda ovako:

$$P = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}.$$

U prvoj iteraciji imamo :

$$v_1 = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix}$$, što već deluje zabrinjavajuće.

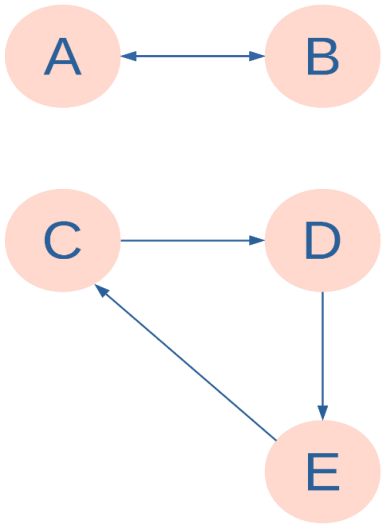
A u drugoj nam je već sasvim jasno da nešto nije u redu jer dobijemo:

$$v_2 = \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$ Dakle – *PageRank svih stranica je 0.*

Jedno od mogućih rešenja ovog problema je da ih potpuno izbacimo. Međutim, među takvim stranicama se nalaze npr. značajni naučni radovi u pdf formatu ka kojima vode neki od značajnih izvora zbog čega zaslužuju zaslužuju velik PageRank, te ovo rešenje nije baš najbolje.

Problem 2: disjunktni grafovi

Vratimo se na:



Slika5 – primer disjunktnog grafa

Problem je to što nema načina da se pređe iz jednog grafa u drugi ako prelazimo sa jedne stranice na drugu samo putem linkova. Zato je nasumični surfer dobro rešenje za ovaj problem (**Ideja3**).

Rešenje za ova dva problema je uvođenje ***faktora prigušenja*** d (**eng. dampling factor**). Jednačina koja opisuje PageRank sa njim izgleda ovako:

$$\pi(i) = (1-d) + d\sum_{j:\, j\ ima\ link\ ka\ i} \frac{\pi(j)}{|indeg(j)|}.$$

Odabir ovog faktora se zasniva na balansiranju. Što je manji brža je konvergencija, ali je i nerealnije opisana struktura weba. Kada faktor teži ka 1 konvergencija je brža, ali algoritam postaje osetljiviji. Male promene faktora mogu da uzrokuju velike promene u PageRanku.

Primer7: Kod za izračunavanje PageRanka

Dat je kod u Pythonu[1] pomoću kog možemo računati PageRank. Imamo dva slučaja – kada unosimo matricu prelaza i kada je za dati broj čvorova sami generisemo poznavajući osobine matrice prelaza. U oba slučaja ćemo koristiti početni vektor čiji

[1]Kod na stranici https://www.tutorialspoint.com/page-rank-algorithm-and-implementation-using-python je korišćen kao osnova.

elementi imaju uniformnu raspodelu. Osnovu čini formula sa faktorom d. Zbog jednostavnosti i preglednosti koristiću mali broj čvorova.

Primer 7.1: Računanje PageRanka za nasumično generisanu matricu prelaza i uneti broj čvorova

```
import numpy as np

#ovde definisemo matricu prelaza za uneti broj cvorova
def randomInputMatrix(N):
  #treba da bude kvadratna i da su joj dimenzije broj cvorova
  mat = np.random.rand(N, N)
  #zbir po kolonama treba da bude jedan
  for i in range (N):
    #normiramo vrste
    mat[i] = mat[i] / np.linalg.norm(mat[i], 1)
    #a zatim transponujemo matricu da postanu kolone
  mat = np.transpose(mat)
  print("Generisana matrica prelaza:\r\n",mat)
  return mat
#Za uneti broj cvorova N racunamo pagerank
#d je damping faktor
#eps je epsilon vrednost za koju se zaustavlja iterativni postupak
def pagerank_N(N, eps=1.0e-8, d=0.85):
  #nasumicno generisemo matricu prelaza za N cvorova
    M = randomInputMatrix(N)
    #pocetni vektor ima uniformnu raspodelu-elementi su mu 1/N
    v = np.ones((N, 1), dtype=np.float32) / N
    print("Početni vektor:\r\n", v)
    #deklarisemo ovu promenljivu tako da upadnemo u while petlju
    last_v = np.ones((N, 1), dtype=np.float32) * 100
    #k je broj iteracije
    k = 0
    #kriterijum za zastavljenje iterativnog postupka
    #stajemo kada je razlika izmedju dve iteracije manja ili jednaka
epsilon
while np.linalg.norm(v - last_v, 2) > eps:
      #svaki put kada udjemo u while petlju broj iteracija treba povecati
      k = k + 1
      #u last_v smestamo poslednju iteraciju
      last_v = v
      #zatim racunamo novu po formuli za pagerank
      v = d * np.matmul(M, v) + (1 - d) / N
      print("U iteraciji ",k," PageRank vektor izgleda
ovako:\r\n",np.transpose(v))
    print("Broj iteracija: ",k)
    print("PageRank vektor:\r\n",np.transpose(v))
    return v
```

Pokrenimo program sa sledećom linijom:

```
pagerank_N(5, 1.0e-3, 0.85)
```

Kao rezultat dobijamo sledeće:

```
Generisana matrica prelaza:
 [[0.27151177 0.26007772 0.03581892 0.06019932 0.10870657]
 [0.18712956 0.21005313 0.36112603 0.10946335 0.20484732]
 [0.17277292 0.06126742 0.13132594 0.27258397 0.24703583]
 [0.03534824 0.1867748  0.15907066 0.44099039 0.23405308]
 [0.33323751 0.28182693 0.31265845 0.11676298 0.2053572 ]]
Početni vektor:
 [[0.2 0.2 0.2 0.2 0.2]]
U iteraciji  1  PageRank vektor izgleda ovako:
 [[0.15517343 0.2123453  0.18044764 0.20956032 0.24247332]]
U iteraciji  2  PageRank vektor izgleda ovako:
 [[0.15137572 0.20970262 0.18345854 0.21956318 0.23589995]]
U iteraciji  3  PageRank vektor izgleda ovako:
 [[0.14991119 0.20933709 0.18403664 0.22187838 0.23483672]]
U iteraciji  4  PageRank vektor izgleda ovako:
 [[0.14953021 0.20924661 0.18418022 0.22251081 0.23453215]]
Broj iteracija:  4
PageRank vektor:
 [[0.14953021 0.20924661 0.18418022 0.22251081 0.23453215]]
```

Za početak, primetimo da poredak nije isti u svim iteracijama.

U prvoj iteraciji poredak je sledeći :

PR(čvor5) > PR(čvor2) > PR(čvor4) > PR(čvor3) > PR(čvor1),

dok je na kraju:

PR(čvor5) > PR(čvor4) > PR(čvor2) > PR(čvor3) > PR(čvor1).

Primećujemo da peti čvor ima najveći rank, a čvor 1 najmanji. Pogledajmo sada matricu. U njenom prvom redu koji odgovara prvom čvoru primećujemo manje vrednosti prelaza (čak dve vrednosti 0.0), dok u petom redu koji odgovara petom čvoru imamo tri verovatnoće ≈ 0.3.

Napomena: Broj iteracija je mali jer je odabran prilično jednostavan graf i realtivno mala preciznost. U literaturi [4] osnivači algoritma spominju da im je bilo potrebno 50-100 iteracija stepenog metoda.

Primer 7.2. Izračunavanje PageRanka za unetu matricu

```
#Za unetu matricu prelaza M racunamo pagerank
#d je damping faktor
#eps je epsilon vrednost za koju se zaustavlja iterativni postupak
def pagerank_M(M, eps=1.0e-8, d=0.85):
    # broj čvorova dobijamo kao dimenziju matrice M
    N = M.shape[1]
    #pocetni vektor ima uniformnu raspodelu - elementi su mu 1/N
    v = np.ones((N, 1), dtype=np.float32) / N
    print("Početni vektor:\r\n", np.transpose(v))
    #deklarisemo last_v tako da udjemo u while petlju
    last_v = np.ones((N, 1), dtype=np.float32) * 100
    #k je broj iteracije
    k = 0
    #kriterijum za zastavljenje iterativnog postupka
    #stajemo kada je razlika izmedju dve iteracije manja ili jednaka
epsilon
    while np.linalg.norm(v - last_v, 2) > eps:
     #u last_v smestamo poslednju iteraciju
        last_v = v
        #zatim racunamo novu po formuli za pagerank
        v = d * np.matmul(M, v) + (1 - d) / N
        #svaki put kada udjemo u while petlju broj iteracija treba povecati
        k = k + 1
    print("Broj iteracija: ",k)
    print("PageRank vektor:\r\n", np.transpose(v))
    return v
```

Ukoliko se ponovo vratimo na graf sa Slike1 i pokrenemo navedeni program sa istom matricom prelaza i istim početnim vektorom sa kojima smo računali u 4.1.2:

```
M = np.array([[0, 0, 1/3, 0],
              [1/2, 0, 1/3, 0],
              [1/2, 0, 0, 1],
              [0, 1, 1/3, 0]])

pagerank_M(M, 1.0e-8, 0.85)
```

dobićemo sledeće :

```
Početni vektor:
 [[0.25 0.25 0.25 0.25]]
Broj iteracija:  1
PageRank vektor:
 [[0.10833333 0.21458333 0.35625    0.32083333]]
```

Razlika u izračunatim vrednostima je nastala kao posledica dodavanja faktora d. Ono što možemo primetiti je da je rangiranje isto - najveći rang ima čvor C, posle njega D, zatim B i na kraju A.

4.2. Personalizovani PageRank

Kako možemo personalizovati PageRank? Uticaćemo na ono što možemo menjati – poziciju sa koje (ponovo) krećemo, tj. biranjem čvora na koje će se "transportovati" naš nasumični surfer (**Ideja3**).

Ako se vratimo na odeljak 4.1.3. na "iteraciju 0" primetimo da je za početni vektor uzeta uniformna raspodela verovatnoća ($\pi_0 = \left(\frac{1}{N}, \dots, \frac{1}{N}\right)$, gde je N broj čvorova). Mi možemo prema ličnim preferencijama dodeliti neke druge vrednosti i krenuti od njih.

Faktor personalizacije je takođe značajan u borbi protiv ""farmi linkova" - dodavanja stranica koji linkuju stranicu kojoj neko želi da poveća rang.

Personalizovani PageRank ima široku primenu i smatra se da se budućnost pretraživanja zasniva upravo na personalizaciji, iako je računanje personalizovanog PageRanka daleko komplikovanije .

4.3. Aproksimacija i ažuriranje

Da bismo održavali PageRank ažuriran pri svakoj promeni čvora, najjednostavnije bi bilo da za svaku promenu računamo vrednosti ispočetka. Naravno, ovo bi koštalo dosta i zato su potraženi načini da se to aproksimira.

Dosadašnje metode koje su korišćene za aproksimaciju ili/i optimizovano ažuriranje PageRanka dele se u dve grupe:

- Metode linearne algebre - koriste tehnike iz linearne i matrične algebre
- Monte Karlo metode - koriste mali broj nasumičnih šetnji po čvorovima

Među metodama iz prve grupe se nalazi npr. stepeni metod i oni pripadaju sporim metodama. Tu su i metode koje se zasnivaju na agregaciji i čija je osnovna ideja da većina promena koje se dese izaziva zahteva samo lokalne izmene te graf možemo podeliti na dva podgrafa: G koji sadrži čvorove na koje utiče promena i njegov komplement $\bar{G}$ koji sadrži sve ostale. Ažuriraju se čvorovi iz G,a zatim se rezulat prenosi na početni graf. Ova metoda takođe ume biti vrlo spora, što nam pogotovo ne odgovara u slučaju kada je potrebno ažuriranje u realnom vremenu (npr. u slučaju društvenih mreža). Osim sa brzinom,metoda iz ove grupe mogu imati problema i sa personalizacijom, kompleksnošću, preciznošću...[2]

Sa druge strane, imamo Monte Karlo metode koje su vrlo efikasne i mogu dostići visok nivo personalizacije.

[2] Detaljnije u literaturi [5]

Monte Karlo (**eng. Monte Carlo**) predstavlja grupu algoritama koje koristimo da dobijemo numeričke rezultate tako sto ponavljamo slučajne pokušaje. Najčešće se koriste u sledeća tri slučaja:

- Optimizacija
- Numerička integracija
- Generisanje uzorka kod raspodele verovatnoće – ono sto nas zanima u ovom radu

Dakle, interesuje nas - *kako generisati uzorak iz nizova raspodele verovatnoće tako da rešimo odredjene nelinearne jednacine?* Odgovor je: pomoću lanaca Markova.

Kako ćemo doći do ovih verovatnoća u našem slucaju? Po zakonu velikih brojeva, verovatnoća slučajnog događaja je približno jednaka verovatnoći tog događaja kada se njegova simulacija ponavlja veliki broj puta.

Objasnimo Monte Karlo metode za aproksimaciju globalnog PageRanka i procenimo koliko je rada potrebno da bi se njegove procene ažurirale sve vreme.

Za aproksimaciju čemo koristiti nasumične šetnje koje počinju u svakom čvoru u mreži traju sve do prvog reseta - dakle,svaka ima prosečnu dužinu $\frac{1}{\varepsilon}$.

R - broj nasumičnih šetnji iz čvora

Xv – broj prolazaka kroz čvor v u toku šetnje

Aproksimacija PageRank-a tada izgleda ovako:

$$\widetilde{\boldsymbol{\pi}_v} = \frac{Xv}{\frac{nR}{\varepsilon}}$$

Teorema2: Aproksimacija $\widetilde{\boldsymbol{\pi}_v}$ je usko rasuta oko svog ocekivanja $\boldsymbol{\pi}_v$.

Dokaz nije relevantan za ovaj rad te ga zbog kompleksnosti nećemo navoditi, a može se pronaći u literaturi [5].

Primedba: Aproksimacija je dovoljno dobra i za R=1, dakle definisano $\widetilde{\boldsymbol{\pi}_v}$ je dovoljno dobra aproksimacija stvarne PageRank vrednosti $\boldsymbol{\pi}_v$.

Analizirajmo sada koliko rada je potrebno da bi se redovno ažurirale šetnje.

Propozicija1*:* Neka je:

- (v_t, u_t) nasumični čvor koji stiže u trenutku t $(1 \leq t \leq m)$,
- M_t broj nasumičnih šetnji koje treba ažurirati u trenutku t $(1 \leq t \leq m)$,
- $outdeg_{u_t}(t)$ broj čvorova koji izlaze iz čvora u_t nakon t pristiglih čvorova

Tada važi:

$$E[M_t] \leq \frac{nR}{\varepsilon} E\left[\frac{\pi_{u_t}}{outdeg_{u_t}(t)}\right].$$

Dokaz*:* Osnovna ideja dokaza je da deo šetnje treba promeniti samo kada prolazi kroz čvor u_t i ako iz njega nasumično ide u čvor v_t. Očekivani broj poseta čvora u_t je $\frac{\pi_{u_t}}{\varepsilon}$. Za svaku ovakvu posetu verovatnoća da će šetnja morati da se izmeni je $\frac{1}{outdeg_{u_t}(t)}$.Iz prethodno navedenog možemo zaključiti da je verovatnoća da deo šetnje treba ažurirati najviše $\frac{\pi_{u_t}}{\varepsilon}\frac{1}{outdeg_{u_t}(t)}$. Šetnji ukupno ima nR. Dakle, ako primenimo linearnost očekivanja dobijamo :

$$E[M_t] \leq \sum_u nR\frac{\pi_{u_t}}{\varepsilon}\frac{1}{outdeg_{u_t}(t)}P[u_t = u]$$

$$= \frac{nR}{\varepsilon}E[\frac{\pi_{u_t}}{outdeg_{u_t}(t)}].$$

Primetimo da $E[\frac{\pi_{u_t}}{outdeg_{u_t}(t)}]$ zavisi od tačnog rasta mreže. Model koji mi pretpostavljamo ovde je nasumični permutacioni model u kom m povezanih usmerenih čvorova pristiže nasumično. Za takav model imamo sledeću lemu:

Lema1*:* Ako je (u_t, v_t) čvor koji pristiže u trenutku t $(1 \leq t \leq m)$ u nasumičnoj permutaciji čvorova,tada je:

$$E\left[\frac{\pi_{u_t}}{outdeg_{u_t}(t)}\right] = \frac{1}{t}.$$

Dokaz*:* Za nasumična dodavanja važi

$$P[u = u_t] = \frac{outdeg_u(t)}{t}.$$

Iz čega dalje možemo zaključiti:

$$E\left[\frac{\pi_{u_t}}{outdeg_{u_t}(t)}\right] = \sum_u \pi_u \frac{1}{outdeg_u(t)}P[u_t = u]$$

$$= \sum_u \pi_u \frac{1}{outdeg_u(t)}\frac{outdeg_u(t)}{t}$$

$$= \sum_u \pi_u \frac{1}{t} = \frac{1}{t} \sum_u \pi_u = \frac{1}{t}.$$

Iz Propozicije2 i Leme3 dobijamo sledeću teoremu:

Teorema3*:* Očekivana količina utrošenog rada potrebna za ažuriranje aproksimacija m pristiglih čvorova je najviše $\frac{nR}{\varepsilon^2} \ln m$.

Dokaz*:* Iz Propozicije2 znamo da je:

$$E[M_t] \leq \frac{nR}{\varepsilon} E\left[\frac{\pi_{u_t}}{outdeg_{u_t}(t)}\right],$$

a iz Leme3 znamo da je

$$E\left[\frac{\pi_{u_t}}{outdeg_{u_t}(t)}\right] = \frac{1}{t}.$$

Dakle, zaključak je:

$$E[M_t] \leq \frac{nR}{\varepsilon} \frac{1}{t}.$$

Naime, za svaki deo šetnje koji treba ažurirati možemo da krenemo šetnju počevši iz ažuriranog čvora ili jos bolje počevši iz čvora koji je izvor. Dakle za svaki deo šetnje potrebno nam je $1/\varepsilon$ utrošenog rada (što je prosečna dužina dela šetnje). Kada uključimo to u prethodno dobijeni rezultat dobijamo: $\frac{nR}{\varepsilon^2} \frac{1}{t}$.

Sumiranjem po t (za sve trenutke) dobijamo količinu ukupnog utrošenog rada koji nam je potreban u slučaju pristizanja m čvorova (da bi smo ažurirali aproksimacije konstantno) :

$$\frac{nR}{\varepsilon^2} \sum_{t=1}^{m} \frac{1}{t} = \frac{nR}{\varepsilon^2} H_m \leq \frac{nR}{\varepsilon^2} \ln m.$$

Pri čemu je H_m harmonijski broj reda m.

Do sada smo razmatrali slučaj kada čvor pristiže u graf. Razmotrimo sada obrnutu situaciju i pokažimo da možemo efikasno da obradimo izlazak čvora iz grafa:

Propozicija2: Ako mreža ima m čvorova i nasumično izabran čvor napusti graf, očekivana količina utrošenog rada potrebnog za ažuriranje delova šetnje je najviše $\frac{nR}{m\varepsilon^2}$.

Dokaz: Ako je M broj delova šetnje koje treba ažurirati i (u^*, v^*) nasumičan čvor koji napušta graf imamo:

Iz Propozicije1 :

$$E[M] \leq \frac{nR}{\varepsilon} E[\frac{\pi_{u^*}}{outdeg_{u^*}}],$$

Iz Leme1 :

$$E\left[\frac{\pi_{u^*}}{outdeg_{u^*}}\right] = \frac{1}{m}.$$

I na kraju, iz Teoreme4 možemo zaključiti da za svaki deo šetnje koji treba ažurirati količina utrošenog rada je $1/\varepsilon$.

Napomena: Ne možemo ažurirati promene u web grafu u realnom vremenu jer nemamo mogućnost da ih uvidimo osim putem ponovnog pretraživanja što nije održivo u realnom vremenu, a i pristup pojedinačnim stranicama može biti skup. Sa druge strane, sa društvenim mrežama nemamo problem jer su promene vidljive provajderu

Napomena2: Nas u stvari zanima samo k najvažnijih stranica jer će algoritam svakako naći i preporučiti njih. Jedan od načina da se uštedi pri računanju je da se fokusiramo samo na njih.

Napomena3 : Trošak ažuriranja je dodatni trošak u odnosu na trošak usled aproksimacija. Svaki čvor koji se pridruži mreži treba dodati u njenu bazu pri čemu delove nasumične šetnje možemo čuvati u drugoj bazi - „skladištu“ (više o skladištenju u sledećem odeljku).

4.4. Skladištenje podataka

Zanemarićemo „inženjerske“ komponente koje treba sačuvati (npr. šeme) i staviti fokus na čuvanje matematičkih komponenti - matrica i vektora koji se koriste u PageRanku.

Sama matrica prelaza (ili njen graf) može i ne mora stati u memoriju. Ako može, proračun se izvršava na standardni način. Ako to nije slučaj, istraživači to obično rešavaju na jedan od ova dva načina:

- Kompresuju potrebne podatke tako da mogu da stanu u memoriju, a zatim primenjuju modifikovani PageRank na kompresovanu verziju
- Ne kompresuju podatke vec razvijaju efikasne ulazne i izlazne implementacije proračuna

Kao što je već navedeno ranije, jedno od mogućih rešenja bi bilo da izbacimo viseće čvorove, ali ovo rešenje ćemo odbaciti jer time rizikujemo da izgubimo mnoge korisne linkove.

Umesto toga ćemo „poboljšati“ matricu prelaza P dodavanjem av^T gde je

$v^T = \frac{1}{n}e^T$, a a je vektor koji se sastoji od elemanata

$a_i = \begin{cases} 1, ako\ red\ i\ matrice\ P\ odgovara\ visećem\ čvoru \\ 0, u\ suprotnom \end{cases}$.

$$\bar{P} = P + av^T \Rightarrow \bar{\bar{P}} = \alpha\bar{P} + (1-\alpha)ev^T$$
$$= \alpha P + (\alpha a + (1-\alpha)e)v^T$$

Kao što smo videli malopre, ono što određuje PageRank je izračunavanje statičkog rešenja lanca Markova. Vektor reda se može odrediti na sledeće načine :

- Nalaženjem karakterističnog vektora : $\boldsymbol{\pi^T\bar{\bar{P}} = \pi^T}$
- Rešavanjem homogenog linearnog sistema : $\boldsymbol{\pi^T(I - \bar{\bar{P}}) = 0^T}$,
  gde je I jedinična matrica

U oba slučaja postoji dopuna u vidu jednačine normalizacije $\boldsymbol{\pi^T e = 1}$, gde je e jedinični vektor kolone. Ovom normalizacijom osiguravamo da je $\boldsymbol{\pi^T}$ vektor verovatnoća. I-ti elemenat tog vektora je $\boldsymbol{\pi_i}$, PageRank stranice i.

Početni izbor je bilo nalaženje karakterističnog korena za koje je korišćen prilično spor stepeni metod . Za ovakav izbor metoda postoje dobri razlozi.

Pogledajmo kako izgledaju iteracije stepenog metoda primenjene na matricu $\bar{\bar{P}}$:

Za neki početni vektor $x^{(0)T}$,

$$x^{(k)T} = x^{(k-1)T}\bar{\bar{P}} = \alpha x^{(k-1)T}\bar{P} + (1-\alpha)x^{(k-1)T}ev^T$$
$$= \alpha x^{(k-1)T}\bar{P} + (1-\alpha)v^T$$
$$= \alpha x^{(k-1)T}P + (\alpha x^{(k-1)T}a + (1-\alpha))v^T$$

Gde je $x^{(k-1)T}$ vektor verovatnoća pa je $x^{(k-1)T}e = 1$.

U ovakvom zapisu,postaje nam jasno da stepeni metod možemo primeniti na retku matricu P, a $\bar{P}$ i $\bar{\bar{P}}$ ne moraju biti formirane ni sačuvane. Ovakav metod „bez matrica“ je prilično zgodan zbog veličine weba. Kako je P retka, svako mnozenje vektora i matrica mozemo izvršiti u $nn(P)$ koraka gde je $nn(P)$ broj ne-nula u P.

5.Poređenje sa sličnim algoritmima

5.1. SALSA

SALSA (Stochastic Approach for Link-Structure Analysis) je algoritam za rangiranje web stranica pomocu *hub*-ova i *authority*-ja zasnovan na kvalitetu hiperlinkova izmedju njih.

Ideja potice od kreiranja web stranica dok se internet formirao : određene stranice (*hub*-ovi) su koriscene kao direktorijumi koji su bili „katalog“ stranica - sadržali su linkove ka njima. Dobar *hub* - onaj koji vodi do mnogih drugih stranica, dobra *authority* stranica koja je linkova od strane mnogih *hub*-ova.

Ovakva šema daje dve ocene za stranice :

- *Authority* ocenu koja ocenjuje sadržaj stranice
- *Hub* ocenu koja ocenjuje vrednost linkova ka drugim stranicama.

Matematički predstavljeno:

- *Hub* ocena : $h_v = \sum_{\{x|(v,x)\in E\}} \frac{a_x}{indeg(x)}$
- *Authority* ocena : $a_x = \sum_{\{v|(v,x)\in E\}} \frac{h_v}{outdeg(v)}$

Pri čemu je E skup svih čvorova u grafu.

Na osnovu ovih ocena formiraju se dva grafa i za svaki od njih matrica prelaza – dakle u ovom algoritmu imamo dva lanca Markova. Takođe praktikujemo nasumičnu šetnju, ali u ovom slučaju možemo da idemo i korak napred i korak nazad. Po načinu na koji smo definisali ocene možemo primetiti da postoji jaka povezanost između njih.

Aproksimacija SALSA ocena[3]

Za aproksimaciju ocena u SALSA metodu sačuvaćemo:

R nasumičnih šetnji koje kreću unapred iz čvora

\+ R nasumičnih šetnji koje kreću unazad iz čvora

= **2R nasumičnih šetnji po čvoru**

[3] Analogno sa aproksimacijom PageRanka, Teorema3,odeljak 4.3.

Na ovaj način možemo aproksimirati slično kao u slučaju PageRanka pa se aproksimacija može dokazati na sličan način.

Neka se čvor (u_t, v_t) pridružuje grafu u trenutku t. Za razliku od PageRanka gde samo u_t može inicirati ažuriranje, u ovom slučaju i u_t i v_t to mogu. Ponovo pretpostavljamo nasumični model permutacije i tada važi :

$$P[u_t = u] = \frac{outdeg_u(t)}{t}$$

$$P[v_t = v] = \frac{indeg_v(t)}{t}$$

Pri čemu je $indeg_v(t)$ broj čvorova koji imaju link ka stranici v.

Navedimo sada teoremu koja je slična **Teoremi3** iz poglavlja 4.3.:

Teorema4: Očekivana količina posla potrebnog za redovno ažuriranje aproksimacija nakon pristiglih m čvorova je najviše $\frac{16nR}{\varepsilon^2}\ln m$.

Dokaz*:* Kako su teoreme vrlo slične, slični su i njihovi dokazi tako da ćemo objasniti odakle se pojavljuje broj 16 u ovoj teoremi koji predstavlja jedinu razliku:

2 (umesto R šetnji sada ih imamo 2R)

x 2 (sada i u_t i v_t mogu inicirati ažuriranje pa duplo više šetnji treba biti ažurirano u svakom trenutku)

x 4 (svaki deo šetnje sada umesto $1/\varepsilon$ ima prošečnu dužinu $2/\varepsilon$ jer su reseti dozvoljeni samo u slučaju koraka unapred, pa umesto $(1/\varepsilon)^2$ imamo $(2/\varepsilon)^2 = 4/\varepsilon^2$)

= 16

5.2. HITS

HITS (Hyperlink-Induced Topic Search) je algoritam za rangiranje web stranica pomocu analize linkova (isto kao i PageRank i SALSA) i kao i SALSA koristi princip dve ocene (*hub*-ove i *authority*-je).

Kao što je definisano u 5.1:

- Hub ocena : $h_v = \sum_{\{x|(v,x)\in E\}} \frac{a_x}{indeg(x)}$, gde je indeg(x) broj ulaza u čvor
- Authority ocena : $a_x = \sum_{\{v|(v,x)\in E\}} \frac{h_v}{outdeg(v)}$, gde je outdeg(x) broj ulaza u čvor

Inicijalno im dodelimo vrednosti 1. Neka je A matrica povezanosti - kvadratna matrica koja ima vrednosti $Aij = \begin{cases} 1, ako\ postoji\ link\ od\ i\ ka\ j \\ 0, inače \end{cases}$ i neka je A^T njena transponovana matrica. Označimo sa $\vec{h}$ vektor svih *hub* ocena i sa $\vec{a}$ vektor svih *authority* ocena. Tada važi :

$$\vec{h} \leftarrow A\vec{a}$$

$$\vec{a} \leftarrow A^T\vec{h}.$$

Iskoristimo cikličnost ovih izraza da bismo mogli da ih zapišemo jedan preko drugog :

$$\vec{h} \leftarrow AA^T\vec{h}$$

$$\vec{a} \leftarrow A^TA\vec{a}.$$

I ako uvedemo sada nepoznate karakteristične vrednosti λ_a (za AA^T) i λ_h(za A^TA) dobijamo:

$$\vec{h} = \lambda_h AA^T\vec{h}$$

$$\vec{a} = \lambda_a A^TA\vec{a}.$$

Ovaj iterativni metod je ekvivalentan stepenom metodu koji koristimo kod PageRanka.

5.3. Poređenje algoritama

- HITS vs PageRank

HITS i PageRank su nastali u približno vreme i imaju dosta sličnosti. HITS algoritam je za razliku od PageRank algoritma imun na viseće čvorove i paukove mreže pa nije potrebno „rezanje“ da bi se zaobišli problemi koje ima PageRank. Algoritam se završava po svakom upitu (za razliku od PR). Najveća mana je što ne mogu da se koriste transponovane matrice ili redukovanje matrica da bi se povećala efikasnost. HITS ne koristi lanac Markova već matricu autoriteta i matricu hubova i kao rezultat toga vraća i autoritet i hub ocenu stranice dok PR vraća samo jednu. Ovo takođe znači i da za svaki upit HITS vrši dva računanja dok PR vrši samo jedno.

- SALSA vs PageRank

SALSA (kao i HITS) dodeljuje dve ocene svakoj web stranici (hub i authority) radi na usmerenom podgrafu koji je odredjen temom za razliku od Pageranka koji daje jedinstvenu ocenu. Kao PageRank, SALSA algoritam izracunava ocene simulirajuci nasumicnu setnju kroz lanac Markova koji predstavlja graf web stranica. Medjutim, SALSA radi sa dva lanca Markova (lanac hub-ova i lanac authority-ja), dok PageRank radi samo sa jednim – daje tezinske koeficijente na posebno na osnovu ulaznih, a posebno na osnovu izlaznih linkova. Manje je osetljiva na viseće čvorove i disjunktne grafove zbog svoje strukture.

6.Primena PageRank i personalizovanog PageRank algoritma na sisteme preporuke

6.1. Sistem za preporuku filmova

Posmatrajmo sistem koji se sastoji od grupe korisnika i grupe proizvoda gde svaki korisnik određuje koji proizvodi su mu se svideli. Uzmimo kao primer sistem za preporuku filmova[4] koji je naveden u referenci [2]. Kako se može iskoristiti PageRank za preporuku novih filmova svakom korisniku zasnovanu na interesovanjima sličnih korisnika?

Ukoliko sam korisnik ovakvog sistema, postoje dve važne činjenice pri odabiru sledećeg filma:

- Film mi je *relevantan* ako se sviđa *sličnim* korisnicima
- Korisnik mi je *sličan* ako mu se sviđaju filmovi koji su meni *relevantni*

Time dobijamo ciklično definisanje kao kod PageRanka, ali smo kod njega takođe videli da možemo da napravimo nezavisne jednačine zasnovane na tome i da zatim rešavamo sistem jednačina uz odgovarajuće uslove.

Kao i kod PageRanka, krenimo sa najjednostavnijom idejom:

$$rel(m) = \sum_{osobe\ i\ koje\ vole\ film\ m} sim(i),$$

$$sim(i) = \sum_{filmovi\ m\ koji\ se\ sviđaju\ osobi\ i} rel(m).$$

Analogno sa **Idejom1** – ne možemo tretirati isto kada se nekome sviđa dosta proizvoda i kada se nekome sviđa njih nekoliko, već ćemo više vrednovati preporuke te druge osobe.

Upravo zato ćemo postupiti analogno sa **Idejom2** – dodaćemo normalizaciju.

Normalizovaćemo relevantnost filma sa brojem ljudi kojima se sviđa:

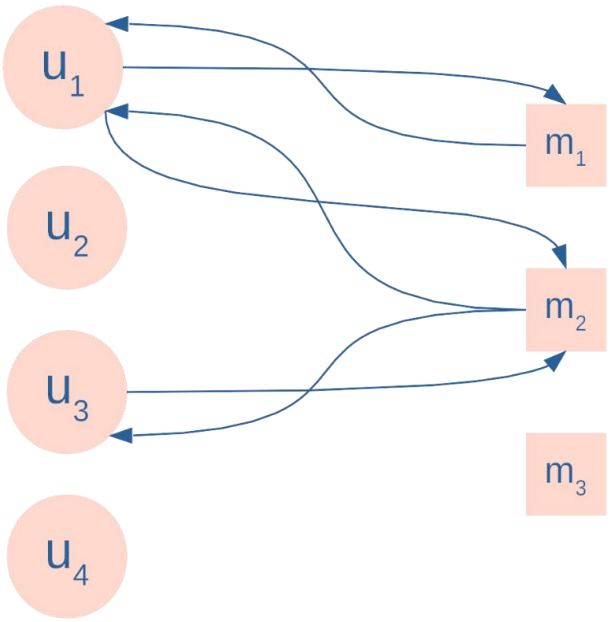
$$rel(m) = \sum_{osobe\ i\ koje\ vole\ film\ m} \frac{sim(i)}{broj\ filmova\ koji\ mu\ se\ sviđaju}$$

[4] U navedenom sistemu imamo korisnike i filmove i relacije među njima su tako uređene da svaki korisnik bira filmove koji mu se sviđaju i korisnike sa sličnim ukusom na osnovu filmova koji se njima sviđaju.

i normalizovaćemo sviđanja korisnika sa brojem filmova koji im se sviđaju:

$$sim(i) = \sum_{filmovi\ m\ koji\ se\ sviđaju\ osobi\ i} \frac{rel(m)}{broj\ korisnika\ kojima\ se\ sviđa}.$$

Sada možemo (slično kao u PageRank algoritmu) da sprovedemo interpretaciju nasumične šetnje za navedene jednačine. Posmatraćemo bipartitivan graf sa dve grupe čvorova: grupu korisnika i grupu proizvoda. Veze između njih idu u oba pravca: ako se korisniku u sviđa film m, jedna strelica ide od čvora u ka čvoru m i jedna obrnuto, od čvora m ka čvoru u.



Slika7 – Graf sa grupom korisnika u_i i grupom proizvoda m_i

Na osnovu prethodno navedenog, nije teško zaključiti da prethodne dve jednačine predstavljaju verovatnoće nasumične šetnje po bipartitivnom grafu.

Ako ponovo napravimo analogiju sa **Idejom2**, shvatićemo šta nam je potrebno da bismo mogli da rešimo ove jednačine - dodavanje uslova koji će obezbediti dobru definisanost :

$$\sum_{svi\ proivodi\ na\ tržištu} rel(m) = 1.$$

Ovim smo definisali globalni rank za filmove. Međutim, često želimo da rankiramo filmove u odnosu na nekog korisnika (npr. korisnika u). Da bismo to mogli, uključićemo teleport u nasumične šetnje na sledeći način : kada smo u čvoru koji predstavlja

proizvod, verovatnoća da se vratimo u čvor u je ε, dok je verovatnoća da nastavimo sa nasumičnom šetnjom $1-\varepsilon$. Ovakav pristup omogućava korisnicima da potraže filmove korisnika koji su im interesantni. Verovatnoće u ovom slučaju izgledaju ovako:

$$rel(m) = \sum_{i\ kojima\ se\ sviđa\ m} \frac{sim(i)}{broj\ filmova\ koji\ mu\ se\ sviđaju},$$

$$sim(i) = (1-\varepsilon)\sum_{m\ koji\ se\ sviđaju\ i} \frac{rel(m)}{broj\ korisnika\ kojima\ se\ sviđa} + \begin{cases} \varepsilon, i = u \\ 0, inače \end{cases}.$$

Na ovaj način je rešen problem sa disjunktnim grafovima (kao i kod PageRank-a).

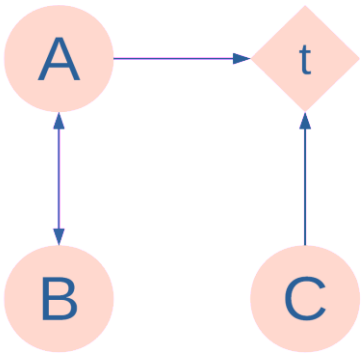
6.2. Socijalna mreža

Na primeru socijalne mreže [5] iz reference [2] ćemo preko korisnika i njihovih objava detaljnije objasniti gore navedeni princip proizvođača i potrošača.

Za rangiranje objava možemo koristiti PageRank, a za rangiranje korisnika personalizovani PageRank.

Ako predstavimo socijalnu mrežu kao graf, neka korisnici i objave budu predstavljeni kao čvorovi. Veze izmedju njih neka postoje u slučaju praćenja, objavljivanja i deljenja objava.

Primer8 : Graf socijalne mreže



Korisnik B prati korisnika A ; korisnik A je objavio t, a korisnik C je podelio tu objavu na svom profilu.

Slika8 - Jednostavan graf socijalne mreže

Iskoristimo **6.1.Sistem za preporuku filmova** kao ideju za algoritam koji se koristi za preporučivanje prijatelja na socijalnoj mreži. Sve korisnike socijalne mreže možemo posmatrati kao proizvođače i potrošače pa ćemo za svakog napraviti njihove sopstvene kopije : proizvođač-kopiju i potrošač-kopiju. Ukoliko u prati v, u je potrošač, a v je proizvođač. Dakle na bipartitivnom grafu ćemo imati strelicu od u do v i stralicu od v do u. Primenom personalizovanog PageRank algoritma dobijamo preporučenog novog prijatelja (proizvođača) za svakog korisnika.

[5] Kao i kod većine socijalnih mreža, svaki korisnik ima svoj profil na kome može da objavljuje željeni sadržaj ili da deli sadržaj koji je neko drugi objavio. Konekciju između korisnika nazivamo „praćenje“ ’ dakle ukoliko korisniku sviđaju objave nekom drugog korisnika,na njegovom profilu može označiti da želi da ga prati, tj. da dobija obaveštenja o njegovim objavama.

7. Zaključak

Algoritam PageRank svakako predstavlja osnovu najpoznatijeg pretraživača, ali od njega zavisi samo određeni procenat rangiranja, dok ostatak zavisi od organizacije i sadržaja koda na sajtu.

Svakim danom se pojavljuje novi način za narušavanje ranga, a kao odgovor na to i nova pravila i zahtevi koje korisnici sa svojim sajtovima treba da poštuju. Poslednjih godina pojavila se potpuno odvojena grana kompjuterskih nauka koja se bavi vidljivošću na webu.

PageRank vrednosti možemo najlakše utvrditi uz pomoć Google ToolBar-a koji pokazuje vrednost između 0 i 10. Google ToolBar ne pokazuje verovatnoću, već tačan rang u vrednostima od 0 do10 koji su definisani po logaritamskoj skali (PR 4-5 je tipičan za većinu sajtova sa prosečnom popularnošću, PR 6 imaju veoma popularni web sajtovi, PR 7 je gotovo nedostižan za obične webmastere, PR 8-9-10 dostižu web sajtovi velikih kompanija kao što su Google, FaceBook itd.).

Pored rangiranja u pretrazi, značajnu ulogu igra i personalizacija pretrage. Personalizacija Googla je posebno osetljiva tema. Po nekim mišljenjima otišla je predaleko jer se kao parametri za personalizaciju koriste sve privatniji podaci. Možemo primetiti da reklame koje nam iskaču prilikom korišćenja raznih aplikacija su u dobroj meri povezane sa pojmovima koje smo „guglali“ i stranicama koje smo posećivali. Balansiranje između poverljivosti informacija i što bolje personalizacije sigurno nije jednostavan posao.

Činjenica je da ovaj algoritam povezuje pretraživanje, oglašavanje i sisteme preporuke. Njegov najjači adut je sposobnost procenjivanja „mera“ mreže. Ostaje da vidimo kako će se sa ovim izazovima (i možda nekim novim) suočavati ovaj algoritam u budućnosti i kako će ih prevazilaziti.

Literatura

[1] Amy N.Langville, Carl D.Mayer, *Deeper inside PageRank,* Internet Mathematics, 2011.

[2] Ashish Goel, *Application of PageRank to Recommendation Systems*, Twitter Inc. and Stanford University, 2013.

[3] Sergey Brin, Lawrence Page, *The anatomy of a large-scale hypertextual Web search engine*, Stanford University, 1998.

[4] Sergey Brin, Lawrence Page,R.Motwami, Terrz Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Stanford University, 1998.

[5] Bahman Bahmani, Abdur Chowdhury, Ashish Goel , *Fast Incremental and Personalized PageRank,* Twitter Inc. and Stanford University, 2010.

[6] Chinng WK., Ng M., *Markov Chains: Models, Algorithms and Applications*, Springer, 2006.

[7] Tibshirani R., *PageRank*, Carnegie Mellon University, 2013.

[8] Danijela Rajter-Ćirić, *Verovatnoća*. Univerzitet u Novom Sadu. Prirodno-matematički fakultet u Novom Sadu. 2008.

[9] Danijela Rajter-Ćirić, *Stohastička analiza*, beleške sa predavanja, 2015

[10] Vladimir Baltić, Dragan Stevanović, Marko Milošević, *DISKRETNA MATEMATIKA, osnove kombinatorike i teorije grafova*, Društvo matematičara Srbije, Beograd, 2004.

[11] Lempel, R.; Moran S. (April 2001). "SALSA: The Stochastic Approach for Link-Structure Analysis". ACM Transactions on Information Systems, 2001

Biografija

Milica Višekruna je rođena 19. aprila 1991. godine u Zrenjaninu. U svom rodnom gradu je završila osnovnu školu i gimanziju, prirodno-matematički smer. Nakon toga upisala je Prirodno-matematicki fakultet u Novom Sadu, smer primenjena matematika, modul matematika finansija, gde je 2015. godine završila osnovne studije. Iste godine upisuje i master studije na istom fakultetu, isti smer. Od maja 2017. radi kao verifikacioni inženjer u „HDL Design House“ u Beogradu na verifikaciji mikročipova.

Milica Višekruna

Novi Sad, 2019.

UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET
KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj:
RBR

Identifikacioni broj:
IBR

Tip dokumentacije: *monografska dokumentacija*
TD

Tip zapisa: *tekstualni štampani materijal*
TZ

Vrsta rada: *master rad*
VR

Autor: *Milica Višekruna*
AU

Mentor: *dr Dušan Jakovetić*
MN

Naslov rada: *Primena teorije Markovljevih lanaca na analizu i sintezu algoritma „PageRank“*
NR

Jezik publikacije: *srpski (latinica)*
JP

Jezik izvoda: *s/e*
JI

Zemlja publikovanja: *Republika Srbija*
ZP

Uže geografsko područje: *Vojvodina*
UGP

Godina: *2019.*
GO

Izdavač: *autorski reprint*
IZ

Mesto i adresa: Novi Sad, Trg Dositeja Obradovića 4
MA

Fizički opis rada: *7 poglavlja, 50 stranica, 8 slika, 1 tabela*
FO

Naučna oblast: *matematika*

NO

Naučna disciplina: *algoritam*
ND

Ključne reči: *pagerank, algoritam, Markov lanac, teorija grafova*
UDK

Čuva se: *u biblioteci Departmana za matematiku i informatiku, Prirodnomatematičkog fakulteta, u Novom Sadu*
ČU

Važna napomena:
VN

Izvod: *U master radu je opisan PageRank algoritam. Navedene su njegove verzije od naivnog do modela slučajnog surfera. Za izračunavanje samog PageRank vektora korišćeni su lanci Markova i stepeni metod. Takođe su date i dve verzije koda za njegovo izračunavanje. Navedeni su problemi pri izračunavanju i njihovo rešenje. Predstavljena su poboljšanja u pravcu aproksimacije, ažuriranja i skladištenja podataka. Napravljeno je poređenje sa sličnim algoritmima i opisana primena na sisteme preporuke.*
IZ

Datum prihvatanja teme od strane NN veća:
DP

Datum odbrane:
DO

Članovi komisije:
KO
Predsednik: *dr Nataša Krejić, redovni profesor*
Član: *dr Dragana Bajović, docent na FTN*
Član: *dr Dušan Jakovetić, docent*

UNIVERSITY OF NOVI SAD
FACULTY OF SCIENCES
KEY WORD DOCUMENTATION

Accession number:
ANO

Identification number:
INO

Document type: *monograph type*
DT

Type of record: *printed text*
TR

Contents code: *Master thesis*
CC

Author: *Milica Višekruna*
AU

Mentor: *Dušan Jakovetić, PhD*
MN

Title: *Application of Markov chain theory on analysis and synthesis of Pagerank algorithm*
XI

Language of text: *Serbian (latin)*
LT

Language of abstract: *s/e*
LA

Country of publication: *Republic of Serbia*
CP

Locality of publication: *Vojvodina*
LP

Publication year: *2019*
PY

Publisher: *author's reprint*
PU

Publ. place: *Novi Sad, Trg Dositeja Obradovića 4*
PP

Physical description: *7 chapters, 50 pages, 8 pictures, 1 table*
PD

Scientific field: *mathematics*
SF

Scientific discipline: algorithm
SD

Key words: pagerank, algorithm, Markov chain, graph theory
UC

Holding data: *Department of Mathematics and Informatics' Library, Faculty of Sciences, Novi Sad*
HD

Note:
N

Abstract: *In this master thesis is presented the PageRank algorithm. His versions are given from naive PageRank to random surfer model. Markov chains and power method are used for computation of PageRank vector. Also, there are two verions of code for computation. Problems for computation are named and is given solution for them. Improvements in fileds of aproximation, update and data storage are given as well. Comparation with similar algorithms is also there and appilication on recommendation systems is described.*
AB

Accepted by the Scientific Board on:
ASB

Defended:
DE

Thesis defended board:
DB
President: *Nataša Krejić, PhD, full professor*
Member: *Dragana Bajović, PhD, assistant professor* at FTN
Member: *Dušan Jakovetić, PhD, assistant professor*