# Informationsåtervinning på webben
# Sökmotorernas framtid


KUNGL. INGENJÖRSVETENSKAPSAKADEMIEN
Royal Swedish Academy of Engineering Sciences

Seminarium 2004-09-02



# Informationsåtervinning på webben
# Sökmotorernas framtid

- Ge inspiration till
  - forskning
  - att skapa nya affärsmöjligheter
  - smart användning av sökverktyg i den egna organisationen
- Belysa sökmotorer ur ett tekniskt, juridiskt och affärsmässigt perspektiv.


KUNGL. INGENJÖRSVETENSKAPSAKADEMIEN
Royal Swedish Academy of Engineering Sciences

# Informationsåtervinning på webben
# Sökmotorernas framtid - **Program**

- Sökmotorer och tekniken bakom, *Bo Kågström*
- Sökmotorer i framtiden, *Jussi Karlgren*
- Integritet, säkerhet och manipulation, *Nicklas Lundblad*

  Affärsperspektiv och applikationer
- Web4Health, *Jacob Palme*
- SiteSeeker och Euroling, *Hercules Dalianis*
- Askology och QuickAsk, *Erik Sneiders*
- Öppen diskussion, *Sture Hägglund* moderator



# Sökmotorer och tekniken bakom

## Ranking av webbsidor med länkanalys - Googles PageRank och liknande metoder

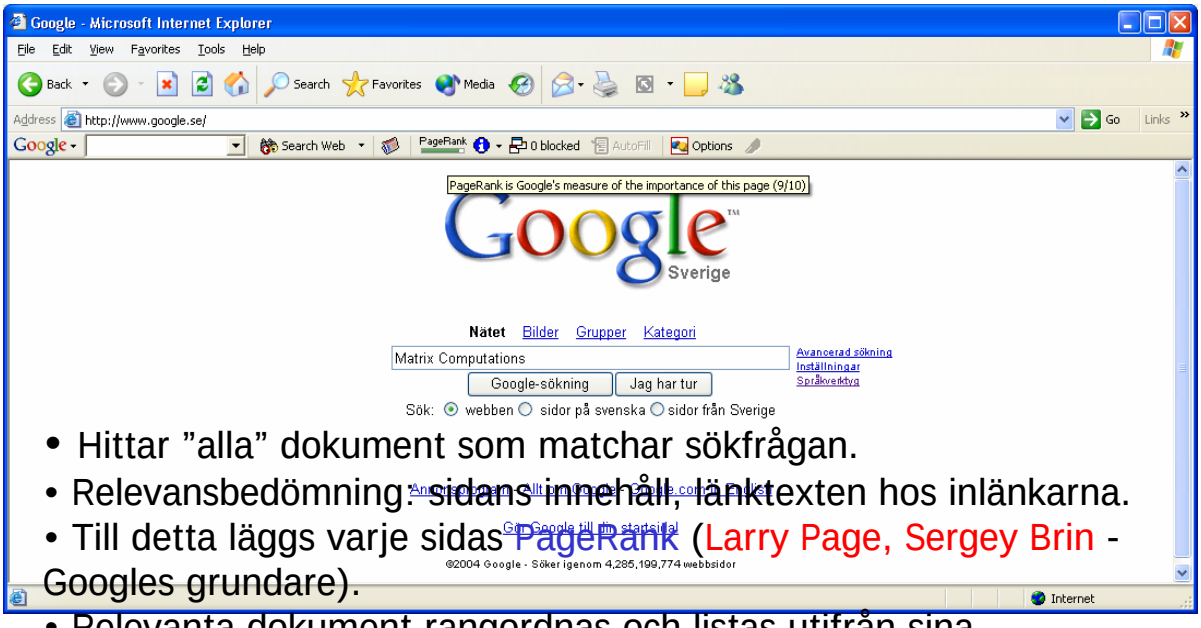
Bo Kågström
Dept. of Computing Science and HPC2N
Umeå University
bokg@cs.umu.se

# Lite bakgrund

- Webb-sökning:
  - Angelägen om ”breda förfrågningar” (*broad-topic queries)*, t.ex. ”web-browsers”.
  - *Överflödsproblematik*: # sidor som hittas och bedöms som relevanta (”träffar”) är alldeles för stort!
  - Behövs en mekanism för att rangordna dessa sidor.
- Hypotes: Om sidan j har en länk till sidan i, så ger den auktoritet till i.
  - Hur används länkinformationen för att rangordna ”träffar”?
  - Önskar relevanta och auktoritativa sidor.

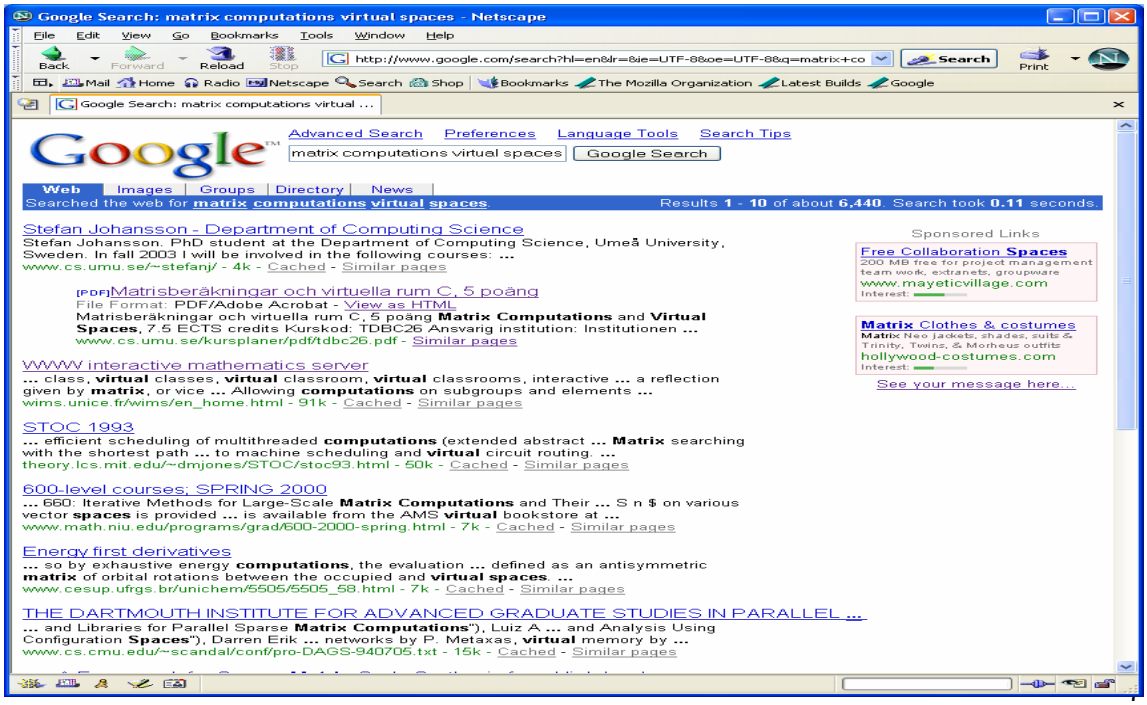


# Hur går en Googlesökning till?




- Hittar ”alla” dokument som matchar sökfrågan.
- Relevansbedömning: sidans innehåll, länktexten hos inlänkarna.
- Till detta läggs varje sidas PageRank (Larry Page, Sergey Brin - Googles grundare).
- Relevanta dokument rangordnas och listas utifrån sina PageRank-värden.

# Exempel på sökresultat




# Innehåll

- Sökning på webben – lite bakgrund
- Googles PageRank-algoritm
  - Definition av PageRank
  - PageRank är en ”dominerande” egenvektor
  - Rank-sinks och -sources – modifierad definition
  - Beräkning av PageRank – världens största matrisberäkning?
  - Varför fungerar det? Konvergensegenskaper
- HITS-algoritmen: Hypertext Internet Topic Search
  - Auktoriteter och hubbar är ”dominerande” singulära vektorer

# Webb-grafen & Webb-matrisen

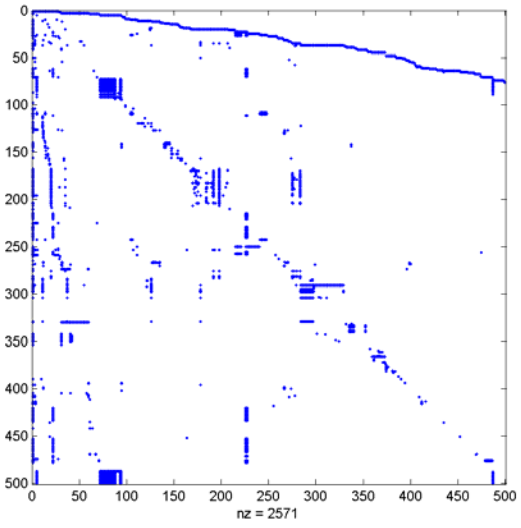
- WWW kan representeras som en graf med ”sites” (t.ex. hemsidor) som noder and länkar som kanter.
- Webbmatrisen A (adjaceny or connectivity) representerar länkstrukturen mellan sidor:

  A(i,j) = 1 om sida i pekar på sida j
  A(i,j) = 0 annars
- A är en gles (sparse ) av storlek n x n, n > 4 miljarder (4 $10^9$)!



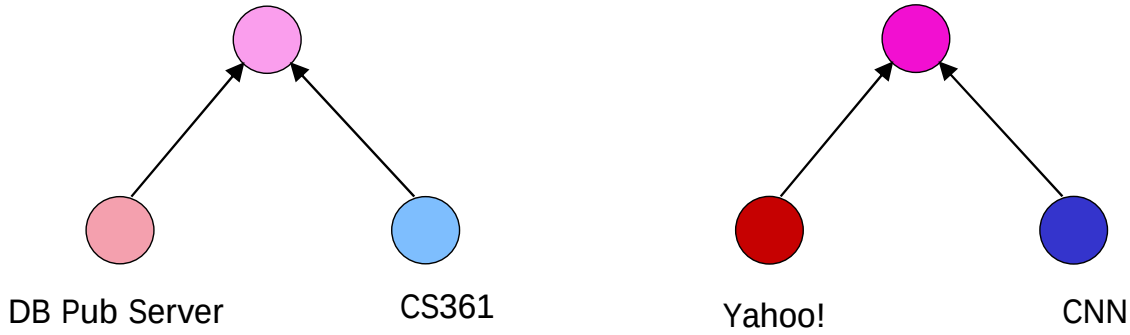
# Webb-matris: Harvard 500x500

G(i,j) = 1 om url{i} länkar till url{j}.




Skapad med [U,G] = surfer('http://www.harvard.edu',500)
U cell-array med besökta URL:er

# Inlänkar räknas!




Länkad av två "mindre viktiga" sidor

Länkad av två "viktiga" sidor



# Definition av PageRank

- En sidas "betydelse" (vikt) ges av "vikten" hos de sidor som pekar på den.

$$x_i = \sum_{j \in B_i} \frac{1}{N_j} x_j$$

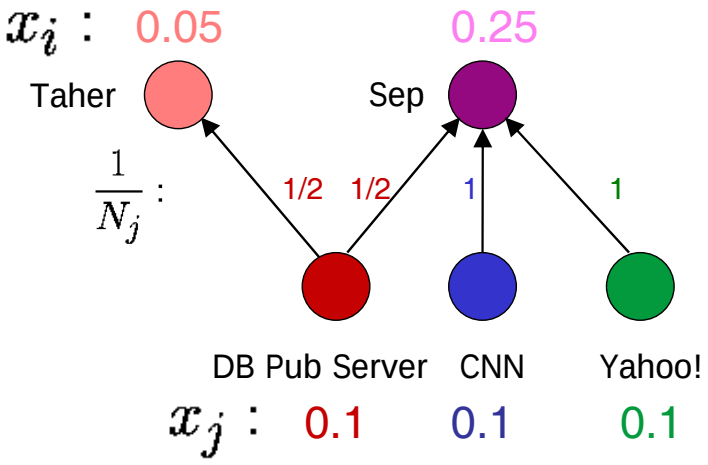
importance of page *i*

pages *j* that link to page *i*

number of out-links from page *j*

importance of page *j*

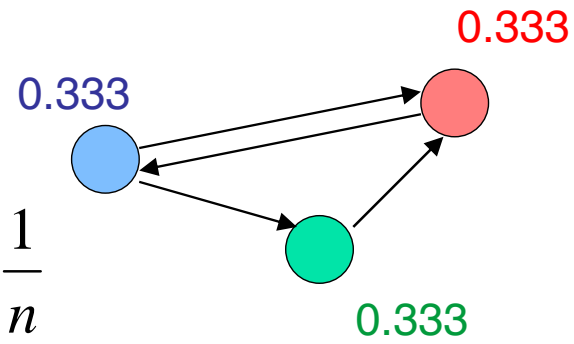
# Definition av PageRank - exempel




$$x_i = \sum_{j \in B_i} \frac{1}{N_j} x_j \qquad 0.1 \cdot \frac{1}{2} + 0.1 \cdot 1 + 0.1 \cdot 1 = 0.25$$



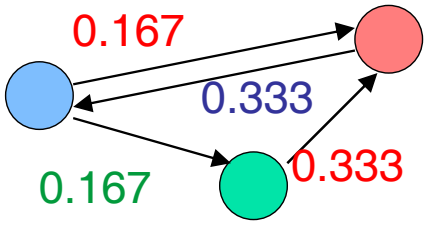
# PageRank-diagram




$$x_i^{(0)} = \frac{1}{n}$$

Initialisera alla noder till samma rang (vikt)

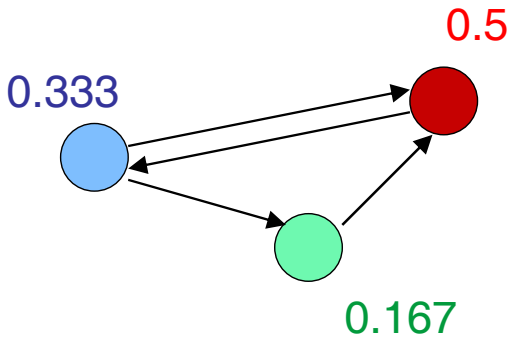
# PageRank-diagram




Propagera rangerna över länkarna
(multiplicera med länkvikter)

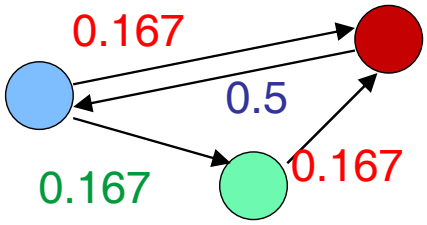


# PageRank-diagram



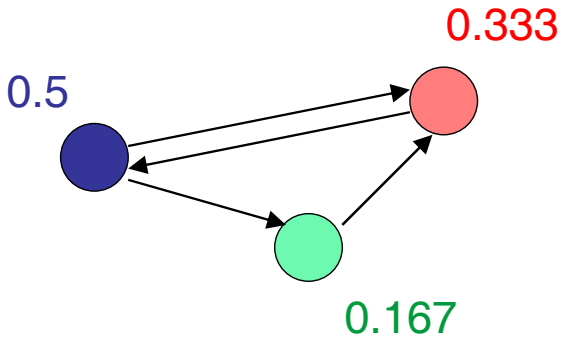

$$x_i^{(1)} = \sum_{j \in B_i} \frac{1}{N_j} x_j^{(0)}$$

# PageRank-diagram



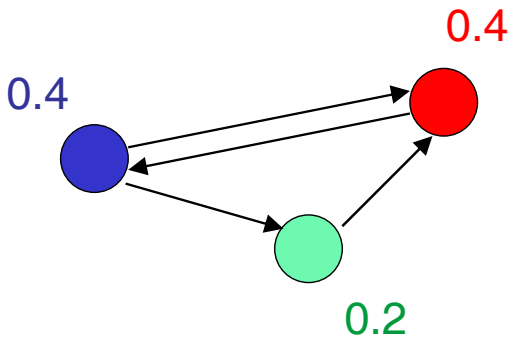



# PageRank-diagram




$$x_i^{(2)} = \sum_{j \in B_i} \frac{1}{N_j} x_j^{(1)}$$

# PageRank-diagram




Efter ett tag …

$$x_i = \sum_{j \in B_i} \frac{1}{N_j} x_j$$



# Förenklad beräkning av PageRank

- Initialisering: $x_i^{(0)} = \frac{1}{n}$
- Upprepa tills konvergens:

$$x_i^{(k+1)} = \sum_{j \in B_i} \frac{1}{N_j} x_j^{(k)}$$

importance of page *i*

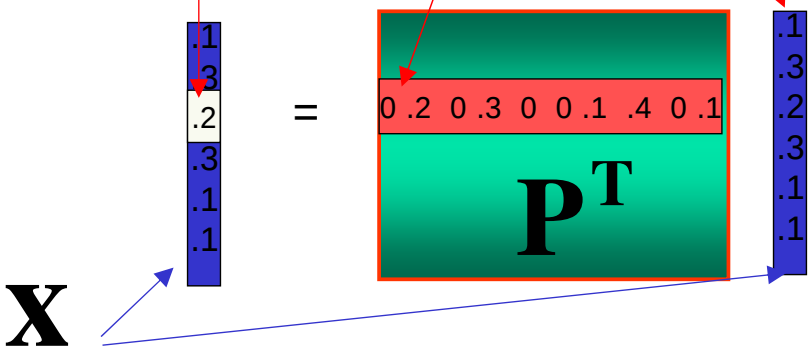
pages *j* that link to page *i*

number of outlinks from page *j*

importance of page *j*

# Matrisnotation – rätt abstraktion

$$x_i = \sum_{j \in B_i} \frac{1}{N_j} x_j$$

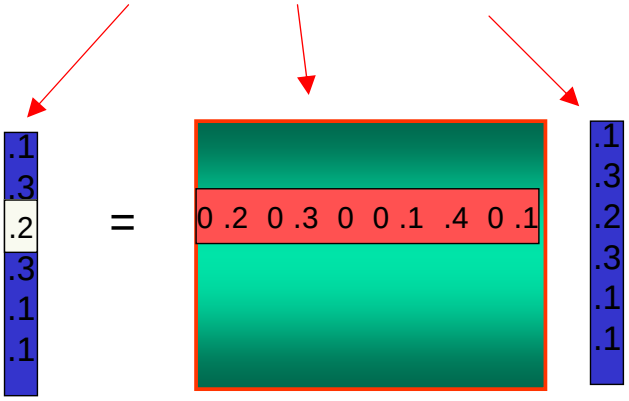





# Sökt: egenvektor svarande mot största egenvärdet = 1

Hitta **x** som uppfyller:

$$\mathbf{x} = \mathbf{P}^T\mathbf{x}$$



# Tillämpa potensmetoden (Power Method)

- Initialisering:

$$\mathbf{x}^{(0)} = \begin{bmatrix} \frac{1}{n} & \cdots & \frac{1}{n} \end{bmatrix}^{\mathbf{T}}$$

- Upprepa tills konvergens:

$$\mathbf{x}^{(k+1)} = \mathbf{P}^{\mathbf{T}}\mathbf{x}^{(k)}$$

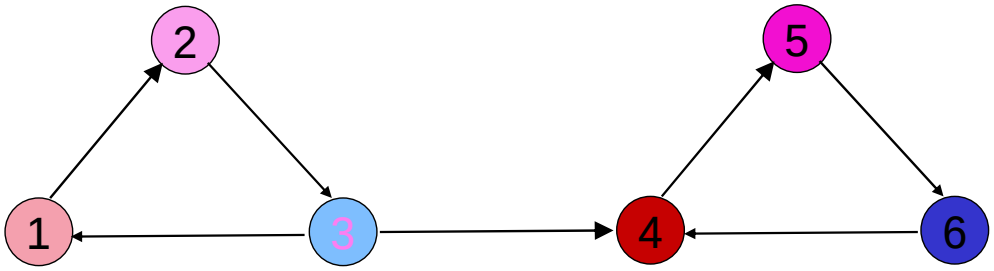
Stoppkriterium: norm( $x^{(k+1)} - x^{(k)}$ ) < tolerance



# Random Walk på webben

- Definitionen av PageRank kan ses som slumpvandringar (random walks) på grafer.
- Surfa från sida till sida genom att slumpmässigt välja en utlänk från en sida för att komma till nästa.
- Kan leda till ”dead ends” hos sidor som saknar utlänkar (dangling pages), eller cykler kring klickar av sammanhängande sidor (loops).

# Loop som en rang-ansamlare




Rank Sink: Loopen 4 -> 5 -> 6 ackumulerar rang men kommer aldrig att distribuera någon rang (inga utgående länkar).



# Rank Sink - problematik

- Alla egenvektorer till webbmatrisen $P^T$ i sista exemplet har nollor i de tre första komponenterna.

$$P = \left[\begin{array}{ccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right]$$

PageRank för sidorna 1, 2 and 3 are 0!

Botemedel: introducera artificiella länkar (rank sources).

# PageRank med Rank Sources

$$\text{PageRank(i)} \equiv x_i = \sum_{j \in B_i} \frac{1}{N_j} x_j + s_i \qquad s = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}$$

Alla PageRank är skilda från noll från början!

- En ”random surfer” kan följa vilken utlänk som helst från en sida med samma sannolikhet (förenklade definitionen).

- Då och då, blir hon ”less” and hoppar till en slumpvis sida på Webben (ny definition med ”rank sources”).



# Googles PageRank-matris

- Periodiskt, väljs en slumpvis sida på webben för att överkomma ”dangling pages” och loopar.
- $A = c\ P^T + (1 - c)\ E^T$

  c = bråkdel av tiden som en “surfare” (random walk) följer en länk (t.ex., c = 0.85)

  1 - c = bråkdel av tiden som en godtycklig sida väljs

  E är nxn med E(i,j) = 1/n (n = # länkar i Webbmatrisen)
- A är “tät” (dense), rang-1-modifiering av en gles matris - de flesta A( i, j ) = (1 - c) / n.

# Perron-Frobenius teorem

- $A = c\,P^T + (1 - c)\,E^T$ är övergångsmatris hos en Markovkedja (transition probability matrix)

$0 <= A(i,j) <= 1$, alla kolumnsummor = 1

- PF: A:s största egenvärde = 1 svarande till en entydig egenvektor x med $x_i >= 0$

$$Ax = x$$

$$\sum_{i=1}^{n} x_i = 1 \quad \Longrightarrow$$

x är Markovkedjans tillståndsvektor (state vector of the Markov chain)



# Potensmetoden tillämpad på A – världens största matrisberäkning?

- Initialisering:

$$\mathbf{x}^{(0)} = \begin{bmatrix} \frac{1}{n} & \cdots & \frac{1}{n} \end{bmatrix}^{\mathbf{T}}$$

- Upprepa tills konvergens:

$$\mathbf{x}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)}$$

Beräkning av $y = Ax$ där $A = c\,P^T + (1 - c)/n\; e\,e^T$
- A beräknas ej explicit
- Utnyttja A:s struktur

Elementen i x är Googles PageRank!

# Varför fungerar det?

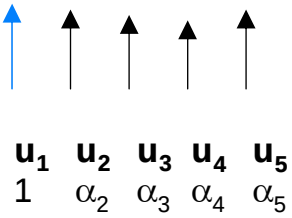
- Antag att *n x n* matrisen **A** har *n* egenvektorer $\mathbf{u_i}$.

$$\mathbf{A}\mathbf{u_i} = \lambda_i \mathbf{u_i}$$

- Då kan en godtycklig *n*-dimensionell vektor skrivas som en linjär-kombination av egen-vektorerna till **A**.

$$\mathbf{x}^{(0)} = \mathbf{u}_1 + \alpha_2\mathbf{u}_2 + \ldots + \alpha_n\mathbf{u_n}$$

$$\lambda_1 = 1; \quad \lambda_1 > |\ \lambda_2\ | \geq \ldots$$

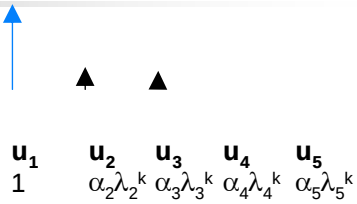





# Konvergensegenskaper

$$\mathbf{x}^{(k)} = \mathbf{u}_1 + \alpha_2\lambda_2{}^k\mathbf{u}_2 + \ldots + \alpha_n\lambda_n{}^k\mathbf{u_n}$$

$$\lambda_1 = 1; \quad \lambda_1 > |\ \lambda_2\ | \geq \ldots$$




- Konvergenshastigheten bestäms av 1/ $|\lambda_2|$ dvs beloppet av kvoten mellan det största och näst största egenvärdet.
- Ju mindre $\lambda_2$, desto snabbare konvergerar potensmetoden (abs($\lambda_2$) <= c, 0 <= c <=1)

# Är potensmetoden (PM) bästa valet?

- Traditionellt:
  - **A**, n << 4 miljarder, ofta tät matris.
  - Risk för att $\lambda_2$ är nära $\lambda_1$ ➔ potensmetoden långsam!.
- För detta problem:
  - **A**, enormt stor, kolumnstokastisk, ofta tät, rank-1 modifiering av en gles matris, där $\lambda_2$ är liten[1]

  ➔ Potensmetoden fungerar mycket bra!!

1 – Se Haveliwala T.H. and Kamvar S.D., "The Second Eigenvalue of the Google Matrix" dbpubs.stanford.edu/pub/2003-20.

- Det pågår forskning med att "snabba upp" PM.
- Andra metoder kan fungera lika bra eller bättre för beräkning av PageRank för mer begränsade domäner.



# PageRank - sammanfattning

- Sökning av webbsidor är huvudtillämpningen – används i fulltext-sökmotorn Google.
- PageRank är en global ranking av alla webbsidor, oberoende av dess innehåll, enbart baserad på dess plats i Webbgraf-strukturen (länk-baserad) - beräknas om ca 1 gång/månad, tar 1-2 veckor!?!
- Ranking används för att rangordna sidorna så att mer centrala webbsidor ges preferens.
- Bakåtlänkar från ”viktiga” sidor är mer signifikanta än bakåtlänkar från ”oviktiga” sidor (rekursiv definition av PageRank).