

Luku 9. Relevanttien sivujen etsintä verkosta: satunnaiskulut verkossa

T-61.2010 Datasta tietoon, syksy 2011

professori Heikki Mannila

Tietojenkäsittelytieteen laitos, Aalto-yliopisto

1.12.2011

Tämän luvun sisältö

1 Relevanttien sivujen etsintä verkosta: satunnaiskulut verkossa

- Webin lyhyt historia
- Etsintä verkosta
- Keskukset ja auktoriteetit
- PageRank

Muista T-ohjelman kurssipalaute (+1 piste joului- ja tammikuun tenttiin):

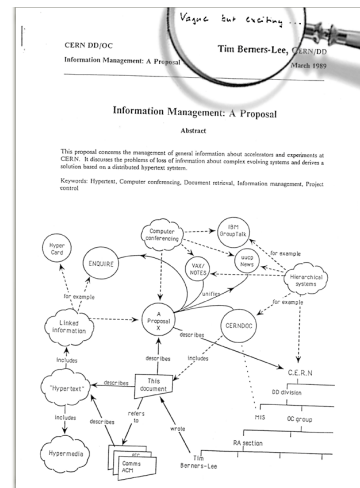
► <http://www.cs.hut.fi/Opinnot/Palaute/kurssipalaute.html>

Webin lyhyt historia

1989: Tim Berners-Lee

- Maaliskuussa 1989 Tim Berners-Lee jätti ehdotuksen CERNin tiedonhallintajärjestelmäksi
- Esimiehen kommentti: “epämääräinen mutta kiinnostava...”

► Proposal.html

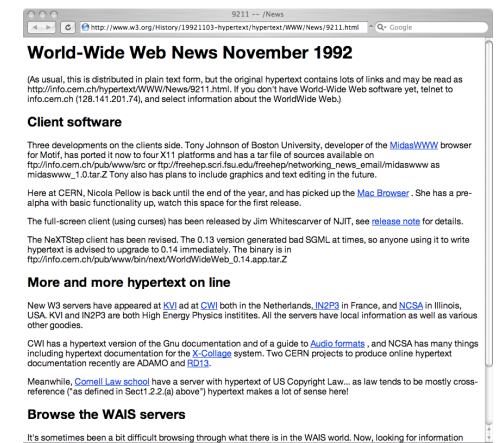


Webin lyhyt historia

1992: ensimmäisiä selaimia

- Syksyllä 1992
info.cern.ch kertoi
uusista palvelimista
ja sivuista

► <http://www.w3.org/History/...>

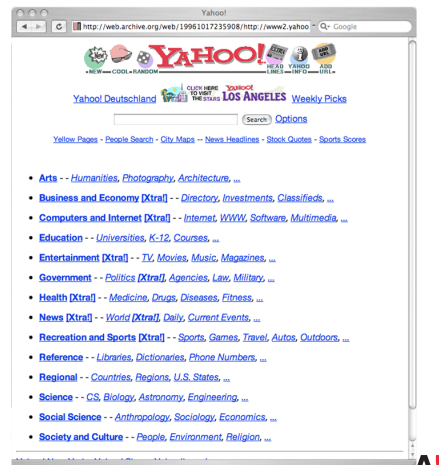


Webin lyhyt historia

Linkkikokoelmia

- Jerry and David's Guide to the World Wide Web. Myöhemmin nimellä Yahoo; iso käsin ylläpidetty lista webbiosoitteita.

▶ vanhoja sivustoja



5/31

Webin lyhyt historia

Altavista oli 1990-luvun Hakukone

- Digital teki Altavistan mainostaakseen servereitään: "Näillä voi indeksoida vaikka koko webin."
- Altavistalla webbisivujen etsiminen muuttui paljon helpommaksi, mutta vastaukset tulivat melko satunnaisessa järjestyksessä.

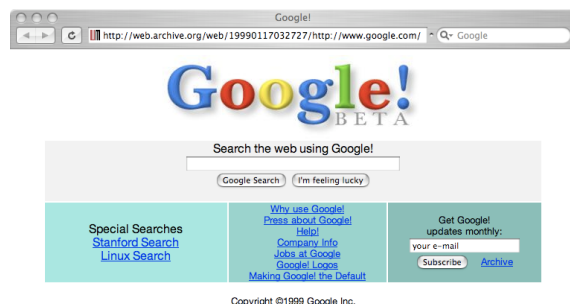


6/31

Webin lyhyt historia

1998: Google

- Google keksi, miten hakutulokset järjestetään hyödyllisyyden mukaan.



7/31

Etsintä verkosta

- Miten löydetään verkosta hyviä sivuja, jotka kertovat vaikkapa tiedon louhinnasta?
- Ensimmäiset menetelmät skaalautuivat huonosti:
 - info.cern.ch: ilmoitustaulu uusista sivuista
 - Yahoo: käsin toimitettu hakemisto
- Annettuna kysely "data mining"
- Miten löydetään automaattisesti sivut, joilla nuo sanat esiintyvät (tai esiintyvät peräkkäin)?
- Miten löydetään näistä sivuista hyödyllisimmät?

8/31

Etsintä verkosta

Sanahaku

- *Merkkijonomenetelmät* (“string algorithms”): paljon hauskoja algoritmeja
- Esivalmisteluna käydään koko verkko läpi seuraamalla linkkejä
- Kullakin sanalla X tallennetaan lista sivuista, joilla X esiintyy
- Kyselyyn “ X ja Y ” vastatessa käydään kummankin sanan osoitelistat läpi ja palautetaan ne osoitteet, jotka esiintyvät molemmissa.
- Käytännössä hankalampaa kuin näyttää paperilla: dataa on paljon, vastaus pitäisi saada alle sekunnissa.
- Paljon mielenkiintoista tekniikkaa, katso esim.

► <http://research.google.com/pubs/papers.html>



9/31

Etsintä verkosta

Relevanssi

- “Results 1 – 10 of about 50,300,000”
- Vanha tapa ratkaista ongelma: käyttäjä tarkentaa kyselyä lisäämällä ovelasti valittuja sanoja ja operaattoreilla AND, OR, NOT, NEAR.
- Miten valitaan ne sivut, jotka näytetään ensimmäiseksi? Heuristiikkoja:
 - Sanojen esiintymisfrekvenssi (huono idea)
 - Sivun on hyvä, jos siihen viitataan paljon
 - Sivun on hyvä lähde asiasta X , jos X mainitaan sivuun viittaavissa linkeissä (eli *ihminen* on tehnyt jo arvioinnin)



10/31

Keskukset ja auktoriteetit

- “Hubs and authorities”: Jon M. Kleinberg: Authoritative Sources in a Hyperlinked Environment, IBM Research Report 1997; SODA 1998
- <http://scholar.google.fi/> →
- Hyvillä auktoriteetti-sivuilla on paljon *yhteisiä* linkittäjiä eli keskuksia.
- Hyvä keskus osoittaa hyvää arvostelukykä linkittämällä hyviin auktoriteetteihin

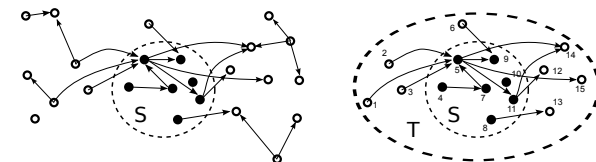


11/31

Keskukset ja auktoriteetit

Etsitään ensin parhaiden sivujen ja linkkien joukot

- Etsitään ensin sivut, joilla “ X ja Y ” esiintyy
- Otetaan näistä heuristiikkojen perusteella esim. 200 parasta; olkoon tämä sivujen *ydinjoukko* S
- Muodostetaan joukko T : S ja sivut jotka viittaavat johonkin joukon S sivuun ja sivut joihin jokin S :n sivu viittaa (katso kuva alla)



Kuva: Keskukset ja auktoriteetit. Ydinjoukko S laajennettuna sivuilla, joihin tai joista on linkki.



12/31

Keskukset ja auktoriteetit (2)

Etsitään ensin parhaiden sivujen ja linkkien joukot

- Tarkastellaan joukkoa T verkkona: solmuina sivut, kaarina linkit (suunnattuja kaaria)
- Olkoon E verkon T kaarien joukko: $(u, v) \in E$ kun sivulta u on linkki sivulle v
- Pelkkä sivuun osoittavien linkkien määrä ei ole kovin hyvä relevanssin mittari



13 / 31

Keskukset ja auktoriteetit

Periaate

- *Periaate:*
 - Hyvä keskus osoittaa hyviin auktoriteetteihin
 - Hyviin auktoriteetteihin tulee linkkejä hyvistä keskuksista
- Kehämääritelmä?



14 / 31

Keskukset ja auktoriteetit

Hyvien keskusten ja auktoriteettien laskenta iteratiivisesti

- Kehämääritelmästä selvittää iteratiivisella menetelmällä (vrt. c-means, Luku 6)
- Kullekin joukon T sivulle s määritellään keskuspaino k_s ja auktoriteettipaino a_s
- Tarvitaan jotkin alkuarvot:

$$a_s = k_s = \frac{1}{\sqrt{n}}, \quad \forall s \in T$$

- Iteratiiviset päivityssäännöt:

$$a_s \leftarrow \sum_{t \in T, (t,s) \in E} k_t, \quad k_s \leftarrow \sum_{t \in T, (s,t) \in E} a_t$$



15 / 31

Keskukset ja auktoriteetit (2)

Hyvien keskusten ja auktoriteettien laskenta iteratiivisesti

- Eli sivun s auktoriteettipainoa varten summataan kaikki keskuspainot niistä sivuista, joista on linkki sivulle s . Sivun s keskuspaino päivitetään vastaavasti summaamalla sivun s linkkaamien sivujen auktoriteettipainot
- Joka iteraatiossa skaalataan painojen neliöiden summat ykkösiksi

$$\sum_{s \in T} a_s^2 = 1, \quad \sum_{s \in T} k_s^2 = 1$$



16 / 31

Keskukset ja auktoriteetit

Hyvien keskusten ja auktoriteettien laskenta matriiseilla

- Edellä saadut iterointikaavat voidaan kirjoittaa helposti myös matriisimuodossa
- Ajatellaan painoja vektoreina

$$\mathbf{a} = (a_s)_{s \in T}, \quad \mathbf{k} = (k_s)_{s \in T}$$

ja verkko matriisina M

$$M = (1, \text{ jos } (s, t) \in E, \text{ muuten } 0)$$

- Nyt iteraation päivityssäännöt voidaan kirjoittaa:

$$\mathbf{a} \leftarrow M^T \mathbf{k}, \quad \mathbf{k} \leftarrow M \mathbf{a}$$



17 / 31

Keskukset ja auktoriteetit (2)

Hyvien keskusten ja auktoriteettien laskenta matriiseilla

- Ottaen huomioon skaalauksen ykköseksi voidaan kirjoittaa koko iterointisääntö:

$$\mathbf{a} \leftarrow \frac{M^T \mathbf{k}}{\|M^T \mathbf{k}\|}, \quad \mathbf{k} \leftarrow \frac{M \mathbf{a}}{\|M \mathbf{a}\|}$$

- Ajetaan iteraatiota läpi muutama kierros, kun alustuksena **1**

$$\begin{cases} \mathbf{a} \leftarrow \frac{1}{\|M^T \mathbf{1}\|} M^T \mathbf{1} \\ \mathbf{k} \leftarrow \frac{1}{\|M \mathbf{1}\|} M \mathbf{1} \end{cases} \quad \begin{cases} \mathbf{a} \leftarrow \frac{1}{\|\dots\|} M M^T \mathbf{1} \\ \mathbf{k} \leftarrow \frac{1}{\|\dots\|} M^T M \mathbf{1} \end{cases} \quad \begin{cases} \mathbf{a} \leftarrow \frac{1}{\|\dots\|} M^T M M^T \mathbf{1} \\ \mathbf{k} \leftarrow \frac{1}{\|\dots\|} M M^T M \mathbf{1} \end{cases}$$

- Siis i :n iteraation jälkeen

$$\mathbf{a} = \frac{1}{\|\dots\|} (M^T M)^i \mathbf{1}$$



18 / 31

Keskukset ja auktoriteetit (3)

Hyvien keskusten ja auktoriteettien laskenta matriiseilla

- Pieni esimerkki
- Matriisi $M^T M$ on symmetrinen, joten sillä on reaaliset ominaisarvot ja -vektorit ja se voidaan diagonalisoida

$$M^T M = V D^2 V^T$$

missä $D = \text{diag}(\lambda_1, \dots, \lambda_n)$

- Teknisellä oletuksella $\lambda_1 > \lambda_2$ saadaan

$$(M^T M)^i \mathbf{1} = V D^i V^T \mathbf{1} \approx V \cdot \text{diag}(\lambda_1^i, 0, \dots, 0) \cdot V^T \mathbf{1} = (\lambda_1^i \mathbf{v}_1^T \mathbf{1}) \mathbf{v}_1$$

- Siis $\mathbf{a}$ on matriisin $M^T M$ suurinta ominaisarvoa vastaava ominaisvektori
- Vastaavasti $\mathbf{k}$ on matriisin $M M^T$ ominaisvektori
- Ominaisvektorit voitaisiin esim. Matlabissa funktiolla $[V, D] = \text{eig}(M' * M)$



19 / 31

Keskukset ja auktoriteetit

Tuloksia Kleinbergin paperista

- Auktoriteetteja (ylempi kuva)
- Toinen sovellus (alempi kuva): Samankaltaisten sivujen löytäminen: sen sijasta, että aloitettaisiin sanahaun löytämistä sivuista, aloitetaan johonkin sivuun linkittävistä sivuista.

(java) Authorities 328 http://www.gamelan.com/ 353 http://java.siu.com/ 190 http://www.digitalfocus.com/digitalfocus/faq/howdoi.html 190 http://lightyear.ncsu.edu/~srp/java/javabooks.html 183 http://suisite.unc.edu/java/faq/faq.html	Gamelan JavaSoft Home Page The Java Developer: How Do I... The Java Book Pages comp.lang.java FAQ
(ownership) Authorities 378 http://www.eff.org/ 344 http://www.eff.org/blueribbon.html 238 http://www.cdt.org/ 235 http://www.votw.org/ 218 http://www.achu.org/	EFFweb - The Electronic Frontier Foundation The Blue Ribbon Campaign for Online Free Speech The Center for Democracy and Technology Voters Telecommunications Watch ACLU: American Civil Liberties Union
(www.honda.com) Authorities .202 http://www.toyota.com/ .199 http://www.honda.com/ .192 http://www.ford.com/ .173 http://www.bmwusa.com/ .162 http://www.volvocars.com/ .158 http://www.saturncars.com/ .155 http://www.nissannmotors.com/ .145 http://www.audi.com/	Welcome to @Toyota Honda Ford Motor Company BMW of North America, Inc. VOLVO Welcome to the Saturn Web Site NISSAN - ENJOY THE RIDE Audi Homepage



20 / 31

PageRank

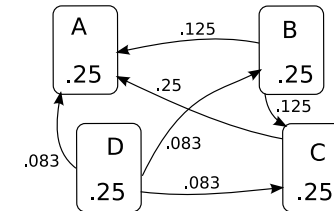
- Sergey Brin and Larry Page, 1998 (Google)
 - ▶ The Anatomy of a Large-Scale Hypertextual Web Search Engine
 - ▶ Wikipedia: PageRank
- Sivu on relevantti, jos relevantit sivut viittaavat siihen.
- Kehämääritelmä? (Jälleen tehdään iteratiivinen menetelmä)



21 / 31

PageRank Laskenta

- Jos sivulta t on linkit sivuille $s_1, s_2, \dots, s_k$, sivun t relevanssista r välittyy $1/k$ jokaiselle näistä sivuista
- Esimerkiksi universumin neljä sivua, niiden nykyiset relevanssit, linkit ja välittyvät relevanssit



22 / 31

PageRank (2) Laskenta

- Siis relevanssivektori ("PageRank") $\mathbf{r} \leftarrow F\mathbf{r}$, missä matriisin F määrittelee

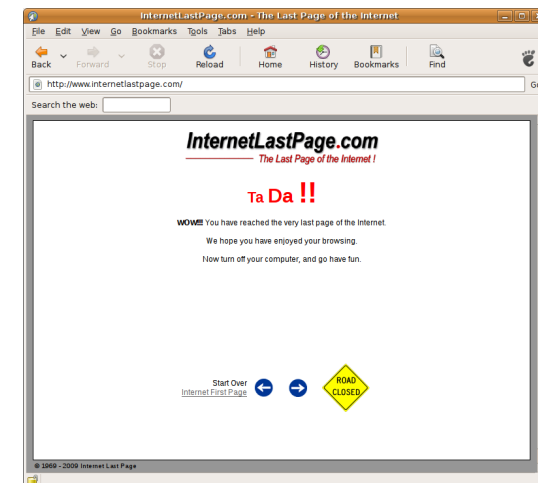
$$F_{s,t} = \begin{cases} 1/\deg(t), & \text{jos } t\text{:stä on linkki } s\text{:ään} \\ 0, & \text{muuten} \end{cases}$$

- jossa $\deg(t)$ on linkkien määrä eli k yllä
- Sopivilla oletuksilla $\mathbf{r}$ on F :n suurinta ominaisarvoa vastaava ominaisvektori, joten iteraatio konvergoi
- Toinen tulkinta: Satunnainen surffailija aloittaa joltain sivulta ja seuraa linkkejä satunnaisesti. Sivun relevanssi on todennäköisyys, jolla surffailija päätyy sivulle (pitkän ajan kuluessa).
- Entä jos...



23 / 31

PageRank (3) Laskenta



24 / 31

PageRank (4)

Laskenta

- Jos sivusta ei ole linkkejä minnekään, lisätään linkit kaikkialle. Tässä n on sivujen määrä.

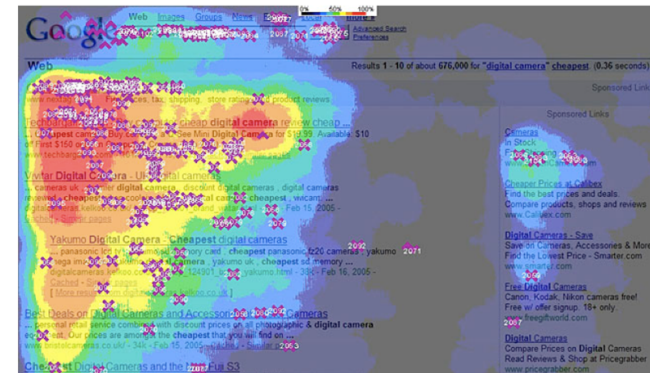
$$F_{s,t} = \begin{cases} 1/\deg(t), & \text{jos } t\text{:stä on linkki } s\text{:ään} \\ 1/n, & \text{jos } t\text{:stä ei ole linkkejä minnekään} \\ 0, & \text{muuten} \end{cases}$$

- Lisätään linkkejä saman tien kaikkialle: sallitaan surffaajan joskus kyllästyä (lisäksi vältetään teknisiltä hankaluuksilta konvergenssitodistuksessa) ▶ Pieni esimerkki

$$F_{s,t} = \begin{cases} \frac{1-\epsilon}{\deg(t)} + \frac{\epsilon}{n}, & \text{jos } t\text{:stä on linkki } s\text{:ään} \\ 1/n, & \text{jos } t\text{:stä ei ole linkkejä minnekään} \\ \epsilon/n, & \text{muuten} \end{cases}$$

PageRank (5)

Laskenta



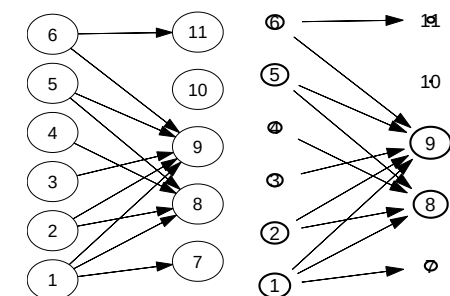
Altavistan käyttäjät jaksivat lukea tuloksia monelta sivulta, Googlen käyttäjät katsovat vain muutamaa ensimmäistä

Yhteenveto

Tässä luvussa keskusteltiin merkitsevien verkkosivujen löytämisestä sivujen linkityksen perusteella. Keskusten ja auktoriteettien algoritmissa periaattena on, että hyvät keskukset linkittävät hyvin auktoriteetteihin ja vastaavasti hyvin auktoriteetteihin tulee linkkejä hyvistä keskuksista. PageRank-algoritmi laskee sivujen relevanssia "satunnaisen surffailijan" tavoin. Algoritmi on käytössä Googlessa.

Esimerkki keskuksista ja auktoriteeteista

Tutkitaan oheista verkkoa, jonka kaarien matriisi M , $M(i, j) = 1$, jos olemassa kaari (s_i, s_j) . Oikeanpuoleisessa kuvassa vasemman puolen solmujen koko viittaa laskettuun keskuspainoon ja oikean puolen solmujen koko auktoriteettipainoon. Seuraavilla kalvoilla Matlab-koodi ja tulokset.



Kuva: Keskukset ja auktoriteetit. (a) Lähtötilanne. (b) Keskuspainot $k_1, \dots, k_6$ ja auktoriteettipainot $a_7, \dots, a_{11}$

Esimerkki keskuksista ja auktoriteeteista

Matlab-koodi

```
M = [ 0 0 0 0 0 0 1 1 1 0 0; <M-matriisi> ]; % M(i,j)=1 <=> s_i-->s_j

n = size(M,1);
a = ones(n,1)/sqrt(n); % auktoriteettipainot
k = ones(n,1)/sqrt(n); % keskuspainot

for s = 1 : 20
    aold = a; kold = k;
    anew = M'*k; knew = M*a; % paivitys
    a = anew/sqrt(sum(anew.^2)); % skaalaus
    k = knew/sqrt(sum(knew.^2));

    disp(['Kierros: ' num2str(s)]);
    disp(['Muutos: ' num2str(sum((a-aold).^2)) ', ' num2str(sum((k-kold).^2))]);
    disp(['# aukt kesk alku']);
    disp([ (1:n)' a k ones(n,1)/sqrt(n) ]);
    pause;
end
```



29 / 31

Esimerkki keskuksista ja auktoriteeteista

Tulokset

```
Kierros: 20
Muutos: 1.2917e-15, 1.911e-14
```

#	aukt	kesk	alku
1.0000	0	0.5583	0.3015
2.0000	0	0.4877	0.3015
3.0000	0	0.2659	0.3015
4.0000	0	0.2219	0.3015
5.0000	0	0.4877	0.3015
6.0000	0	0.3043	0.3015
7.0000	0.1985	0	0.3015
8.0000	0.6241	0	0.3015
9.0000	0.7479	0	0.3015
10.0000	0	0	0.3015
11.0000	0.1082	0	0.3015

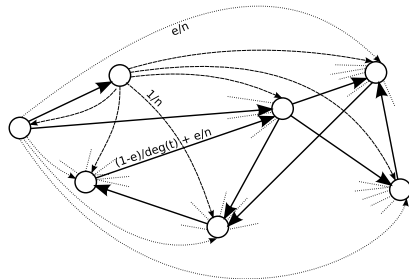
◀ Takaisin kalvoihin



30 / 31

Esimerkki PageRank:sta

$$F_{s,t} = \begin{cases} \frac{1-\epsilon}{\deg(t)} + \frac{\epsilon}{n}, & \text{jos } t\text{:stä on linkki } s\text{:ään} \\ 1/n, & \text{jos } t\text{:stä ei ole linkkejä minnekään} \\ \epsilon/n, & \text{muuten} \end{cases}$$



Kuva: PageRank-esimerkki. Varsinaiset linkit paksummalla viivalla ja nuolella.



31 / 31

◀ Takaisin kalvoihin