Steady-state Analysis of Google-like Matrices with Structured Methods

Tuğrul Dayar
Department of Computer Engineering
Bilkent University
tugrul@cs.bilkent.edu.tr
(joint work with Gökçe Nil Noyan)

11-16 February 2007
Dagstuhl Seminar 07071
Research supported by grants from TÜBA-GEBİP and TÜBİTAK
Outline

- Background
  - Example
  - Problem description

- Computing steady-state vector
  - Power method
  - Quadractically extrapolated power method
  - Iterative methods based on splittings
  - Some results

- Computing cutsets
  - Definition
  - Example
  - Two methods
  - Cutsets for reducible matrices
  - Some results

- Partitioning Google-like matrices
  - Two block partitionings
  - Some results

- Conclusion
Matrices used by Google search engine in ranking web pages are among largest of Markov chains (MCs).

These are “stochastic” matrices with properties:

- sparse,
- reducible,
- some zero rows.

Ranking web pages amounts to solving for positive left-hand eigenvectors of linear combinations of these matrices with appropriately chosen rank-1 matrices.

Since matrices are extremely large and always changing, computation lasts long and needs to be repeated.
Consider

\[
P = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\
4 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\
5 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \\
6 & 0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}
\]

A. N. Langville, C. D. Meyer,
*A survey of eigenvector methods for web information retrieval*,

\(P\) corresponds to *web graph* of 6 pages, where

- probabilities of outgoing links are uniformly distributed,
- page 2 does not have any outgoing links.
Problem description

Given $P$ as Google matrix:

- let $P' = P + av^T$, where
  - $a = e - P e$ ($a_i = 1$ implies node $i$ is dangling),
  - $e$ is vector of 1s,
  - $v$ is personalization vector
    (mostly, $v = e/n$, uniform distribution);
- $P'$ has row sums of 1;

- let $P'' = \alpha P' + (1 - \alpha)ev^T$, where
  - $(1 - \alpha)$ is teleportation probability; $0 < \alpha < 1$
    (mostly, $\alpha = 0.85$);
- $P''$ is full, implying it is irreducible and acyclic;

- compute steady-state vector, $\pi$, in
  $\pi P'' = \pi$ such that $\pi > 0, \pi e = 1$.
Computing steady-state vector

L. Page, S. Brin, R. Motwani, T. Winograd,
*The pagerank citation ranking: Bringing order to the web*,
Technical Report 1999-66, Stanford University, California, 1999:

- problem is introduced during Stanford Digital Library Technologies project;

- solution algorithm based on power method is called PageRank.
Given
\[ \pi^{(0)} > 0 \quad \text{and} \quad \pi^{(0)} e = 1, \]

\[ \pi^{(k+1)} = \pi^{(k)}(\alpha P) + (\pi^{(k)}(\alpha a) + (1 - \alpha))v^T \quad \text{for } k = 0, 1, \ldots \]
can be implemented with
- one vector-matrix multiplication using \( \alpha P \),
- two level-1 operations (i.e., \textit{dot-product} and \textit{saxpy}).

Convergence takes place at rate by which \( \alpha^k \to 0 \).

Smaller \( \alpha \) is, lesser effect of hyperlink structure of web.
Quadratically extrapolated power method

Use power method, but periodically (say, every 10 iterations):

- **compute**
  
  \[
  \begin{align*}
  y^{(k-2)} &= \pi^{(k-2)} - \pi^{(k-3)} \\
  y^{(k-1)} &= \pi^{(k-1)} - \pi^{(k-1)} \\
  y^{(k)} &= \pi^{(k)} - \pi^{(k-3)}
  \end{align*}
  \]

- **solve**
  
  \[
  \begin{pmatrix} y^{(k-2)} & y^{(k-1)} \end{pmatrix} \begin{pmatrix} \gamma_1 & \gamma_2 \end{pmatrix}^T = -y^{(k)}
  \]
  using truncated QR factorization;

- **update**
  
  \[
  \pi^{(k)} = (\gamma_1 + \gamma_2 + 1)\pi^{(k-2)} + (\gamma_2 + 1)\pi^{(k-1)} + \pi^{(k)}
  \]

Iterative methods based on splittings

Let $A = P^{TT} - I$, $x = \pi^T$, and consider splitting $A = D - L - U = M - N$ for $Ax = 0$, where

\[
D = \text{diag}(A_{1,1}, A_{2,2}, \ldots, A_{N,N}),
\]

\[
-L = \begin{pmatrix}
A_{2,1} & A_{3,2} & \cdots & A_{N,N-1} \\
A_{3,1} & A_{3,2} & \cdots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
A_{N,1} & A_{N,2} & \cdots & A_{N,N-1}
\end{pmatrix},
\]

\[
-U = \begin{pmatrix}
A_{1,2} & A_{1,3} & \cdots & A_{1,N} \\
A_{2,3} & \cdots & A_{2,N} & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
A_{N-1,2} & A_{N-1,3} & \cdots & A_{N-1,N}
\end{pmatrix},
\]

$1 < N \leq n_A$, and $M$ is nonsingular.
Given $x^{(0)} > 0$ and $e^T x^{(0)} = 1$, 

\[
M x^{(k+1)} = N x^{(k)} \quad \text{for} \quad k = 0, 1, \ldots,
\]

where \( M = D, N = L + U \) is (block) Jacobi, (B)Jacobi, \( M = D - L, N = U \) is (block) Gauss-Seidel, (B)GS.

These become point methods when \( N = n_A \).

For updating steady-state vector:

A. N. Langville, C. D. Meyer,

*Updating Markov chains with an eye on Google’s PageRank*,


For generalization of quadratic extrapolation and related issues:

C. Brezinski, M. Redivo-Zaglia,

*The PageRank vector: Properties, computation, approximation, and acceleration*,

Some results

Platform: 3.4 GHz Pentium IV processor and a 2 Gigabytes main memory under Cygwin.

Table 1: Results for two web graphs for $\alpha \in \{0.85, 0.99\}$ and $v = e/n$.

<table>
<thead>
<tr>
<th>Matrix</th>
<th>$n_A$</th>
<th>$nz_A / nz_P$</th>
<th>$\alpha$</th>
<th>Method</th>
<th>Iter</th>
<th>TimeSol</th>
<th>Res</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford</td>
<td>281,903</td>
<td>2,312,497</td>
<td>0.85</td>
<td>POWER</td>
<td>103</td>
<td>6.5</td>
<td>8e-11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>QPOWER</td>
<td>86</td>
<td>5.7</td>
<td>9e-11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JACOBI</td>
<td>104</td>
<td>8.3</td>
<td>7e-11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GS</td>
<td>45</td>
<td>3.7</td>
<td>6e-11</td>
</tr>
<tr>
<td></td>
<td>2,594,400</td>
<td></td>
<td>0.99</td>
<td>POWER</td>
<td>1,319</td>
<td>83.7</td>
<td>1e-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>QPOWER</td>
<td>1,073</td>
<td>70.2</td>
<td>1e-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JACOBI</td>
<td>1,323</td>
<td>104.8</td>
<td>1e-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GS</td>
<td>574</td>
<td>45.4</td>
<td>6e-11</td>
</tr>
<tr>
<td>In2004</td>
<td>1,382,908</td>
<td>16,917,053</td>
<td>0.85</td>
<td>POWER</td>
<td>100</td>
<td>11.8</td>
<td>4e-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>QPOWER</td>
<td>82</td>
<td>10.9</td>
<td>6e-11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JACOBI</td>
<td>100</td>
<td>14.6</td>
<td>4e-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GS</td>
<td>57</td>
<td>8.5</td>
<td>6e-11</td>
</tr>
<tr>
<td></td>
<td>18,299,961</td>
<td></td>
<td>0.99</td>
<td>POWER</td>
<td>1,379</td>
<td>161.0</td>
<td>2e-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>QPOWER</td>
<td>792</td>
<td>105.6</td>
<td>7e-11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>JACOBI</td>
<td>1,379</td>
<td>214.4</td>
<td>2e-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GS</td>
<td>757</td>
<td>119.2</td>
<td>1e-10</td>
</tr>
</tbody>
</table>

Remark: POWER and QPOWER use $P$ as iteration matrix. Hence, for them $nz_P = nz_A - n_A$. 

Dagstuhl Seminar 07071

11-16 February 2007 – 11 / 32
Computing cutsets

Let $A$ be given $(n_A \times n_A)$ *irreducible* matrix with

- $n_A$ nonzero elements,
- *zero-free* diagonal.

Without loss of generality, problem is to compute *permutation* matrix $Q$ such that

$$QAQ^T = \begin{pmatrix} n_T & n_C \\ n_T & n_C \end{pmatrix} \begin{pmatrix} T & X \\ Y & C \end{pmatrix},$$

where

- $n_A = n_T + n_C$,
- $T$ is $(n_T \times n_T)$ *triangular* submatrix.

$Q$ together with $n_T$ defines symmetric permutation on $A$ yielding $(2 \times 2)$ block partition in which first diagonal block of order $n_T$ is *triangular* and *invertible*.
Let

\[ \mathcal{V} = \{1, 2, \ldots, n_A\} \] be node (or vertex) set,
\[ \mathcal{E} = \{(i, j) \mid a_{i,j} \neq 0, i \neq j, \text{ and } i, j \in \mathcal{V}\} \] be edge (or arc) set

of directed graph (digraph) \( G(\mathcal{V}, \mathcal{E}) \) associated with off-diagonal part of \( A \).

Irreducibility of \( A \) translates to strong connectedness of \( G(\mathcal{V}, \mathcal{E}) \), that is, reachability of each node by following sequence of edges from every other node in \( \mathcal{V} \).

Problem then becomes one of computing cutset (or feedback vertex set), \( \mathcal{C} \subset \mathcal{V} \), whose elements cut all cycles in \( G(\mathcal{V}, \mathcal{E}) \).

If \( \mathcal{C} \) and edges incident on \( \mathcal{C} \) are removed from \( G(\mathcal{V}, \mathcal{E}) \) to give node set \( \mathcal{T} = \mathcal{V} - \mathcal{C} \), then resulting subgraph should become acyclic.
Definition (continued)

- Submatrix corresponding to acyclic subgraph is denoted by triangular matrix $T$.
- Smaller $n_C = |C|$ is, larger $n_T = |T|$, and in general, # of nonzeros, $nz_T$, in $T$ become.
- Objective is then to compute as large and as fast $T$ as possible.

**Remark:** If diagonal elements of $A$ are not omitted when forming $G(V, E)$, one obtains $C = V$ since each diagonal element in $A$ represents different cycle which needs to be cut.

T. Dayar,
*Obtaining triangular diagonal blocks in sparse matrices using cutsets*,
Technical Report BU-CE-0701, Department of Computer Engineering,
Bilkent University, Ankara, Turkey, January 2007.
Definition (continued)

- **Cycle**: sequence of edges \((v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k)\) in \(G(\mathcal{V}, \mathcal{E})\) with \(v_1 = v_k\), where
  - \(v_i \in \mathcal{V}\) for \(i \in \{1, 2, \ldots, k\}\),
  - \(v_i\) are distinct except \(v_1\) and \(v_k\),
  - \((v_j, v_{j+1}) \in \mathcal{E}\) for \(j \in \{1, 2, \ldots, k - 1\}\).

- **Cycle**, sometimes called simple cycle, is represented as \((v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow \cdots \rightarrow v_{k-1} \rightarrow v_k)\).

- Length of cycle is \((k - 1)\) and \(k - 1 \leq n_A\).
Consider

\[
A = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & X & X & & & & & \\
2 & & X & & X & & & \\
3 & & & X & & X & & \\
4 & & & & & X & X & X \\
5 & & & & & & X & X & X \\
6 & & & & X & X & & X \\
7 & & & & & X & & X \\
8 & & & & X & & & X
\end{pmatrix}.
\]

\(A\) is irreducible, has zero-free diagonal, and \(G(\mathcal{V}, \mathcal{E})\) has five cycles:

\((1 \to 5 \to 6 \to 2 \to 8 \to 3 \to 1), \quad (1 \to 5 \to 7 \to 4 \to 6 \to 2 \to 8 \to 3 \to 1), \quad (4 \to 5 \to 6 \to 4), \quad (4 \to 6 \to 4), \quad \text{and} \quad (5 \to 7 \to 4 \to 5).\)
Cutset must include at least two nodes one of which is either 4 or 6 due to need to cut cycle of length two:

- if we choose to put 4 in cutset, all cycles but one of length six are cut and we can choose any of \{1, 2, 3, 5, 6, 8\} to be second node in cutset;
- if we choose to put 6 in cutset, all cycles but last one of length three are cut, implying our second node of choice can be any of \{4, 5, 7\}.

There are 8 different minimum cutsets of cardinality two.

Although \textit{shape of triangularity} defines a permutation on nodes in \(T\), nodes in \(C\) can be permuted arbitrarily.
Example (continued)

Assume that $\mathcal{C} = \{4, 6\}$ and

\[
\begin{bmatrix}
2 & 8 & 3 & 1 & 5 & 7 & 4 & 6 \\
2 \\
8 \\
3 \\
1 \\
5 \\
7 \\
4 \\
6
\end{bmatrix}
\]

for which $\mathcal{T} = \{1, 2, 3, 5, 7, 8\}$, $n_T = 6$, and $n_C = 2$.

Using column partitioning, $Q = (e_2 \ e_8 \ e_3 \ e_1 \ e_5 \ e_7 \ e_4 \ e_6)^T$, where $e_j$ denotes $j$th principal axis vector; or equivalently $q = (2 \ 8 \ 3 \ 1 \ 5 \ 7 \ 4 \ 6)^T$. 

\[
\begin{array}{cccccccc}
\hline
\quad & 2 & 8 & 3 & 1 & 5 & 7 & 4 & 6 \\
2 & X & X & & & & & & \\
8 & & X & X & & & & & \\
3 & & & X & X & & & & \\
1 & & & & X & X & & & \\
5 & & & & & X & X & & \\
7 & & & & & X & X & & \\
4 & & & & & & X & X & X \\
6 & & & & & & & X & X \\
\hline
\end{array}
\]

\[
QAQ^T = 
\begin{pmatrix}
X & X & & & & & & \\
X & X & & & & & & \\
X & X & & & & & & \\
X & X & & & & & & \\
X & X & & & & & & \\
X & X & & & & & & \\
X & X & & & & & & \\
X & X & & & & & & \\
\end{pmatrix}
\]
Two methods

Comprehensive survey on *feedback set problems*:


- Various versions of these problems arise in combinatorial circuit design, constraint satisfaction, and operating systems.
- Computing minimum cutset of general graph is *NP-complete* (Karp'72).
- There are certain classes of graphs for which problem is solvable in polynomial time.
- One such class is reducible (flow) graphs (Hecht & Ullman'74).
Reducible graphs arise, for instance, in analysis of program flows with purpose of code optimization and detecting/breaking deadlocks.

They are defined by

- existence of a node, called root (or initial node), from which every other node in $\mathcal{V}$ is reachable by following a sequence of edges;

- uniqueness of directed acyclic graph (dag) generated by different depth first search (DFS) orders (Tarjan’72) of $G(\mathcal{V}, \mathcal{E})$ starting from root.

Graph reducibility not to be mixed with matrix reducibility.

Reducible matrix is one whose corresponding graph is not strongly connected.

There are reducible graphs which are strongly connected and non-reducible graphs which are not strongly connected.
Shamir’79 gives algorithm that computes *minimum cutset* for reducible graph in time and space linear in $(|V| + |E|)$:

- algorithm may abort on non-reducible graph;

- when it does not abort on non-reducible graph, cutset produced by algorithm is also minimum;

- non-reducible graphs for which Shamir’s algorithm do not abort are named *quasi-reducible* (Rosen’82).

Rosen gives algorithm Cutfind, which runs in linear time and space, also computing cutsets of graphs that are not (quasi-)reducible:

- cutsets computed by Cutfind may not be minimum for graphs that are not (quasi-)reducible.

- on an example, which is not strongly connected, Rosen shows that quasi-reducibility of graph depends on DFS order of visiting nodes starting from root.
Since

- Rosen’s algorithm works on general graphs,
- each node in strongly connected graph can be root,

it can be used to compute cutset of graph $G(V,E)$ corresponding to off-diagonal part of irreducible matrix with zero-free diagonal.

*Greedy randomized adaptive search procedure* (GRASP) ([Festa, Pardalos & Resende’99](#)) is currently considered to be most effective algorithm for computing cutsets in graphs with large number of nodes:

- **gfvs** routine ([Festa, Pardalos & Resende’01](#)) in GRASP provides iterative algorithm, which returns cutset with smallest cardinality among all iterations as solution;
- similar to Rosen’s algorithm, **gfvs** routine of GRASP does not provide any guarantee on quality of computed cutset for general graphs.
Without loss of generality, let given \((n_A \times n_A)\) reducible matrix with zero-free diagonal be as in

\[
A = \begin{pmatrix}
    n_1 & n_2 & \cdots & n_K \\
n_1 & A_{11} & A_{12} & \cdots & A_{1K} \\
n_2 & A_{22} & \cdots & A_{2K} \\
    \vdots & \vdots & \ddots & \vdots \\
n_K & & & A_{KK}
\end{pmatrix},
\]

where \(A_{kk}\) is \((n_k \times n_k)\) irreducible submatrix with zero-free diagonal for \(k = 1, 2, \ldots, K\) and \(n_A = \sum_{k=1}^{K} n_k\).

Irreducible \(A\) is special case with \(K = 1\).
Permutation matrices $Q_{kk}$ can be computed such that

$$Q_{kk} A_{kk} Q_{kk}^T = \begin{pmatrix} n_{T_{kk}} & n_{C_{kk}} \\ n_{T_{kk}} & n_{C_{kk}} \end{pmatrix} \begin{pmatrix} T_{kk} & X_{kk} \\ Y_{kk} & C_{kk} \end{pmatrix},$$

where $n_k = n_{T_{kk}} + n_{C_{kk}}$ and $T_{kk}$ is $(n_{T_{kk}} \times n_{T_{kk}})$ triangular submatrix for $k = 1, 2, \ldots, K$.

For consistency assume $T_{kk}$ is upper-triangular.

Observe that for global index set $\{1, 2, \ldots, n_A\}$, $Q_{kk}$ is permutation matrix defined over indices in

$$\{1 + \sum_{j=1}^{k-1} n_j, 2 + \sum_{j=1}^{k-1} n_j, \ldots, \sum_{j=1}^{k} n_j\}.$$

Furthermore, let permutation vector $q_k$ corresponding to permutation matrix $Q_{kk}$ be partitioned into two subvectors as $q_k^T = (q_{T_{kk}}^T, q_{C_{kk}}^T)$.
Then permutation matrix $Q$ can be obtained such that

$$QAQ^T = \begin{pmatrix}
  n_{T_{11}} & n_{T_{22}} & \cdots & n_{T_{KK}} & n_{C_{11}} & n_{C_{22}} & \cdots & n_{C_{KK}} \\
  T_{11} & T_{12} & \cdots & T_{1K} & X_{11} & X_{12} & \cdots & X_{1K} \\
  T_{22} & \ddots & \ddots & \ddots & X_{22} & \ddots & \ddots & \ddots \\
  \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
  T_{KK} & \ddots & \ddots & \ddots & X_{KK} & & & & \\
  Y_{11} & Y_{12} & \cdots & Y_{1K} & C_{11} & C_{12} & \cdots & C_{1K} \\
  Y_{22} & \ddots & \ddots & \ddots & C_{22} & \ddots & \ddots & \ddots \\
  \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
  Y_{KK} & & \cdots & \ddots & Y_{KK} & & & & C_{KK}
\end{pmatrix}.$$  

Letting $n_T = \sum_{k=1}^{K} n_{T_{kk}}$ and $n_C = \sum_{k=1}^{K} n_{C_{kk}}$ yields $(2 \times 2)$ block partition. First diagonal block is upper-triangular and permutation vector on which $Q$ is based is

$$q^T = (q_{T_{11}}^T \ q_{T_{22}}^T \ \cdots \ q_{T_{KK}}^T \ q_{C_{11}}^T \ q_{C_{22}}^T \ \cdots \ q_{C_{KK}}^T).$$
Algorithm 1. Computes permutation for $(2 \times 2)$ block partition of $A$ in which first diagonal block is triangular.

**Precondition:** $A$ is in block triangular form with zero-free diagonal and order $n_A > 1$.

**Postcondition:** $q$ is permutation vector corresponding to permutation matrix $Q$, $n_T > 0$, $n_C > 0$, and $n_A = n_T + n_C$.

1. Compute cutset $C_k$ of graph $G(V_k, E_k)$ associated with off-diagonal part of $A_{kk}$ for $k = 1, 2, \ldots, K$.

2. Compute permutation vector $q_T^k = (q_{T_{kk}}^T, q_{C_{kk}}^T)$ that triangularizes submatrix of $A_{kk}$ associated with nodes in $T_k = V_k - C_k$ for $k = 1, 2, \ldots, K$.

3. Form permutation vector $q_T^T = (q_{T_{11}}^T, q_{T_{22}}^T, \ldots, q_{T_{KK}}^T, q_{C_{11}}^T, q_{C_{22}}^T, \ldots, q_{C_{KK}}^T)$, which has $n_T = \sum_{k=1}^{K} n_{T_{kk}}$ nodes in $T = \bigcup_{k=1}^{K} T_k$ at its beginning and $n_C = \sum_{k=1}^{K} n_{C_{kk}}$ nodes in $C = \bigcup_{k=1}^{K} C_k$ at its end.
Some results

Table 2: Characteristics of blocks in partition of $A$ obtained using Algorithm 1 for some irreducible MCs.

<table>
<thead>
<tr>
<th>Matrix</th>
<th>$n_A$</th>
<th>$nz_A$</th>
<th>sym</th>
<th>Partition</th>
<th>$n_T$</th>
<th>$nz_T$</th>
<th>$n_C$</th>
<th>$nz_C$</th>
<th>Time $p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>16,641</td>
<td>66,049</td>
<td>no</td>
<td>rosen</td>
<td>9,058</td>
<td>19,452</td>
<td>7,583</td>
<td>13,799</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>gfvs</td>
<td>10,863</td>
<td>26,648</td>
<td>5,778</td>
<td>6,501</td>
<td>35.4</td>
</tr>
<tr>
<td>easy</td>
<td>20,301</td>
<td>140,504</td>
<td>no</td>
<td>rosen</td>
<td>5,152</td>
<td>5,155</td>
<td>15,149</td>
<td>75,144</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>gfvs</td>
<td>5,588</td>
<td>5,984</td>
<td>14,713</td>
<td>70,985</td>
<td>544.4</td>
</tr>
<tr>
<td>telecom</td>
<td>20,491</td>
<td>101,041</td>
<td>no</td>
<td>rosen</td>
<td>10,246</td>
<td>10,246</td>
<td>10,245</td>
<td>10,305</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>gfvs</td>
<td>10,261</td>
<td>10,291</td>
<td>10,230</td>
<td>10,230</td>
<td>79.8</td>
</tr>
<tr>
<td>ncd</td>
<td>23,426</td>
<td>156,026</td>
<td>yes</td>
<td>rosen</td>
<td>11,841</td>
<td>11,841</td>
<td>11,585</td>
<td>13,983</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>gfvs</td>
<td>12,051</td>
<td>12,051</td>
<td>11,375</td>
<td>11,375</td>
<td>209.9</td>
</tr>
<tr>
<td>mutex</td>
<td>39,203</td>
<td>563,491</td>
<td>yes</td>
<td>rosen</td>
<td>22,819</td>
<td>22,819</td>
<td>16,384</td>
<td>16,384</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>gfvs</td>
<td>20,219</td>
<td>20,219</td>
<td>18,984</td>
<td>18,984</td>
<td>5,561.0</td>
</tr>
<tr>
<td>qn</td>
<td>104,625</td>
<td>593,115</td>
<td>no</td>
<td>rosen</td>
<td>53,655</td>
<td>159,288</td>
<td>50,970</td>
<td>147,693</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>gfvs</td>
<td>46,364</td>
<td>58,261</td>
<td>n/a</td>
<td>18,130.0</td>
<td></td>
</tr>
</tbody>
</table>

It is not possible to use gfvs for computing triangular blocks in large, sparse (irreducible) matrices.
Partitioning Google-like matrices

Objective is to obtain partition of \( B = \alpha P^T - I \) so that

- # of blocks, \( N \), is relatively low;
- it is relatively easy to solve diagonal blocks.

Let

\[ nb = \# \text{ of blocks returned by Tarjan’s algorithm for } B. \]

Then

- group \( (1 \times 1) \) blocks along diagonal into two blocks depending on shape of triangularity;
- triangular blocks obtained by computing cutsets for graphs associated with off-diagonal parts of remaining diagonal blocks can be solved through Sherman-Morrison formula.
### Two block partitionings

Consider partitioning *lower* obtained using cutsets, where $(1 \times 1)$ blocks with

- no off-diagonal row elements are grouped in $T_{0,0}$,
- some off-diagonal row elements are grouped in $C_{K+1,K+1}$

as in

\[
QBQ^T = \begin{pmatrix}
\begin{array}{cccc}
T_{0,0} & T_{1,0} & \cdots & T_{K,0} \\
T_{1,0} & T_{1,1} & & \\
\vdots & \vdots & \ddots & \\
T_{K,0} & T_{K,1} & \cdots & T_{K,K}
\end{array}
& \begin{array}{cccc}
X_{1,1} & \cdots & \cdots & \cdots \\
\vdots & \ddots & \ddots & \\
\vdots & \vdots & \ddots & \\
X_{K,1} & \cdots & \cdots & X_{K,K}
\end{array}

\begin{array}{cccc}
Y_{1,0} & Y_{1,1} & & \\
\vdots & \vdots & \ddots & \\
\vdots & \vdots & \vdots & \\
Y_{K,0} & Y_{K,1} & \cdots & Y_{K,K}
\end{array}
& \begin{array}{cccc}
C_{1,1} & \cdots & \cdots & \cdots \\
\vdots & \ddots & \ddots & \\
\vdots & \vdots & \ddots & \\
C_{K,1} & \cdots & \cdots & C_{K,K}
\end{array}

\begin{array}{cccc}
Y_{K+1,0} & Y_{K+1,1} & \cdots & Y_{K+1,K}
\end{array}
& \begin{array}{cccc}
C_{K+1,1} & \cdots & \cdots & \cdots \\
\vdots & \ddots & \ddots & \\
\vdots & \vdots & \ddots & \\
C_{K+1,K} & \cdots & \cdots & C_{K+1,K+1}
\end{array}
\end{pmatrix}
\]

so that \[nb = n_{T_{0,0}} + n_{C_{K+1,K+1}} + K\] and \[N = K + 1 \sum_{k=0}^{K} n_{T_{k,k}} + 1 n_{C_{K+1,K+1}} > 1.\]
Consider partitioning *upper* obtained using cutsets, where \((1 \times 1)\) blocks with

- no off-diagonal column elements are grouped in \(T_{0,0}\),
- some off-diagonal column elements are grouped in \(C_{K+1,K+1}\).

Let \(T\) be first aggregate diagonal block which is triangular.
### Some results

#### Stanford:

- Number of dangling nodes in $P$ = 20,315
- $nb = 29,914$
- Number of $(1 \times 1)$ blocks = 26,396

#### In 2004:

- Number of dangling nodes in $P$ = 86
- $nb = 367,675$
- Number of $(1 \times 1)$ blocks = 351,033

#### Table 3: Results for two web graphs with Tarjan (T) and Tarjan+Rosen (T+R).

<table>
<thead>
<tr>
<th>Matrix</th>
<th>$N$</th>
<th>Shape</th>
<th>Partition</th>
<th>$n_T$</th>
<th>$nz_T$</th>
<th>$\sum_k nz_{G_{k,k}}$</th>
<th>Time$_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford</td>
<td>3,520</td>
<td>lower</td>
<td>T</td>
<td>172</td>
<td>172</td>
<td>2,369,337</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T+R</td>
<td>159,037</td>
<td>452,823</td>
<td>715,600</td>
<td>1.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>upper</td>
<td>T</td>
<td>20,315</td>
<td>20,315</td>
<td>2,329,471</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T+R</td>
<td>179,180</td>
<td>517,281</td>
<td>675,734</td>
<td>1.1</td>
</tr>
<tr>
<td>In2004</td>
<td>16,644</td>
<td>lower</td>
<td>T</td>
<td>294,780</td>
<td>294,780</td>
<td>16,214,035</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T+R</td>
<td>929,778</td>
<td>2,573,755</td>
<td>8,089,436</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>upper</td>
<td>T</td>
<td>94</td>
<td>94</td>
<td>16,683,397</td>
<td>1.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T+R</td>
<td>635,092</td>
<td>1,745,311</td>
<td>8,558,798</td>
<td>1.9</td>
</tr>
</tbody>
</table>
Conclusion

- Order of triangular blocks obtained using cutsets are encouraging.
- Time to obtain cutsets is encouraging when $\alpha \rightarrow 1$. 