Multimédiás és webes adatbányászat

KISS LÁSZLÓ

Tartalom

Webes keresések kezdete

PageRank
- Alapok
- Számítása a valóságban
- Topic-Sensitive PageRank
- Trust Rank

Egyéb algoritmusok
- HITS
- Google Panda
- Google Hummingbird

Webes keresések kezdete

Mikor publikálták az első kereső motort?

?

Mikor publikálták az első kereső motort?

1990

Pár szó a kezdetekről

1990:

- Archie kereső motor
  - McGill University, Montreal
  - Publikus FTP-n elérhető fájlok listázása és kereshetővé tétele

1993:

- JumpStation
  - Jonathon Fletcher, skót egyetemista
  - Első „rendes” kereső: Index, webet feltérképező botok és kereshetőség

…

1998:

- Google

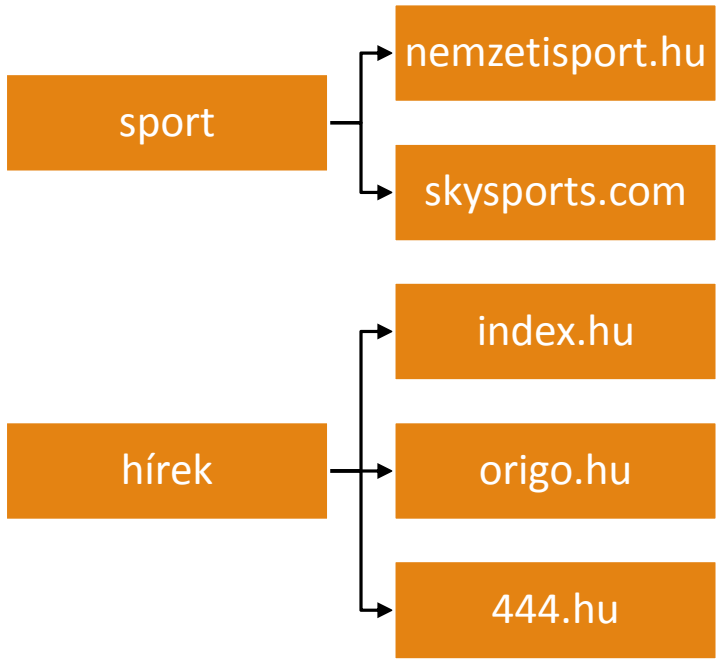
Keresések kezdete

Miért a Google?

- Nem volt az első
- DE a legnépszerűbb

Korai keresők

- Mászkálnak a weben
- Invertált indexeket építenek
- Mi a baj vele?



Term Spam

Hogyan priorizáltak a korai keresők?

- Kifejezések száma
- Előfordulási helyük

Ezek megtévesztésére: Term Spam

- Sok-sok népszerű kifejezés beillesztése
- Láthatatlan is lehet
- Népszerű oldalak tartalma az oldalon belül

Random surferök

Valós felhasználók szimulálása

- Navigálás linkek mentén
- Kezdés véletlenszerűen

Több surfer egy oldalon -> Fontosabb

Mit tett a Google?

Page Rank

- Több random surfer -> magasabb PageRank

Kifejezések az oldalra mutató linkekben

- Könnyű hamis kifejezéseket tenni az oldalra
- Nehezebb a szomszédos oldalakra

PageRank

ALAPOK

PageRank

Név <- Larry Page

Egy függvény, amely valós számot rendel minden weboldalhoz (amiket feltérképezett az algoritmus).

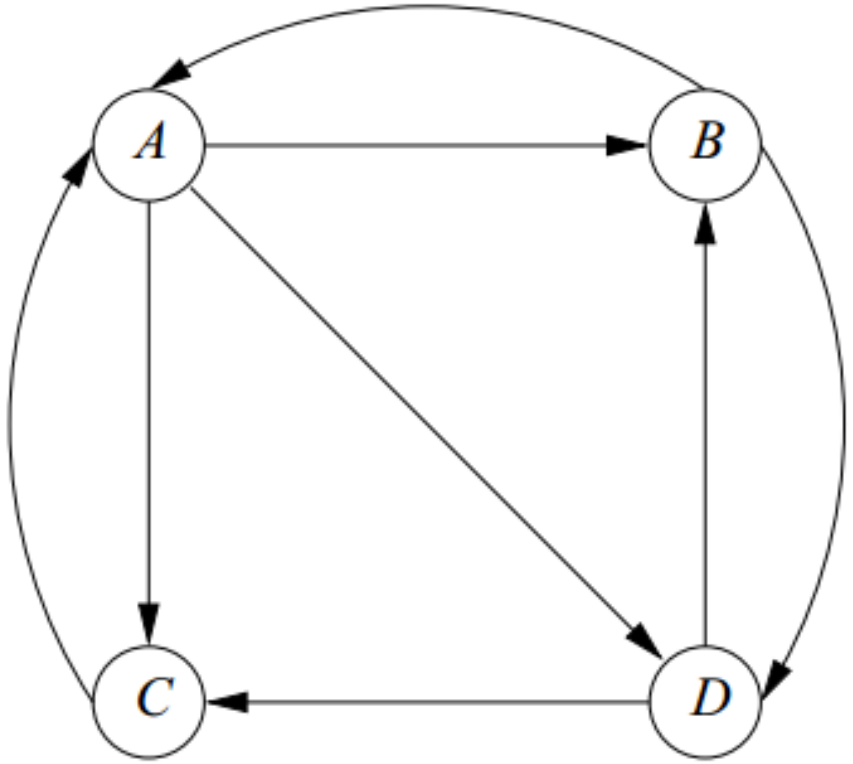
Magasabb érték, fontosabb weboldal.

Nincs egy konkrét megvalósítás

Idealizált modell

Web->irányított gráf
- Csúcsok->Honlapok
- Élek->Honlapok közötti linkek

Random surferök mászkálnak
- Véletlen pontban kezdenek
- Véletlen linkre mennek



Idealizált modell (folyt.)

Átviteli mátrix a gráfra (n pont)

- n sor és n oszlop
- m_{ij} 1/k, ha a j. oldalnak k db kimenő linkje van és vezet link az i. oldalra
- m_{ij} 0, ha a j. oldalnak nincs kimenő linkje vagy nem vezet az i. oldalra

M sztochasztikus

- Minden oszlopban 1 az elemek összege*

$$M = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

Idealizált modell (folyt.)

Random surfer helyének valószínűsége -> oszlopvektor

- j. elem értéke a relatív valószínűsége annak, hogy a surfer a j. honlapon van

Kezdési helynek azonos esély

- $v_0 = [\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}]$ oszlopvektor

M átviteli mátrix esetén az előfordulás esélye

- 1. lépés: v $= Mv_0$
- 2. lépés: v $= M^2 v_0$
- N. lépés: v $= M^N v_0$

Markov-folyamat*

Számítás az idealizált modellel

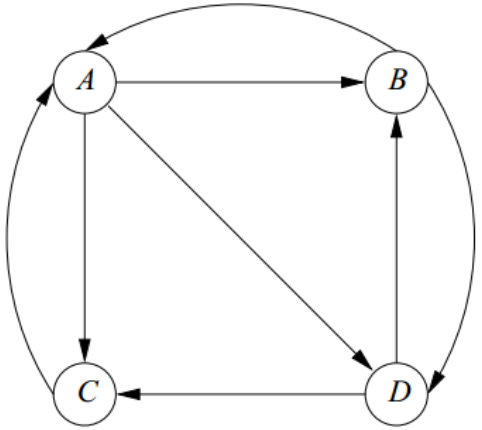
Megkötések a gráfra

- Erősen összefüggő gráf
- Nyelők nélkül

Addig hatványozzuk az M-et, amíg nem változik szignifikánsan az $M^* v_0$ értéke

- Gráf sajátvektora a v
- Gyakorlatban 50-75 iteráció elég

Példa számítás



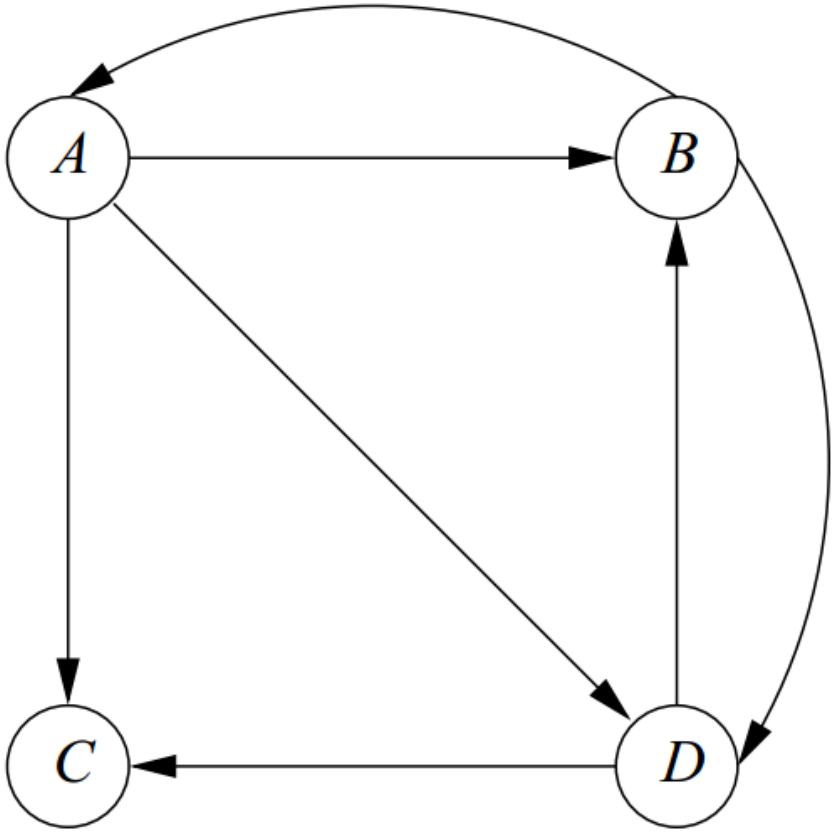
$$M = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 9/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix}, \begin{bmatrix} 15/48 \\ 11/48 \\ 11/48 \\ 11/48 \end{bmatrix}, \begin{bmatrix} 11/32 \\ 7/32 \\ 7/32 \\ 7/32 \end{bmatrix}, \ldots, \begin{bmatrix} 3/9 \\ 2/9 \\ 2/9 \\ 2/9 \end{bmatrix}$$

Problémák az idealizált modellel

A web nem felel meg a megkötéseknek

- Nem erősen összefüggő
- **Vannak nyelők**
- **Vannak ún. Spider trapek**



Nyelők kiküszöbölése

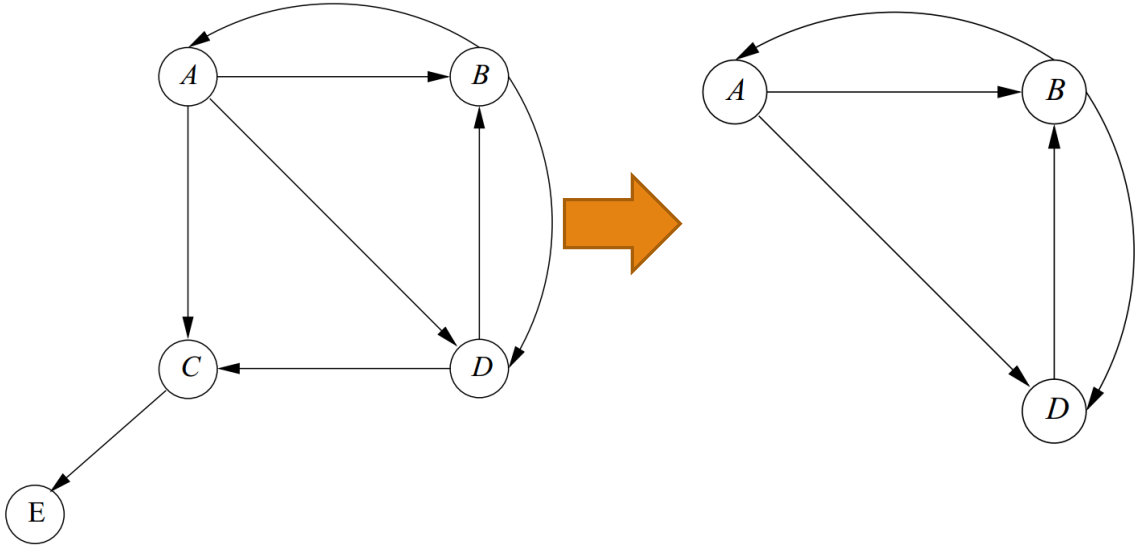
Mi is a gond velük?

- 0 kimenő él -> oszlop elemeinek összege nem 1, hanem 0
- M nem sztochasztikus -> Hatványozással „kiszívódnak” az értékek

Megoldások

- Nyelők elhagyása
  - Rekurzívan elhagyjuk a nyelőket
  - Kiszámoljuk a PageRank-et a módosított gráfra
  - Az elhagyásnak fordított sorrendjében kiszámoljuk az értékeket az elhagyott csúcsokra
- Random surfer-ek mozgásának módosítása

Példa a nyelők kiküszöbölésére



$$M = \begin{bmatrix} 0 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 1/2 & 1/2 & 0 \end{bmatrix}$$

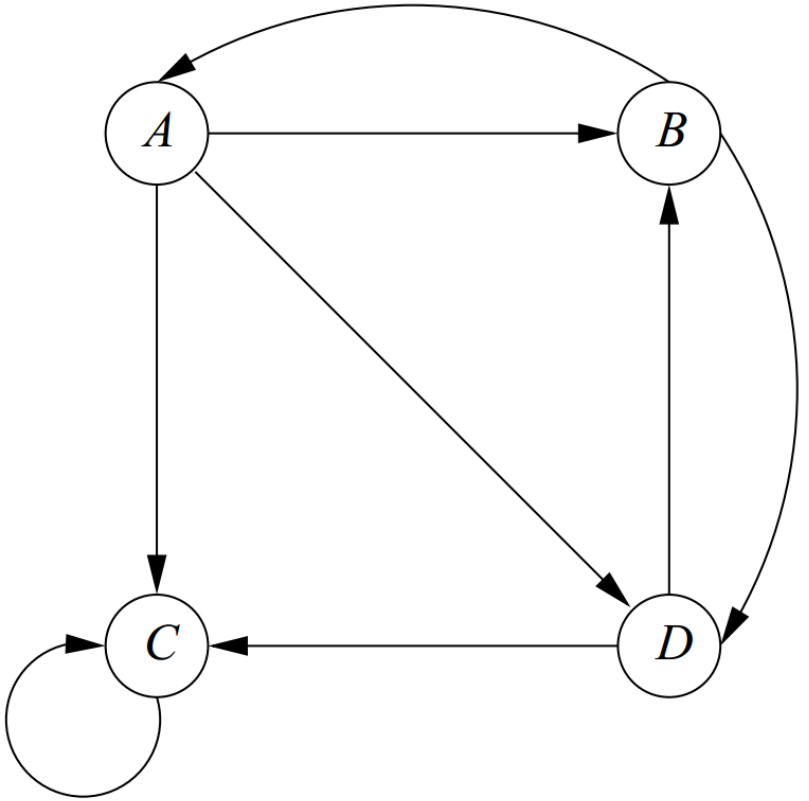
$$\begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}, \begin{bmatrix} 1/6 \\ 3/6 \\ 2/6 \end{bmatrix}, \begin{bmatrix} 3/12 \\ 5/12 \\ 4/12 \end{bmatrix}, \begin{bmatrix} 5/24 \\ 11/24 \\ 8/24 \end{bmatrix}, \ldots, \begin{bmatrix} 2/9 \\ 4/9 \\ 3/9 \end{bmatrix}$$

$$C = \frac{1}{3} * \frac{2}{9} + \frac{1}{2} * \frac{3}{9} = \frac{13}{54} = E$$

Spider Trapek

Spider Trap

- Csúcshalmazok, amelyek mindegyikéből vezet ki él, de csak a halmazon belülre
- Csapdába ejti a random surfer-eket
- Elrontja a Page Rank-et



Taxation

Alapötlet

- A link követése helyett néha teleportálnak a random surferek

Megoldás: $v' = \beta M v + \frac{(1-\beta)e}{n}$

- β: konstans (0.8 és 0.9 között)
- e: csupa egyes vektor

Nyelők esetén nem tökéletes

PageRank használata valós kereső eszközökben

A találati lista sorrendezésére használják

Csak egy a számos komponens közül

- Googlenél több, mint 250 ilyen van
- Pl. hol található az oldalon a kereső szó
- Hány keresett kifejezés van az oldalon
- De a PageRank az egyik legfontosabb (!)

Melynek legmagasabb a PageRank-je?

- **A:** http://en.wikipedia.org/
- **B:** http://cnn.com
- **C:** http://www.w3.org/
- **D:** http://www.nasa.gov/

Melynek legmagasabb a PageRank-je?

- **A:** http://en.wikipedia.org/ - 8
- **B:** http://cnn.com - 8
- **C:** http://www.w3.org/ - 9
- **D:** http://www.nasa.gov/ - 8

PageRank

SZÁMÍTÁSA A VALÓSÁGBAN

Alapprobléma

Valóságban óriási gráfok vannak

Ezekből átviteli mátrix

Majd azt ~50-szer hatványozni kell

Átviteli mátrix tárolása

Web gráfja
- Nagyon sok csúcs
- Viszont ritkák az élek (átlagban 10 oldalanként)

Szomszédossági mátrix nem hatékony
- 10 milliárd csúcs esetén minden milliárdadik mező nem 0

Listázzuk inkább a csúcsokat
- Hozzájuk tartozó kimenő élekkel

$$M = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

Source	Degree	Destinations
A	3	B, C, D
B	2	A, D
C	1	A
D	2	B, C

MapReduce

Egy programozási modell nagy adathalmazok párhuzamos feldolgozására.

Két részből áll

- Map(): Adatok szűrése és osztályozása
- Reduce(): Adatok összegzése

Gyakorlatban egy MapReduce rendszer az alábbi lépéseket végzi

1. A rendszer szétosztja a bemeneti K1 kulcs-érték párokat a szálak Map() függvényének
2. A szálak a saját kulcshoz tartozó értékeken elvégzik a műveletüket és legenerálják a K2 kulcs-érték párjukat – **Map()**
3. A K2 kulcs-értékpár okat a rendszer szétosztja a Reduce() függvényeknek – **Shuffle()**
4. A szálak elvégzik az összegző műveleteiket a K2 kulcs-érték párjukon – **Reduce()**
5. A rendszer összegyűjti a Reduce() függvények kimeneteit és összeállítja a végső eredményt

PageRank számítása MapReduce-szal

Emlékeztető: $v' = \beta M v + \frac{(1-\beta)e}{n}$

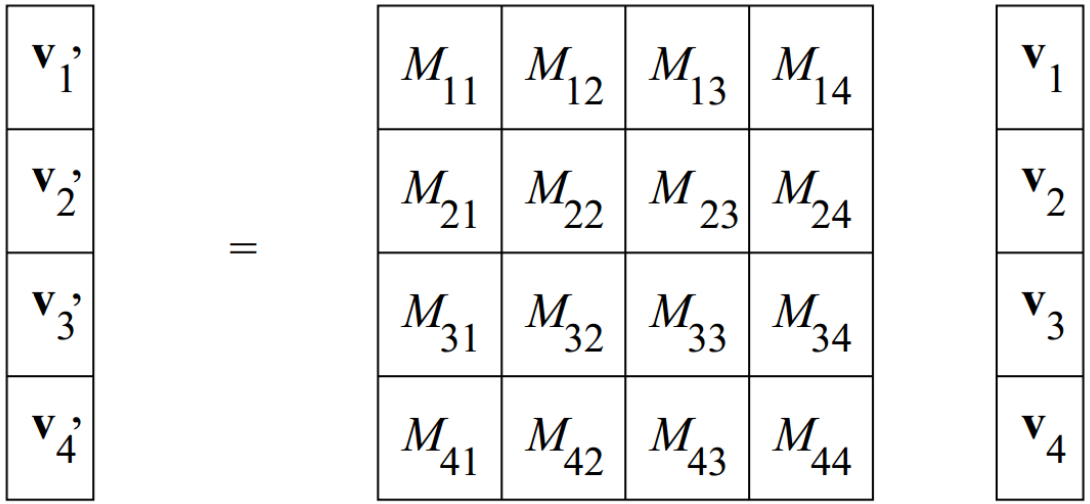
A konstans szorzások magától értetődőek, ezért csak a $v' = Mv$-re koncentrálunk.

Map:

- K^2 db szál
- Bemenet: M_{ij} és v_i
- Kimenet: v'_{ij}

Reduce:

- K db szál
- Bemenet: v'_{ij}-k
- Kimenet v'_i



Felbontás K=4 esetén

PageRank MapReduce példa

$$\begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

M bementi mátrix

Kiinduló vektor: $v = [\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]^T$

$$M_{11} = \begin{bmatrix} 0 & 1/2 \\ 1/3 & 0 \end{bmatrix} \quad M_{12} = \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix}$$

$$M_{21} = \begin{bmatrix} 1/3 & 0 \\ 1/3 & 1/2 \end{bmatrix} \quad M_{22} = \begin{bmatrix} 0 & 1/2 \\ 0 & 0 \end{bmatrix}$$

$$v_1 = \begin{bmatrix} 1/4 \\ 1/4 \end{bmatrix} \qquad v_2 = \begin{bmatrix} 1/4 \\ 1/4 \end{bmatrix}$$

Első Map() függvény:

$$v'_{11} = M_{11} * v_1 = \begin{bmatrix} 0 & 1/2 \\ 1/3 & 0 \end{bmatrix} * \begin{bmatrix} 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1/8 \\ 1/12 \end{bmatrix}$$

Második Map() függvény:

$$v'_{12} = M_{12} * v_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix} * \begin{bmatrix} 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1/4 \\ 1/8 \end{bmatrix}$$

Első Reduce() függvény:

$$v'_1 = v'_{11} + v'_{12} = \begin{bmatrix} 1/8 \\ 1/12 \end{bmatrix} + \begin{bmatrix} 1/4 \\ 1/8 \end{bmatrix} = \begin{bmatrix} 3/8 \\ 5/24 \end{bmatrix}$$

Melynek legmagasabb a PageRank-je?

- **A:** http://www.bme.hu/
- **B:** http://www.nemzetisport.hu/
- **C:** https://www.google.hu/
- **D:** http://index.hu/

Melynek legmagasabb a PageRank-je?

- **A:** http://www.bme.hu/ - 8
- **B:** http://www.nemzetisport.hu/ - 6
- **C:** https://www.google.hu/ - 6
- **D:** http://index.hu/ - 7

PageRank

TOPIC-SENSITIVE PAGERANK

Alapgondolat

PageRank egy továbbfejlesztése

- Témák bevezetése
- Oldalak súlyozása a tematikájuk alapján

Random surferek viselkedésének módosítása

- Inkább mennek olyan oldalakra, amelyek lefedik az adott témakört

Motiváció

Motiváció (folyt.)

Ideális eset: minden felhasználónak külön PageRank vektor

- Lehetetlen

Külön vektor minden (viszonylag kevés) témának

- A témába tartozó oldalak PageRankjai nagyobbak

Felhasználókat osztályozzuk érdeklődésük alapján

Számítása

Legyen egy új oszlopvektor, amely a témába tartozó honlapokhoz 1-et, a többihez 0-t rendel.

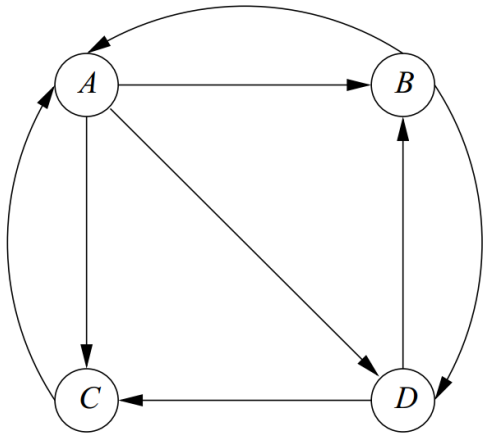
Ez legyen az e_s, ahol az S a témába tartozó honlapok száma

Új egyenlet: $v' = \beta M v + (1 - \beta) e_S / |S|$

- β: 1-nél kicsit kisebb konstans

Minden lépésnél véletlen teleportáció egy a témába tartozó honlapra

Példa a Topic-Sensitive PageRankra



$$M = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

$\beta = 0.8, S = \{B, D\}$

$$\beta M = \begin{bmatrix} 0 & 2/5 & 4/5 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{bmatrix} \quad e_S = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{v}' = \begin{bmatrix} 0 & 2/5 & 4/5 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{bmatrix} \mathbf{v} + \begin{bmatrix} 0 \\ 1/10 \\ 0 \\ 1/10 \end{bmatrix}$$

Eredmény: $\begin{bmatrix} 0/2 \\ 1/2 \\ 0/2 \\ 1/2 \end{bmatrix}, \begin{bmatrix} 2/10 \\ 3/10 \\ 2/10 \\ 3/10 \end{bmatrix}, \begin{bmatrix} 42/150 \\ 41/150 \\ 26/150 \\ 41/150 \end{bmatrix}, \begin{bmatrix} 62/250 \\ 71/250 \\ 46/250 \\ 71/250 \end{bmatrix}, \ldots, \begin{bmatrix} 54/210 \\ 59/210 \\ 38/210 \\ 59/210 \end{bmatrix}$

Használata a keresőmotorokban

1. El kell döntenünk, hogy milyen témákra készítsünk vektort
2. Minden témára el kell készíteni a teleport vektort
3. **Meg kell találni a legrelevánsabb témát az adott keresésekhez**
   - Kiválaszthatja maga a felhasználó a témát
   - Korábbi keresések alapján tippelünk
   - A felhasználó internetes lábnyoma alapján választunk
4. Használni kell vektort a keresés módosítására

Téma kitalálása a szavakból

Egy dokumentum témájának kitalálása a szövege alapján nehéz feladat

Egyszerűsítünk

- Az oldalon előforduló, máshol ritka szavakat vizsgáljuk
- Túl ritka szavak kihagyása
- Minimum előfordulási szám meghatározása

$S_1, S_2, \dots, S_k$: A témákhoz tartozó szavak halmazai

P: Az oldalon előforduló szavak halmaza

Jaccard-hasonlóság a P és az S_i halmazok között

Page Rank

TRUSTRANK

Link Spam

A PageRank minimalizálta a Term Spam hatékonyságát

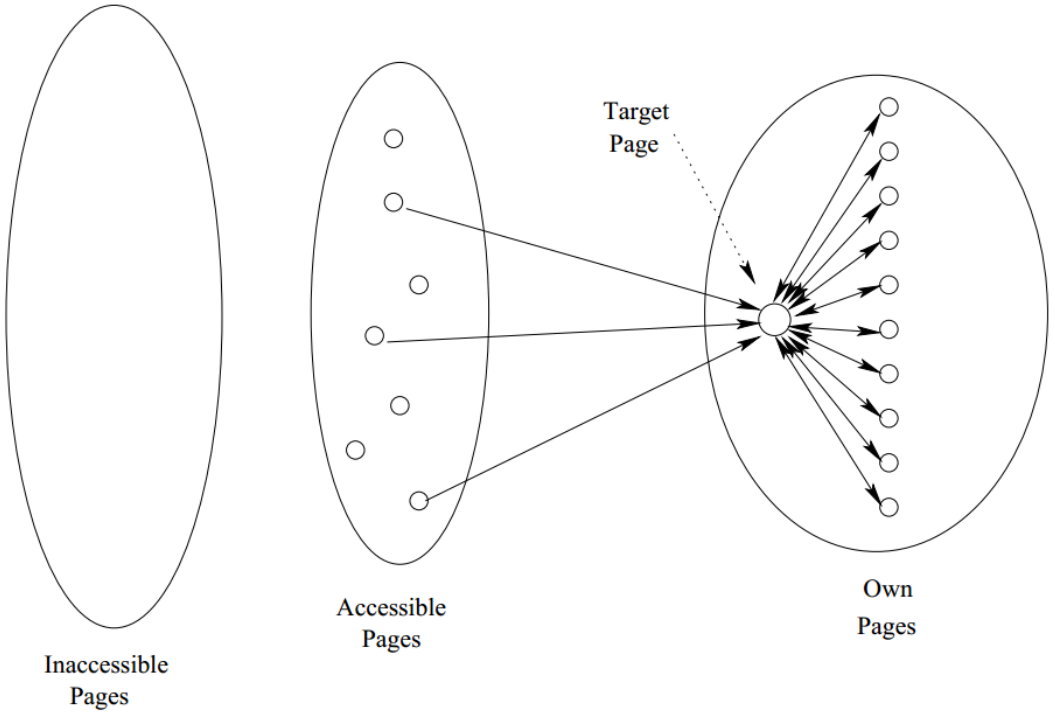
Emiatt új módszer: Link Spam

Azok az oldalakat, amelyek egy adott oldal PageRankjának mesterséges növelésére hoztak létre nevezzük **spam farm**nak.

Spam Farm architektúrája

Három részre osztjuk a webet a spammer szempontjából

- Elérhetetlen oldalak
- Elérhető oldalak
- Saját oldalak



Harc a Link Spam ellen

Struktúrák detektálása

- Mindig alkothatók újak

Link Spam oldalak PageRankjának automatikus módosítása

- TrustRank
- Spam mass

TrustRank

A Topic-Sensitive PageRank egy variációja

Honlapok két csoportba sorolása

- Megbízható
- Nem megbízható

Megbízható honlapok ritkán linkelnek nem megbízhatóakra

Mik a megbízható honlapok?

- A legmagasabb PageRankkel rendelekzők
- Pl. .gov, .edu TLD-k.

Spam Mass

Ötlet: A spam gyanús honlapokat vegyük ki a PageRankből

Számoljuk ki a hagyományos PageRanket illetve a TrustRanket

Spam mass = $\frac{(r-t)}{t}$

- r: PageRank
- t: TrustRank

Negatív vagy kicsit pozitív-> Nem spam

1-hez közelítő -> Spam

Egyéb algoritmusok

HITS/Hubs and authorities bevezetés

HITS – Hyperlink-induced topic search

A PageRank után lett feltalálva

Viszonya a PageRankhöz

- Nem a PageRank helyett, inkább mellette használatos
- PageRank: 1 dimenziós értékek
- HITS: 2 dimenziós értékek

Gyakorlatban az Ask.com kereső használja biztosan

HITS alapok

Weblapok két csoportba

- Authorities: Információkat szolgáltatnak az adott témáról
- Hubs: Nem a konkrét információt tartalmazzák, hanem azt, hogy hol kell megtalálni ezeket

Alapgondolat

- Egy hub „jó”, ha „jó” authority-ket linkel.
- Egy authority „jó”, ha „jó” hubok linkelik.

HITS formalizálása

Minden honlaphoz két pontszám

- Hogy mennyire jó hub
- Hogy mennyire jó authority

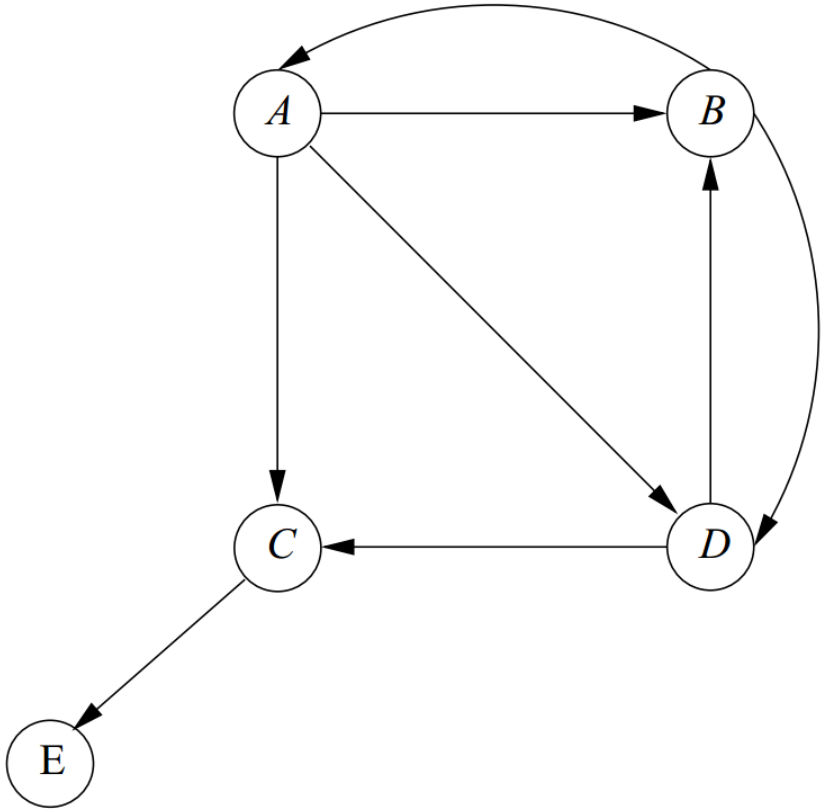
Két vektor

- $h = [h_1, h_2, \dots, h_n]$, az i. érték az i. honlap „hub-fokával”.
- $a = [a_1, a_2, \dots, a_n]$, az i. érték az i. honlap „authority-fokával”.

L mátrix egy olyan mátrix, amelyben $L_{ij} = 1$, ha az i. honlap linkeli a j.-et, különben 0.

L^T mátrix ennek transzponáltja

Példa a HITS-re



$$L = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$L^{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

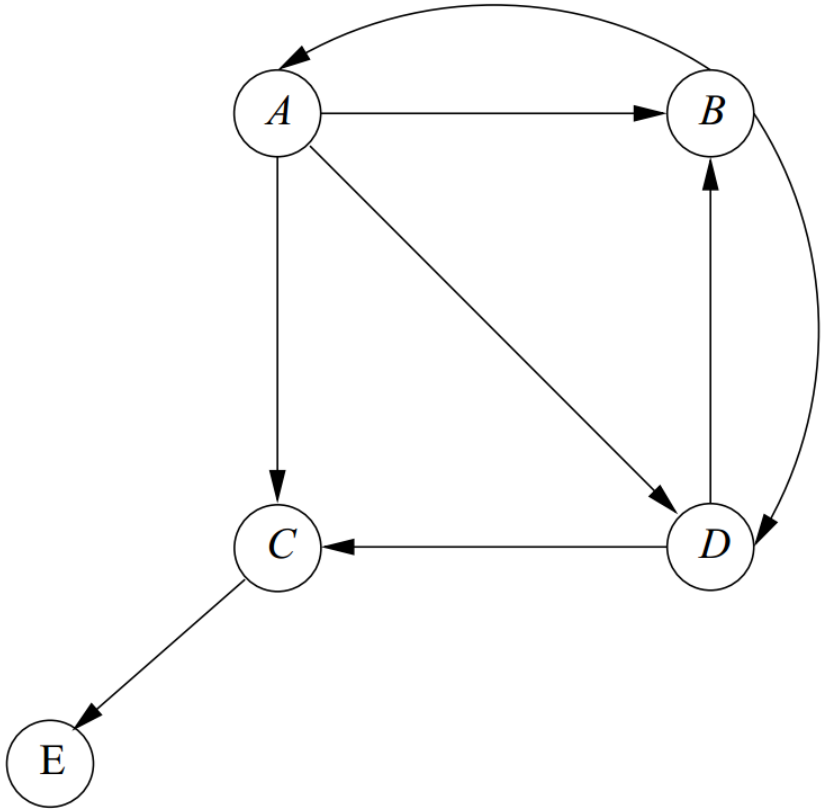
HITS számítása

Kiindulunk egy csupa 1 h vektorból

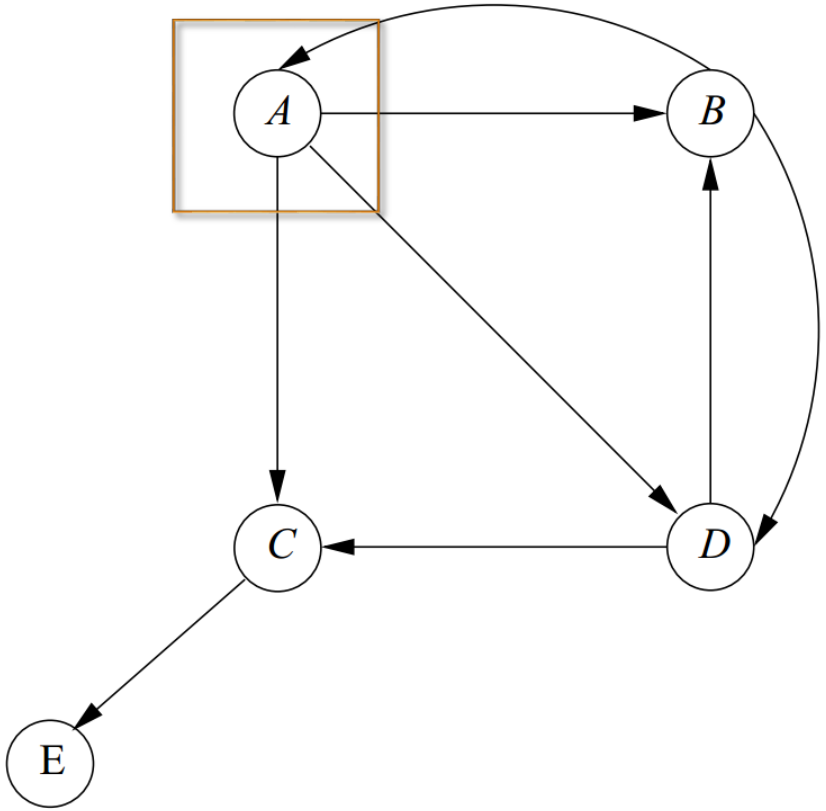
Majd többször végrehajtjuk az alábbi műveleteket

- $\boldsymbol{a} = L^T \boldsymbol{h}$
- Skálázzuk az $\boldsymbol{a}$ vektort úgy, hogy a legnagyobb komponens 1 legyen
- $\boldsymbol{h} = L\boldsymbol{a}$

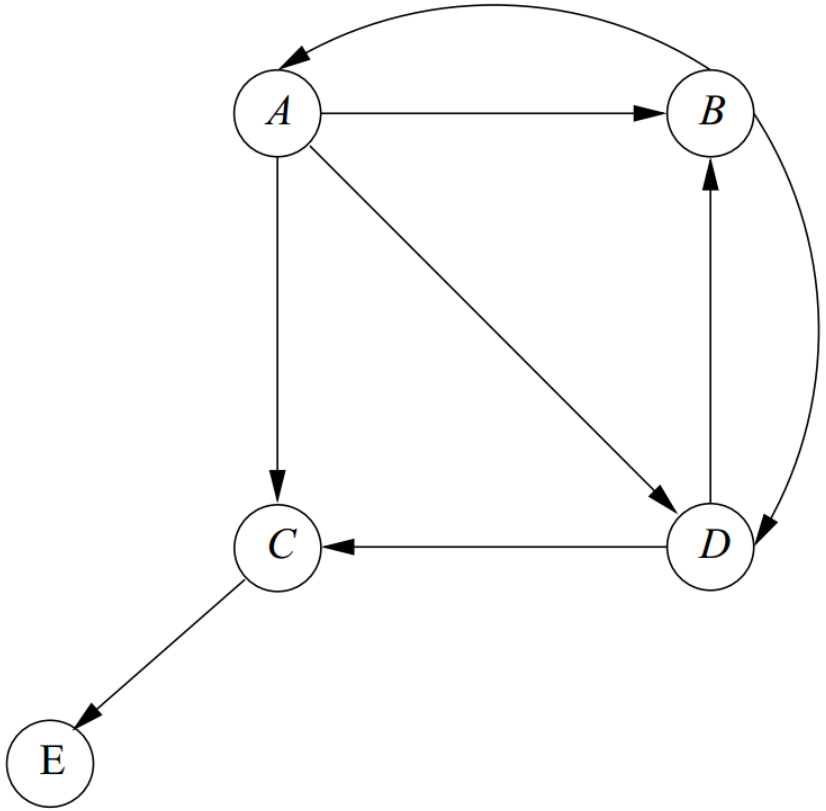
Melyik a legjobb hub?



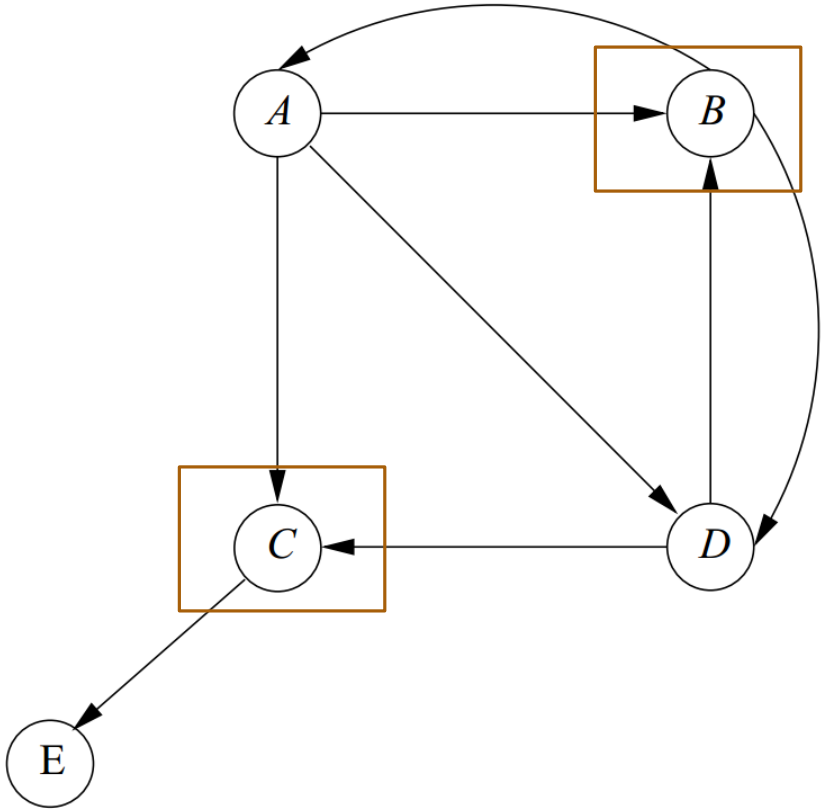
Melyik a legjobb hub?



Melyik kettő a legjobb authority?



Melyik kettő a legjobb authority?



HITS számítás példa

$$L = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$L^{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1/2 \\ 1 \\ 1 \\ 1 \\ 1/2 \end{bmatrix} \quad \begin{bmatrix} 3 \\ 3/2 \\ 1/2 \\ 2 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1/2 \\ 1/6 \\ 2/3 \\ 0 \end{bmatrix}$$

$$\mathbf{h} \qquad L^{\mathrm{T}}\mathbf{h} \qquad \mathbf{a} \qquad L\mathbf{a} \qquad \mathbf{h}$$

$$\begin{bmatrix} 1/2 \\ 5/3 \\ 5/3 \\ 3/2 \\ 1/6 \end{bmatrix} \quad \begin{bmatrix} 3/10 \\ 1 \\ 1 \\ 9/10 \\ 1/10 \end{bmatrix} \quad \begin{bmatrix} 29/10 \\ 6/5 \\ 1/10 \\ 2 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 12/29 \\ 1/29 \\ 20/29 \\ 0 \end{bmatrix}$$

$$L^{\mathrm{T}}\mathbf{h} \qquad \mathbf{a} \qquad L\mathbf{a} \qquad \mathbf{h}$$

Google Panda

Google keresőjének egy eleme

- 2011 elején vezették be
- Az „alacsony minőségű” honlapok kiszórása volt a cél

Alacsony minőség

- Sok reklám az oldalon
- Tárgyi tévedések cikkekben
- Helyesírási hibák
- Tartalmaz-e új információt más oldalakhoz képest

Újdonság

- Implied linkek figyelése
  - Pl. ha egy fórumban nem linkelik az oldalt, de megemlítik a nevét

Google Hummingbird

Google kereső legújabb publikus algoritmusa

- 2013 végén került bevezetésre

Kontextus és szinonimák

- Nem a keresési listák rangsorolására elsődlegesen
- Keresési kifejezések megértésére koncentráltak
  - Nem szavanként, hanem egyben értelmezik a keresett kifejezést

Szemantikus keresés korát hozhatja a remények szerint

Köszönöm a figyelmet!