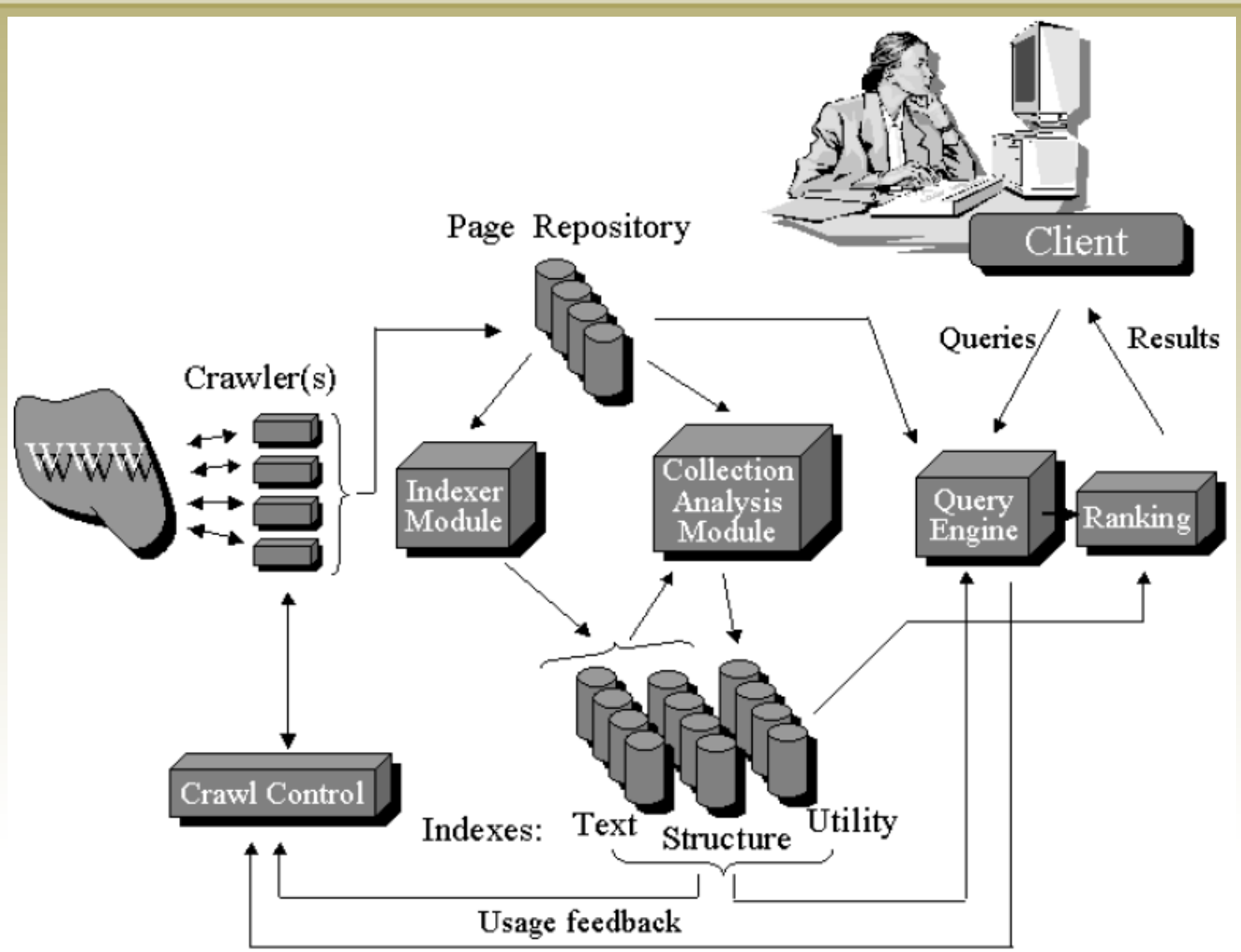
Ανάκληση Πληροφορίας

Διδάσκων –
Δημήτριος Κατσαρός

Διάλεξη 16η: 14/05/2014

Ζητήματα Μεγάλης-Κλίμακας Υλοποίησης του PageRank

Αρχιτεκτονική Μηχανής Αναζήτησης



Ευρετήρια (Indexes)

- Ευρετήρια Περιεχομένου (αντιστραμμένο ευρετήριο)
- Βοηθητικά Ευρετήρια
  - Θέσης (γεωγραφικής)
  - Εικόνων
- Ευρετήρια Συνδεσμολογίας (Link indexes)
- Ποιες "ποσότητες" μας ενδιαφέρουν?

Οντότητα	**Περιγραφή**	**Αποθήκευση**
$\mathbf{H}$	Αραιός πίνακας υπερσυνδέσμων	nnz($\mathbf{H}$)
$\mathbf{a}$	Αραιό δυαδικό διάνυσμα dangling κόμβων	$\|D\|$ ακέραιοι
$\mathbf{v}^T$	Πυκνό διάνυσμα προσωποποίησης, εκτός εάν $\mathbf{v}^T=\mathbf{e}^T/n$	n doubles
$\boldsymbol{\pi}^{(k)T}$	Πυκνό διάνυσμα PageRank τρέχουσας επανάληψης της power μεθόδου	n doubles

Αποθήκευση **H** (1/2)

- Εάν χωράει στην κύρια μνήμη, τότε OK
- Εάν όχι, τότε
  - Συμπίεση, ώστε να χωρέσει στην κύρια μνήμη και επινόηση μιας νέας μορφής του PageRank για να εκτελείται πάνω στη συμπιεσμένη μορφή
  - Αποτελεσματική υλοποίηση του I/O στα μη συμπιεσμένα δεδομένα
- Ακόμα και για μικρά γραφήματα, για τα οποία ο **H** χωράει στην κύρια μνήμη, απαιτούνται έξυπνες τεχνικές για να ελαττωθεί ο φόρτος εργασίας
- Για τον random surfer, ο H μπορεί να γραφεί ως

$$\mathbf{H} = \mathbf{D}^{-1}\,\mathbf{L}$$

  - **D**: κρατά τους outdegrees των κόμβων
    - $[\mathbf{D}^{-1}]_{ii} = 1/d_{ii}$, εάν δεν είναι dangling κόμβος
    - $[\mathbf{D}^{-1}]_{ii} = 0$, εάν είναι dangling κόμβος
  - **L**: πίνακας γειτνίασης (adjacency matrix)

Αποθήκευση **H** (2/2)

- Με τον τρόπο αυτό εξοικονομείται χώρος: αντί για nnz(**H**)
- Με τον τρόπο αυτό επιταχύνεται η εκτέλεση της power μεθόδου: $\boldsymbol{\pi}^{(k+1)T} = \alpha\boldsymbol{\pi}^{(k)T}\mathbf{H} + (\alpha\boldsymbol{\pi}^{(k)T}a + 1-\alpha)\mathbf{v}^T$
  - Το $\boldsymbol{\pi}^{(k)T}\mathbf{H}$ απαιτεί nnz(**H**) πολλαπλασιασμούς και nnz(**H**) προσθέσεις
  - Με χρήση του diag(D^{-1}), το $\boldsymbol{\pi}^{(k)T}\mathbf{H}$ μπορεί να επιτευχθεί ως

$$\boldsymbol{\pi}^{(k)T}\mathbf{D}^{-1}\mathbf{L} = \boldsymbol{\pi}^{(k)T} \ .* (\text{diag}(\mathbf{D}^{-1}))\mathbf{L}$$

    - Το πρώτο κομμάτι απαιτεί n πολλαπλασιασμούς, και ο πολλαπλασιασμός με το **L** απαιτεί nnz(**H**) προσθέσεις. Άρα, κερδίσαμε nnz(**H**)-n πολλαπλασιασμούς
- Δυστυχώς, αυτή η αποσύνθεση ισχύει μόνο για τον random surfer
- Για τον intelligent surfer πρέπει να καταφύγουμε στις τεχνικές compressed row (column) storage
  - Κερδίζουμε χώρο, αλλά χάνουμε σε επεξεργασία

Compressed Row Storage (CRS) (1/2)

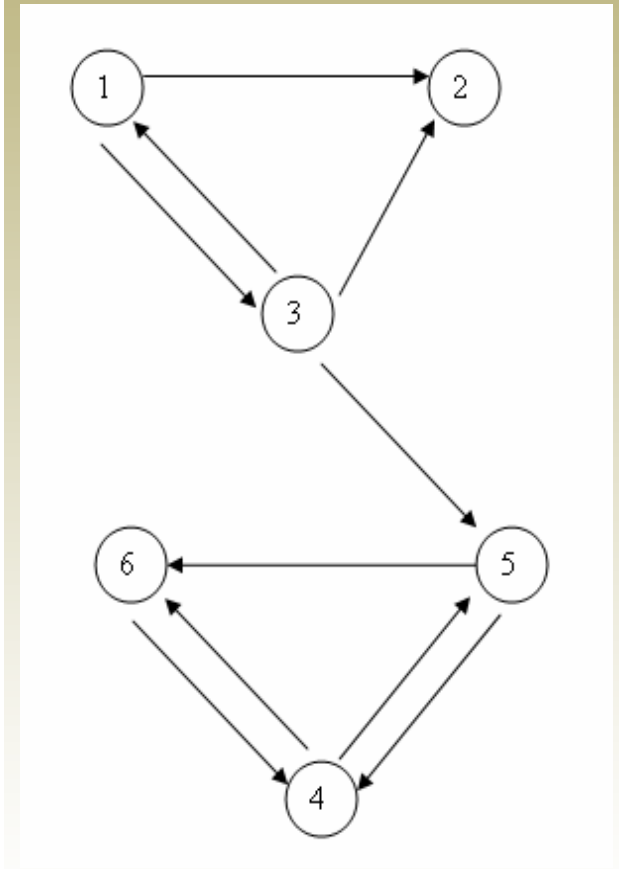
- Απαιτούνται οι εξής πίνακες
  - **val**: αποθηκεύει τις μη-μηδενικές τιμές
  - **col_ind**: αποθηκεύει τους αντίστοιχους δείκτες της στήλης όπου εμφανίζονται μη-μηδενικές τιμές
  - **row_ptr**: αποθηκεύει τις θέσεις του πίνακα val όπου αρχίζουν νέες γραμμές
    - Παραδοσιακά: row_ptr(n+1) = nnz(val) + 1
  - Ισχύει ότι: row_ptr(i+1) – row_ptr(i) = αριθμός των μη-μηδενικών τιμών που υπάρχουν στην i-οστή γραμμή
  - Για έναν πίνακα $\mathbf{A}_{mxn}$: η CRS απαιτεί 2nnz(**A**)+mμ+1, αντί για το mxn που απαιτεί η παραδοσιακή αποθήκευση
- Ανάλογα ισχύουν και για τη CCS

Compressed Row Storage (CRS) (2/2)

$$\begin{pmatrix} 0 & 0.5774 & 0 & 0.4472 & 0.7071 & 0 & 0.7071 \\ 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7071 & 0.7071 \\ 0 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7071 & 0.7071 & 0 \\ 0 & 0 & 0.5774 & 0.4472 & 0 & 0 & 0 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \end{pmatrix}$$

val	col_ind	row_ptr
0.5774	2	1
0.4472	4	5
0.7071	5	7
0.7071	7	9
0.5774	2	10
0.5774	3	12
0.7071	6	14
0.7071	7	16
0.4472	4	18
0.5774	2	**20: nnz(val)**
0.5774	3	
0.7071	1	
0.4472	4	
0.7071	5	
0.7071	6	
0.5774	3	
0.4472	4	
0.7071	1	
0.4472	4	

Λίστες γειτνίασης (Adjacency Lists)



- Adjacency list των στηλών του πίνακα **H** (ή του **L**): θυμηθείτε ότι κάθε στήλη κρατά την inlink πληροφορία για την ιστοσελίδα i
- Ο παρακάτω πίνακας είναι η adjacency list αναπαράσταση για το διπλανό γράφημα

Κόμβος	**Εισερχόμενοι σύνδεσμοι από**
1	3
2	1, 3
3	1
4	5, 6
5	3, 4
6	4, 5

Η τεχνική του κενού (The gap technique)

- Για τη συμπίεση των adjacency lists, δυο σημαντικές τεχνικές
  - Gap technique
  - Reference encoding technique
- Η gap τεχνική εκμεταλλεύεται την τοπικότητα (locality) των διασυνδεδεμένων ιστοσελίδων
  - Η σελίδα "πηγή" και η σελίδα "προορισμός" είναι "κοντά" λεξικογραφικά, δηλ., η σελίδα 100 είναι πιθανότερο να έχει εισερχόμενους υπερσυνδέσμους από τις σελίδες 112, 113, 116, 117, κ.τ.λ., και είναι λιγότερο πιθανό να έχει συνδέσμους από τις π.χ., 924, 4.931.010

Κόμβος	Εισερχόμενοι σύνδεσμοι από
100	112 0 2 0

- Αποθηκεύεται η ετικέτα της πρώτης σελίδας που έχει σύνδεσμο προς την 100, δηλ., η 112, και κατόπιν μόνο τα κενά μεταξύ διαδοχικών σελίδων

Η τεχνική του reference encoding (1/2)

- Εκμεταλλεύεται την ομοιότητα μεταξύ των ιστοσελίδων
  - Εάν οι ιστοσελίδες P_i και P_j έχουν παρόμοιες adjacency lists μπορούμε να αναπαραστήσουμε τη λίστα της P_j με όρους της P_i
  - Η P_i λέγεται reference page της P_j
  - Οι ιστοσελίδες του ίδιου domain συχνά έχουν τους ίδιους εξερχόμενους υπερσυνδέσμους
  - Θεωρήστε το παρακάτω παράδειγμα:

5	7	12	89	101	190	390

5	6	12	50	101	190

reference encode

P_j σε σχέση με τη P_i

1010110

6	50

Η τεχνική του reference encoding (2/2)

- Δημιουργούμε δυο διανύσματα
  - sharing vector (binary):
    - Ίδιο μήκος με την adjacency list της P_i
    - Περιέχει 1 στην k-οστή θέση, εάν το k-οστό στοιχείο της λίστας της P_i εμφανίζεται στη λίστα της P_j
  - dissimilarity vector
    - Περιέχει όλα τα στοιχεία της λίστας της P_j που δεν εμφανίζονται στη λίστα της P_i
- Το sharing vector είναι πιο οικονομικό από την adjacency λίστα
- Η αποτελεσματικότητα της κωδικοποίησης εξαρτάται από τον αριθμό των "διαφορετικών" σελίδων
  - Μεγάλη επικάλυψη σημαίνει μικρό dissimilarity vector
  - Δεν είναι εύκολο να εντοπίσουμε για την κάθε σελίδα μια reference page

Σύγκλιση

- Είδαμε ότι ένας τρόπος εύρεσης του PageRank διανύσματος είναι με την power μέθοδο, η οποία εφαρμόζεται μέχρι να ικανοποιηθεί κάποιο κριτήριο σύγκλισης
- Συνήθως, $||\boldsymbol{\pi}^{(k+1)T} - \boldsymbol{\pi}^{(k)T}||_1 < \tau$
- Ο Taher Haveliwala ορθά παρατήρησε ότι δεν μας ενδιαφέρουν οι ακριβείς τιμές του διανύσματος, αλλά η σωστή διάταξη των τιμών στο διάνυσμα αυτό
- Με ~10 επαναλήψεις μπορούμε να βρούμε τη σωστή διάταξη!!!
- Ερωτήματα:
  - Πώς μετράμε τη διαφορά μεταξύ δυο rankings;
    - Kendall Tau, rank aggregation, set overlap, …
  - Πώς αποφασίζουμε ότι ένα ranking έχει συγκλίνει ικανοποιητικά;
    - Μπορούμε να γράψουμε μια power μέθοδο η οποία να επενεργεί στα rankings και όχι στις τιμές του PageRank σε κάθε επανάληψη;

Power-law στο Web (1/3)

A non-negative random variable X is said to have a Power-Law distribution if, for some constants c>0 and α>0:

Prob[X>x] ~ $x^{-\alpha}$, or equivalently f(x) ~ $x^{-(\alpha+1)}$

Taking logs from both sides, we have:

log Prob[X>x] = -αlog(x) + constant

Power Law distributions have "heavy/long tails", i.e. the probability mass of events whose value is far from the expectancy or median of the distribution is significant

- Unlike Normal or Geometric/Exponential distributions, where the probability mass of the tail decreases exponentially, in Power Law distributions the mass of the tail decreases by the constant power of α
- Another point of view: in an Exponential distribution, f(x)/p(x+k) is constant, whereas in a Power-Law distribution, f(x)/f(kx) is constant.
- The "average" quantity in a Power-Law distribution is not "typical"

Examples of Power-Law distributions are Pareto and Zipf

Power-law στο Web (2/3)

A random variable X follows Zipf's Law (is "Zipfian") with parameter α when the j'th most popular value of X occurs with probability that is proportional to $j^{-\alpha}$

- Essentially the distribution is over the discrete ranks

Whenever $\alpha>1$, X may take an infinite number of values (i.e. have infinitely many different value popularities)

Named after the American Linguist George Kingsley Zipf (1902-1950), who observed it on the frequencies of words in the English language

- On a large corpus of English text, the 135 most frequently occurring words accounted for half of the text

Power-law στο Web (3/3)

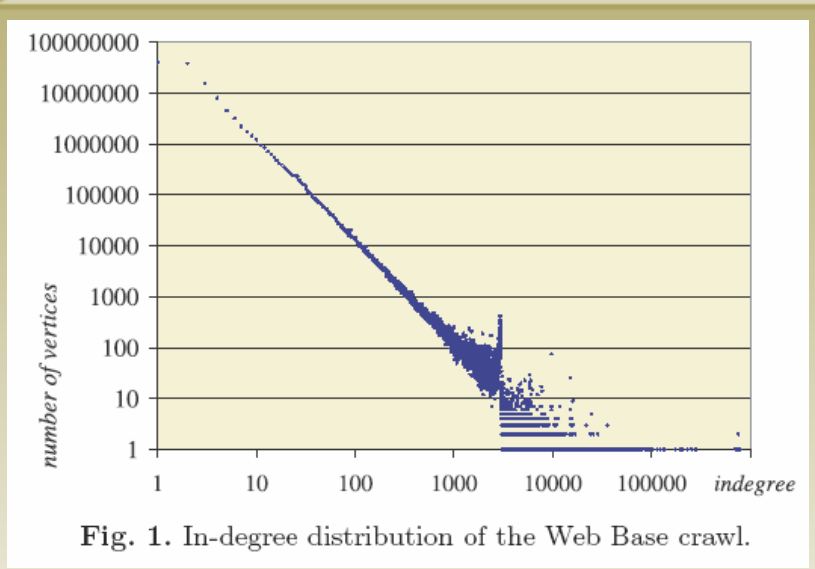


Fig. 1. In-degree distribution of the Web Base crawl.

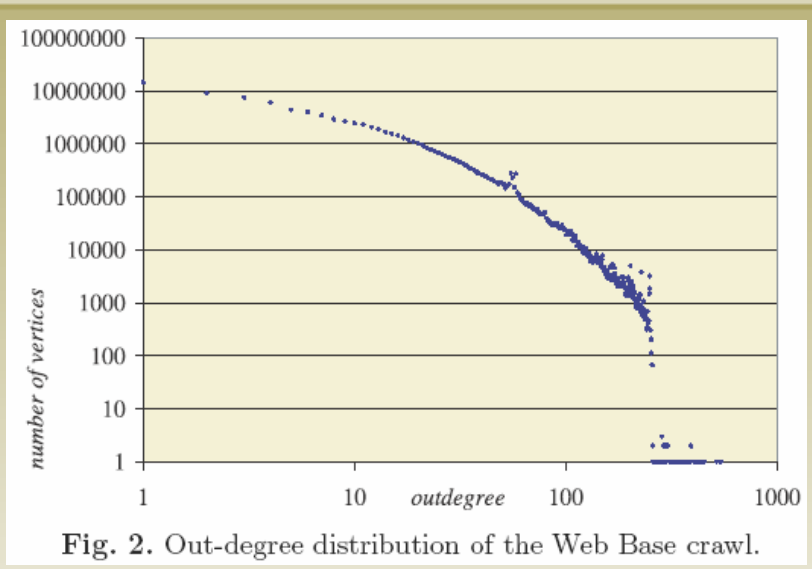


Fig. 2. Out-degree distribution of the Web Base crawl.

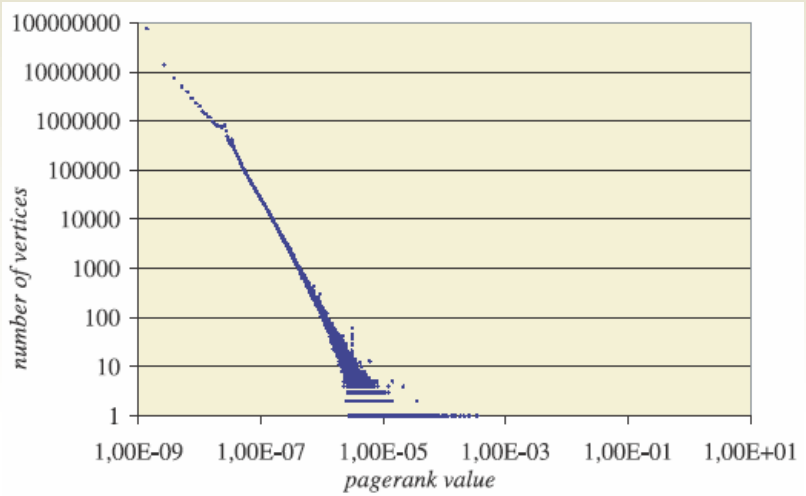


Fig. 4. PageRank distribution of the Web Base crawl.

Ακρίβεια (1/2)

- Αφού οι τιμές του διανύσματος PageRank ακολουθούν power-law είναι πιθανό ένα μικρό τμήμα του διανύσματος να έχει τη μορφή: π^T=(0.000001532, 0.0000015316, 0.0000015312, 0.0000015210)
- Συνεπώς απαιτείται ακρίβεια της τάξης του 10^{-9} για να διακρίνουμε μεταξύ των στοιχείων του διανύσματος
- Φυσικά, παρόλο που οι τιμές του διανύσματος μπορεί να είναι "σφιχτά πακεταρισμένες" σε μερικά τμήματά του, τα στοιχεία του διανύσματος που αφορούν ένα ερώτημα (δηλ., οι τιμές PageRank των ιστοσελίδων που είναι σχετικές με το ερώτημα) είναι πολύ λιγότερο πιθανό να είναι το ίδιο "σφιχτά πακεταρισμένες", και συνεπώς δεν απαιτείται ακρίβεια της τάξης 10^{-12}

Ακρίβεια (2/2)

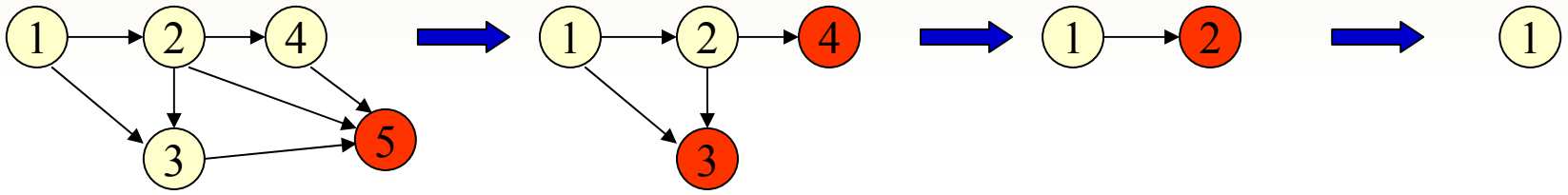
- Αρχικά, οι Page & Brin πρότειναν περίπου 50 επαναλήψεις για να συγκλίνει η power μέθοδος, άρα
  - Οι εκτιμήσεις τους για το $\boldsymbol{\pi}^T$ δεν είναι πολύ ακριβείς, ή
  - Η υπο-κυρίαρχη (subdominant) ιδιοτιμή του πίνακα επαναλήψεων απέχει αρκετά από την κυρίαρχη ιδιοτιμή $\lambda_1=1$
- Την πρώτη εικασία δεν μπορούμε να την ελέγξουμε, γιατί ποτέ δεν δημοσιοποιήθηκαν αποτελέσματα σχετικά με τη σύγκλιση του Google
- Η δεύτερη εικασία υπονοεί ότι ο πίνακας τηλεμεταφοράς $\mathbf{E}=\mathbf{e}\mathbf{v}^T$ κατέχει σημαντικό βάρος, και άρα ίσως έχει τεθεί $\alpha=0.8$
  - Επιταχύνεται η σύγκλιση
  - Απομακρυνόμαστε όμως από την ουσία των υπερσυνδέσμων

Dangling κόμβοι (1/10)

- Υπάρχουν διάφορες μορφές dangling κόμβων
  - Σελίδα με δεδομένα
  - Αρχείο pdf, ps, jpeg, …
  - Ιστοσελίδα που την "κατέβασε"ο crawler, αλλά ακόμα δεν εξερευνήθηκαν οι εξερχόμενοι υπερσύνδεσμοί της, δηλ., το **Web Frontier**
- Οι Page & Brin πρότειναν να αντικατασταθούν οι γραμμές $\mathbf{0}^T$ με το διάνυσμα $\mathbf{e}^T/n$ (ή το πιο γενικό διάνυσμα $\mathbf{v}^T$)
  - Αυτό αυξάνει κατά πολύ τις απαιτήσεις σε αποθηκευτικό χώρο, οπότε ήδη έχουμε δει ότι είναι προτιμότερη η λύση του δυαδικού διανύσματος dangling κόμβων **a**
- Για να είμαστε όμως ακριβείς, οι Page & Brin πρότειναν να απομακρύνουμε τους dangling κόμβους κατά το υπολογισμό του PageRank και να τους προσθέσουμε ξανά, αφού το διάνυσμα PageRank έχει συγκλίνει

Dangling κόμβοι (2/10)

- Αυτό, σε μεγάλο βαθμό, είναι ένα φιλοσοφικό ερώτημα:
  - Εάν αφήσουμε εκτός τους dangling κόμβους, τότε δεν έχουν καμία πιθανότητα να αποκτήσουμε μεγάλη PageRank, οπότε εκ προοιμίου δεν τους θεωρούμε σημαντικούς
    1. Όμως, μια ιστοσελίδα με πολλούς εισερχόμενους συνδέσμους από σημαντικές ιστοσελίδες, αλλά χωρίς εξερχόμενους υπερσυνδέσμους, είναι μια σημαντική ιστοσελίδα
       - Όντως, ο Kevin McCurley έδειξε ότι (ACM WWW conf. 2004, "Ranking the Web frontier") σε μικρά γραφήματα, αλλά και σε τεράστια γραφήματα μερικοί dangling κόμβοι μπορεί να έχουν μεγαλύτερη τιμή PageRank από πολλούς non-dangling κόμβους
    2. Επιπλέον, η διαδικασία αφαίρεσης των dangling κόμβων μπορεί να δημιουργήσει επαναληπτικά νέους και νέους dangling κόμβους και τελικά να μην μείνει κανένας κόμβος στο γράφημα



Dangling κόμβοι (3/10)

- Μια καλύτερη λύση είναι να θεωρήσουμε όλους τους κόμβους ισότιμα από την αρχή, αλλά να εκμεταλλευτούμε τις μοναδικές τους ιδιότητες. Το έκαναν οι:
  - Lee, Golub & Zenios (Technical Report SCCM-2003-15)
  - Eiron, McCurley & Tomlin (ACM WWW 2004)
  - Langville & Meyer (SIAM Journal on Scientific Computing, vol. 27, no. 6, 2006)
- Στη θεμελίωση του PageRank προβλήματος που έχουμε δώσει έως τώρα, είτε ως πρόβλημα ιδιοδιανύσματος είτε ως γραμμικό σύστημα, θεωρήσαμε ισότιμους όλους τους κόμβους, αλλά δεν εκμεταλλευτήκαμε την ιδιομορφία τους
- Στην ουσία
  - όλοι οι dangling κόμβοι έχουν την ίδια φύση σχετικά με τις γραμμές τους στον πίνακα **H** (στον **S** και **G**)
  - όταν ο random surfer φτάνει σε έναν dangling κόμβο συμπεριφέρεται το ίδιο: τηλεμεταφέρεται σε έναν άλλο κόμβο (είτε ομοιόμορφα $\mathbf{e}^T/n$, είτε με βάση το διάνυσμα προσωποποίησης $\mathbf{v}^T$)

Dangling κόμβοι (4/10)

- Συνεπώς, γιατί να μην συμπτύξουμε όλους τους danling κόμβους σε έναν;
- Αυτό έχει ως συνέπεια την ελάττωση του μεγέθους του προβλήματος, ειδικά εάν το ποσοστό των danling κόμβων είναι μεγάλο
- Όμως, η επίλυση του μικρότερου (|ND|+1) x (|ND|+1) προβλήματος δημιουργεί δυο νέα:
  - Έχουμε στη διάθεσή μας μόνο τις τιμές PageRank των non-danling κόμβων καθώς και του κόμβου που προέκυψε από την σύμπτυξη των dangling κόμβων, αλλά όχι τις τιμές PageRank των επιμέρους dangling κόμβων
  - Αυτό το μικρότερο σύνολο των rankings είναι πολωμένο
- Η απάντηση στα δυο ερωτήματα αυτά δίνεται από τις τεχνικές της aggregation και της stochastic complementation

Dangling κόμβοι (5/10)

- Εδώ όμως θα δώσουμε μια απλούστερη εξήγηση χωρίς πολλούς μαθηματικούς όρους
- Έστω ότι αναδιατάσσουμε τις γραμμές του πίνακα H οι οποίες αντιστοιχούν στους dangling κόμβους, ώστε να εμφανίζονται στο κάτω μέρος του πίνακα

$$\mathbf{H} = \begin{array}{c} \\ \text{ND} \\ \text{D} \end{array} \begin{array}{c} \begin{array}{cc} \text{ND} & \text{D} \end{array} \\ \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \end{array}$$

όπου ND είναι το σύνολο των non-dangling κόμβων και D είναι το σύνολο των dangling κόμβων

- Ο πίνακας συντελεστών ($\boldsymbol{\pi}^T(\mathbf{I}-\alpha\mathbf{H})=\mathbf{v}^T$), στη διατύπωση ως αραιό γραμμικό σύστημα, γίνεται πλέον

$$(\mathbf{I} - \alpha\mathbf{H}) = \begin{pmatrix} \mathbf{I} - \alpha\mathbf{H}_{11} & -\alpha\mathbf{H}_{12} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

και ο αντίστροφος αυτού:

$$(\mathbf{I} - \alpha\mathbf{H})^{-1} = \begin{pmatrix} (\mathbf{I} - \alpha\mathbf{H}_{11})^{-1} & \alpha(\mathbf{I} - \alpha\mathbf{H}_{11})^{-1}\mathbf{H}_{12} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

Dangling κόμβοι (6/10)

- Επομένως, το μη-κανονικοποιημένο διάνυσμα PageRank

$$\mathbf{x}^T = \mathbf{v}^T(\mathbf{I}-\alpha\mathbf{H})^{-1}$$

μπορεί να γραφεί ως

$$\mathbf{x}^T = (\mathbf{v}_1^T(\mathbf{I}-\alpha\mathbf{H}_{11})^{-1} \quad | \quad \alpha\mathbf{v}_1^T(\mathbf{I}-\alpha\mathbf{H}_{11})^{-1}\mathbf{H}_{12}+\mathbf{v}_2^T)$$

όπου το διάνυσμα προσωποποίησης $\mathbf{v}^T$ έχει διαμεριστεί στο τμήμα $\mathbf{v}_1^T$ για τους non-dangling κόμβους και στο τμήμα $\mathbf{v}_2^T$ για τους dangling κόμβους

- Ο αλγόριθμος που υπολογίζει το διάνυσμα PageRank κάνοντας χρήση μόνο το non-dangling τμήμα του Web δίνεται παρακάτω:

Αλγόριθμος-1 PageRank με dangling κόμβους

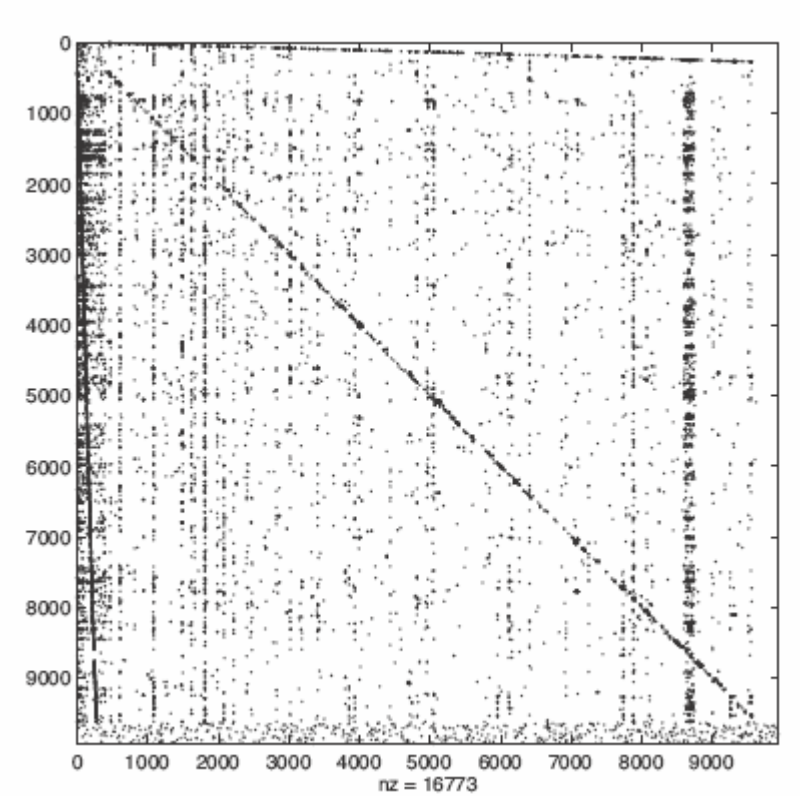
- Επίλυση ως προς $\mathbf{x}_1^T$ του: $\mathbf{x}_1^T(\mathbf{I}-\alpha\mathbf{H}_{11}) = \mathbf{v}_1^T$
- Υπολογισμός του: $\mathbf{x}_2^T = \alpha x_1^T\mathbf{H}_{12} + \mathbf{v}_2^T$
- Κανονικοποίηση: $\pi^T = [\mathbf{x}_1^T \quad \mathbf{x}_2^T] \;/\; \|[\mathbf{x}_1^T \quad \mathbf{x}_2^T]\|_1$

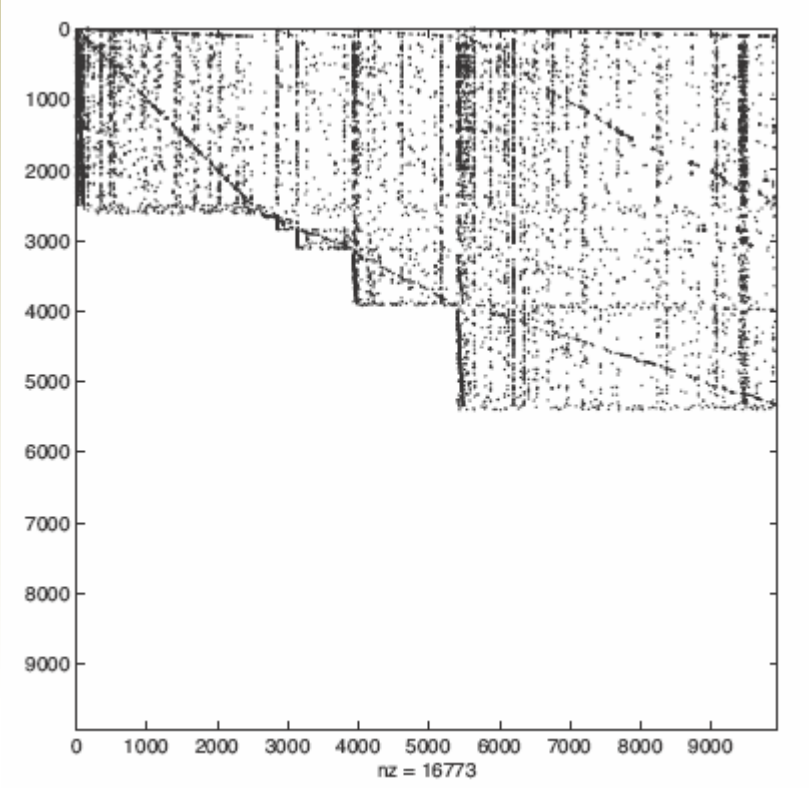
Dangling κόμβοι (7/10)

- Ο αλγόριθμος αυτός είναι ισοδύναμος με τον αντίστοιχο που πρότειναν οι Lee, Golub & Zenios, ο οποίος μπορεί να ελαττώσει τον υπολογισμό κατά 1/5 σε ένα γράφημα με 80% dangling κόμβους
- Μπορούμε να επιτύχουμε κάτι καλύτερο;
- Δηλ., μπορούμε να βρούμε "μηδενικές γραμμές" $\mathbf{0}^T$ σε υποπίνακες του **H**;
- Όντως, μπορούμε να εφαρμόζουμε αναδρομικά τη διαδικασία αναδιάταξης γραμμών/στηλών του H, ώστε να οδηγηθούμε σε υποπίνακες που δεν περιέχουν καθόλου μηδενικές γραμμές

Dangling κόμβοι (8/10)

- Για παράδειγμα, ένας πίνακας **H** με 9664 γραμμές, που περιέχει συνολικά 16773 μη μηδενικά στοιχεία, μπορεί να αναδιαταχτεί αναδρομικά με την προηγούμενη διαδικασία





Dangling κόμβοι (9/10)

- Γενικά, μετά από αυτή τη συμμετρική αναδιάταξη, ο πίνακας συντελεστών του γραμμικού συστήματος του PageRank έχει την εξής δομή:

$$(\mathbf{I}-\alpha\mathbf{H}) = \begin{pmatrix} \mathbf{I}-\alpha\mathbf{H}_{11} & -\alpha\mathbf{H}_{12} & -\alpha\mathbf{H}_{13} & \cdots & -\alpha\mathbf{H}_{1b} \\ & \mathbf{I} & -\alpha\mathbf{H}_{23} & \cdots & -\alpha\mathbf{H}_{2b} \\ & & \mathbf{I} & \cdots & -\alpha\mathbf{H}_{3b} \\ & & & \ddots & \\ & & & & \mathbf{I} \end{pmatrix}$$

  όπου b είναι ο αριθμός των τετραγωνικών διαγωνίων μπλοκ στον αναδιαταγμένο πίνακα

- Επομένως το αναδιατεταγμένο σύστημα μπορεί να λυθεί με forward substitution
- Το μόνο σύστημα που πρέπει να λυθεί άμεσα είναι το πρώτο υποσύστημα $\mathbf{x}_1^T(\mathbf{I}-\alpha\mathbf{H}_{11}) = \mathbf{v}_1^T$ όπου τα $\boldsymbol{\pi}^T$ και $\mathbf{v}^T$ έχουν διαμεριστεί κατάλληλα
- Τα υπόλοιπα υποδιανύσματα του $\mathbf{x}^T$ υπολογίζονται γρήγορα με forward substitution

Dangling κόμβοι (10/10)

Αλγόριθμος-2 PageRank με dangling κόμβους

- Αναδιατάσσουμε τον **H**, ώστε να επιτύχουμε τη δομή που δείξαμε
- Επίλυση ως προς $\mathbf{x}_1^T$ του: $\mathbf{x}_1^T(\mathbf{I} - \alpha\mathbf{H}_{11}) = \mathbf{v}_1^T$
- Για i=2 μέχρι b, υπολογισμός του: $\mathbf{x}_i^T = \alpha \sum_{j=1}^{i-1} \mathbf{x}_j^T \mathbf{H}_{ji} + \mathbf{v}_i^T$
- Κανονικοποίηση: $\pi^T = [\mathbf{x}_1^T \ \ \mathbf{x}_2^T \ \ \ldots \mathbf{x}_b^T] \ / \ \|[\mathbf{x}_1^T \ \ \mathbf{x}_2^T \ \ \ldots \mathbf{x}_b^T]\|_1$

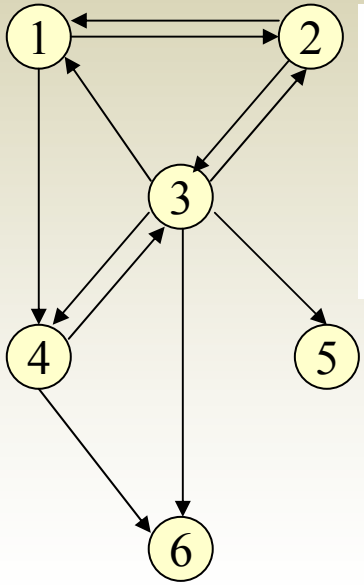
- Στο παράδειγμα με τον αραιό πίνακα που δείξαμε, λύνουμε τελικά ένα σύστημα 2622 x 2622 αντί για το αρχικό 9664 x 9664
- Το μικρό υποσύστημα $\mathbf{x}_1^T(\mathbf{I}-\alpha\mathbf{H}_{11})=\mathbf{v}_1^T$ μπορεί να λυθεί με μια ευθεία μέθοδο (εάν είναι αρκετά μικρό) ή με μια επαναληπτική μέθοδο (π.χ., Jacobi)

Μοντελοποίηση του “back button” (1/6)

- Το αρχικό μοντέλο του PageRank δεν λάμβανε υπόψη το “back button”
- Η εισαγωγή του περιπλέκει την κατάσταση
- Άλλωστε, η θεμελιώδης ιδιότητα της συγκεκριμένη Markov αλυσίδας είναι ότι δεν έχει μνήμη (***memoryless property***)
- Αρκετοί προσπάθησαν να λάβουν υπόψη τους το “back button”
- Υπάρχουν διάφοροι τρόποι να μοντελοποιήσουμε το “back button”
- Μια πάρα πολύ απλή μεθοδολογία είναι η εξής: “*…όταν φτάσουμε σε έναν dangling κόμβο χρησιμοποιούμε το back-button για να επιστρέψουμε στη σελίδα απ΄ όπου ήρθαμε*”

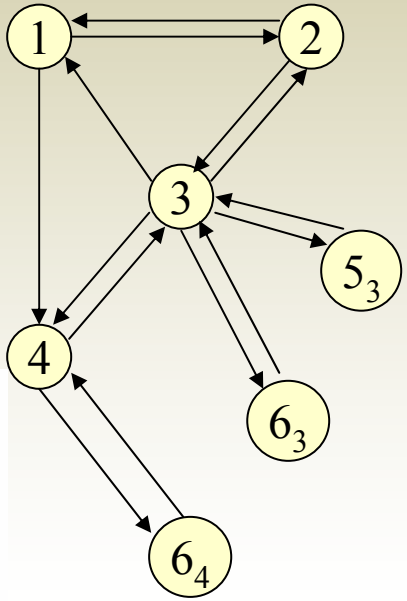
Μοντελοποίηση του “back button” (2/6)

- Αυτή η προσέγγιση όμως μοντελοποιεί το “back-button” μόνο για τους dangling κόμβους
- Επίσης, δυστυχώς μας οδηγεί στο να προσθέσουμε έναν νέο κόμβο για κάθε εισερχόμενο υπερσύνδεσμο ενός dangling κόμβου
- Ευτυχώς, ο πίνακας που προκύπτει $\hat{\mathbf{H}}$ έχει όμορφη δομή



$$\mathbf{H} = \begin{pmatrix} 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/5 & 1/5 & 0 & 1/5 & 1/5 & 1/5 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\hat{\mathbf{H}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5_3 \\ 6_3 \\ 6_4 \end{array} \begin{pmatrix} 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 1/5 & 1/5 & 0 & 1/5 & 1/5 & 1/5 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$



Μοντελοποίηση του “back button” (3/6)

- Ο $\hat{\mathbf{H}}$ είναι row-stochastic
- Πρέπει όμως να εφαρμοστεί μια διόρθωση ώστε να γίνει irreducible
- Συγκεντρωτικά, τα βήματα για την κατασκευή του $\hat{\mathbf{H}}$ είναι τα παρακάτω:
  - **ΒΗΜΑ 1**. Αναδιατάσσουμε τον **H**, ώστε να έχουμε:

$$\mathbf{H} = \begin{array}{c} \\ \text{ND} \\ \text{D} \end{array} \begin{array}{c} \begin{array}{cc} \text{ND} & \text{D} \end{array} \\ \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \end{array}$$

  - **ΒΗΜΑ 2**. Για κάθε εισερχόμενο σύνδεσμο ενός dangling κόμβου, δημιουργούμε έναν κόμβο παλινδρόμησης (bounce-back node). Συνολικά θα υπάρχουν $\text{nnz}(\mathbf{H}_{12})$ τέτοιοι κόμβοι, αντί για τους αρχικούς |D| dangling κόμβους
    - Εάν κάθε dangling κόμβος έχει περισσότερους από έναν εισερχόμενους κόμβους, τότε θα δημιουργηθούν πολλοί περισσότεροι κόμβοι παλινδρόμησης σε σχέση με το πόσοι είναι οι dangling κόμβοι
    - Ο πίνακας υπερσυνδέσμων με τους κόμβους παλινδρόμησης έχει την εξής μορφή μπλοκ:

$$\hat{\mathbf{H}} = \begin{array}{c} \\ \text{ND} \\ \text{BB} \end{array} \begin{array}{c} \begin{array}{cc} \text{ND} & \text{BB} \end{array} \\ \begin{pmatrix} \hat{\mathbf{H}}_{11} & \hat{\mathbf{H}}_{12} \\ \hat{\mathbf{H}}_{21} & \mathbf{0} \end{pmatrix} \end{array}$$

Μοντελοποίηση του “back button” (4/6)

- **ΒΗΜΑ 3**. Σχηματίζουμε τα τρία μη-μηδενικά μπλοκ του $\hat{\mathbf{H}}$
  - Πρώτα, $\hat{\mathbf{H}}_{11}=\mathbf{H}_{11}$
  - Κατόπιν, υπάρχει συμμετρία στη δομή των $\hat{\mathbf{H}}_{12}$ και $\hat{\mathbf{H}}_{21}$ που μπορούμε να εκμεταλλευτούμε:
    - Δηλαδή, εάν το στοιχείο (i,j) του $\hat{\mathbf{H}}_{12}$ είναι μη μηδενικό, τότε το στοιχείο (j,i) του $\hat{\mathbf{H}}_{21}$ = 1
    - Επιπλέον, ενώ το μέγεθος του $\hat{\mathbf{H}}$ μπορεί να είναι πολύ μεγαλύτερο από το μέγεθος του $\mathbf{H}$, ο $\hat{\mathbf{H}}$ έχει $\text{nnz}(\mathbf{H}_{12})$ περισσότερα μη μηδενικά στοιχεία από τον $\mathbf{H}$, και όλα αυτά είναι ο ακέραιος 1

- Για να υπολογίσουμε το “παλινδρομικό” διάνυσμα PageRank, απλά εκτελούμε οποιονδήποτε αλγόριθμο PageRank
- Φυσικά, ο αλγόριθμος θα είναι ελαφρά τροποποιημένοι, αφού ο $\hat{\mathbf{H}}$ είναι επίσης στοχαστικός
- Συνεπώς, η “παλινδρομική” power μέθοδος θα είναι:

$$\hat{\pi}^{(k+1)T} = \hat{\pi}^{(k)T}\hat{\mathbf{G}}$$
$$= \alpha\hat{\pi}^{(k)T}\hat{\mathbf{H}} + (1-\alpha)\mathbf{v}^T$$

Μοντελοποίηση του “back button” (5/6)

- Το “παλινδρομικό” διάνυσμα PageRank για τον $\hat{\mathbf{H}}$ είναι φυσικά μεγαλύτερο από το τυπικό PageRank διάνυσμα του **H**
- Για να συγκρίνουμε τα δυο διανύσματα, απλά συγχωνεύουμε τους πολλαπλούς παλινδρομικούς κόμβους που αντιστοιχούν σε έναν dangling κόμβο, σε έναν κόμβο
- Για το προηγούμενο παράδειγμα, με α=0.85 και $\mathbf{v}^T=\mathbf{e}^T/n$

$$
\begin{array}{lcccccccc}
 & & 1 & 2 & 3 & 4 & 5 & 6 & \\
\pi^T(\mathbf{H}) & = & (0.1726 & 0.1726 & 0.2102 & 0.1726 & 0.0993 & 0.1726) & \\
 & & & & & & & & \\
 & & 1 & 2 & 3 & 4 & 5_3 & 6_3 & 6_4 \\
\hat{\pi}^T(\hat{\mathbf{H}}) & = & (0.1214 & 0.1214 & 0.2846 & 0.2186 & 0.0698 & 0.0698 & 0.1143)
\end{array}
$$

Μοντελοποίηση του “back button” (6/6)

- Το “συγχωνευμένο” διάνυσμα PageRank για τον $\hat{\mathbf{H}}$ είναι το:

$$\hat{\pi}^T(\mathbf{H}) = (0.1214 \quad 0.1214 \quad 0.2846 \quad 0.2186 \quad 0.0698 \quad 0.1841)$$

- Το ranking των σελίδων με βάση το $\boldsymbol{\pi}^{\mathrm{T}}$ είναι (3 1/2/4/6 5)
- Το ranking των σελίδων με βάση το $\tilde{\mathbf{n}}^{\mathrm{T}}$ είναι (3 4 6 1/2 5)
- Φυσικά το παράδειγμα είναι μικρό και έτσι η διαφορά στο ranking κατέστη προφανής