Graph Induced Rank Structured Matrices and their Representations

S. Chandrasekaran
UCSB
Email: shiv@ucsb.edu

N. Govindarajan
KU Leuven
Email: nithin.govindarajan@kuleuven.be

August 29, 2023

August 14, 2023

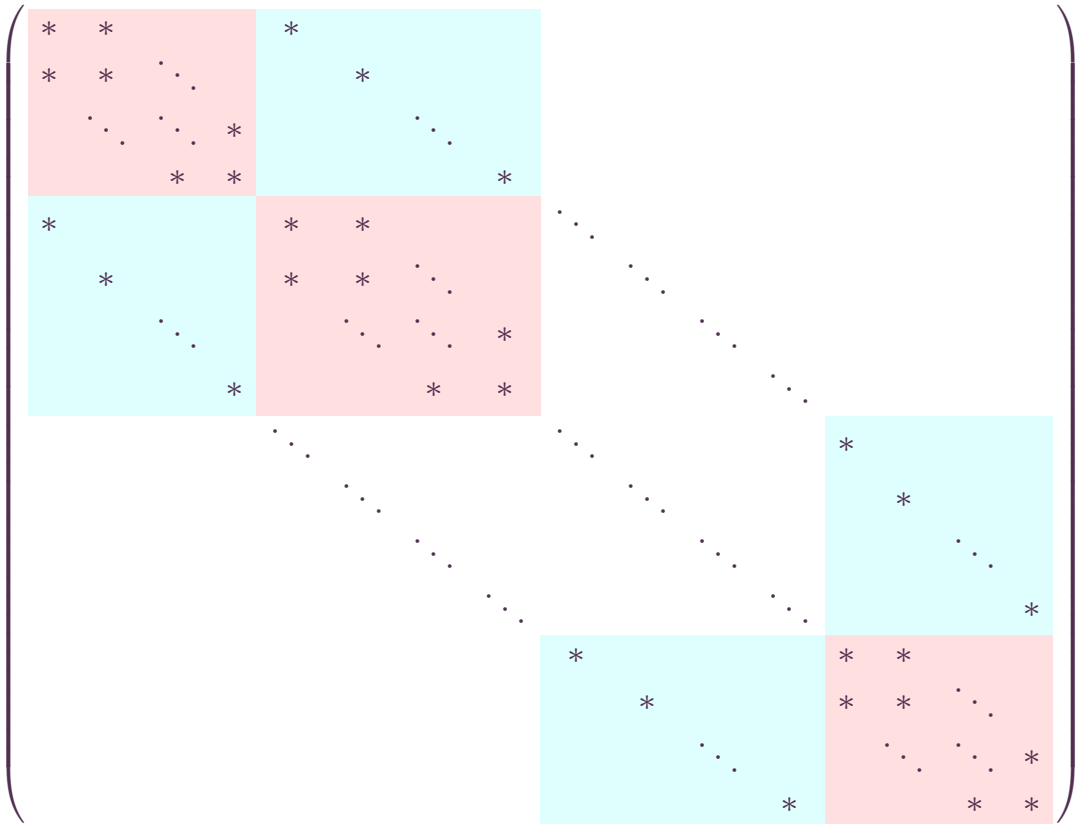
Computational and Applied Mathematics, Selva di Fasano

The problem

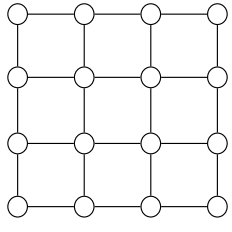
- What is the precise low-rank structure of the **inverse** of a sparse matrix?
- Is there a *good* representation of this inverse that allows **fast** matrix-vector multiplies?
  - Can this representation be computed rapidly given the entries of the inverse?
  - Can this representation be computed rapidly directly from the sparse matrix?
- SSS & HSS representations are being used to approximately capture fill-in during Gaussian elimination of sparse matrices, leading to practical pre-conditioners.
  - Are the resulting algorithms fast?
  - Are they the best we can do?
- A sparse matrix from a mesh graph in D dimensions of size $N \times N \times \cdots \times N$, requires $O\left(N^{\frac{D(1+D)}{2}}\right)$ operations during Gaussian elimination currently.
  - If $N \sim k$, where k is the wave number in a Helmholtz problem, then it has been conjectured that this is essentially the **lower** bound (modulo fast matrix multiplication speedups). [Personal communication from Yu Chen (Courant) attributing it to Coifman (Yale).]

Graphs of sparse matrices

Sparse matrix:



An associated graph:



Associated graphs

- Let $\mathbb{G}$ be an undirected graph with n nodes.
- Partition the matrix A:

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \ldots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n21} & \cdots & A_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

  allowing for **empty** row and column partitions.
- Associate each pair (x_i, b_i) with a unique node in $\mathbb{G}$.
- Some associations are more useful than others.
  - For the **inverse of a block sparse** matrix the block **adjacency graph** gives a good association.
  - For FMM-type matrices the signal flow graphs give a good association.
    - $\rightarrow$ For SSS these are line graphs.
    - $\rightarrow$ For HSS these are binary trees.

Hankel blocks

- For a square matrix A and permutations Π_1 and Π_2, let

$$\Pi_1 A \Pi_2 = \begin{pmatrix} \text{☎} & m & n-m \\ m & A_{11} & A_{12} \\ n-m & A_{21} & A_{22} \end{pmatrix}.$$

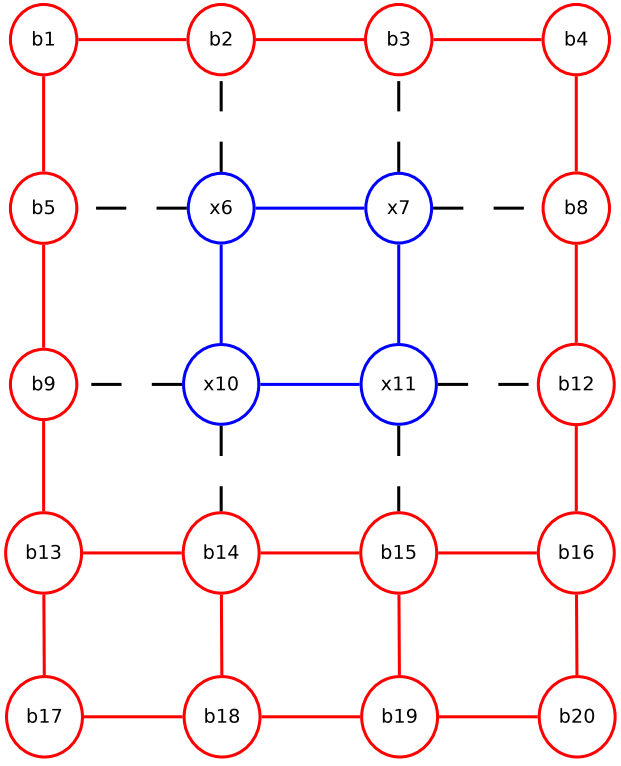
  We will refer to the off-diagonal blocks A_{12} and A_{21} as Hankel blocks.

- Note that not every sub-matrix of A can be a Hankel block: the sum of the number of rows and columns must be the size of A.

- It is well-known that the rank of a Hankel block of A^{-1} is at most the rank of the same Hankel block of A itself.

An induced sub-graph and its complement

- Let $\mathbb{H}$ be an **induced** sub-graph of the graph $\mathbb{G}$ associated to A. Then $\mathbb{H}$ automatically defines two Hankel blocks of A.
- Example of $\mathbb{G}$ with $\mathbb{H}$ nodes in blue and $\overline{\mathbb{H}}$ nodes in red:



- Notation: $x_{\mathbb{H}} = (\ x_6 \ \ x_7 \ \ x_{10} \ \ x_{11}\ )$ and $b_{\mathbb{H}} = (\ b_6 \ \ b_7 \ \ b_{10} \ \ b_{11}\ )$.

- We can choose block permutations Π_i such that

$$\Pi_1 A \Pi_2 = \begin{pmatrix} \text{☎} & x_{\overline{\mathbb{H}}} & x_{\mathbb{H}} \\ b_{\overline{\mathbb{H}}} & A_{\overline{\mathbb{H}},\overline{\mathbb{H}}} & A_{\overline{\mathbb{H}},\mathbb{H}} \\ b_{\mathbb{H}} & A_{\mathbb{H},\overline{\mathbb{H}}} & A_{\mathbb{H},\mathbb{H}} \end{pmatrix}.$$

- We call $A_{\overline{\mathbb{H}},\mathbb{H}}$ the Hankel block induced by $\mathbb{H}$.
- An edge in $\mathbb{G}$ is called a **border edge** if it connects a node in $\mathbb{H}$ to a node in $\overline{\mathbb{H}}$.
- The number of border edges induced by $\mathbb{H}$ is called the **border rank** of $\mathbb{H}$, and we denote it as $\rho(\mathbb{H})$.
- The pair $(A, \mathbb{G})$ is said to have a **graph induced rank structure** if there exists a *non-trivial* constant c such that

$$\operatorname{rank}(A_{\overline{\mathbb{H}},\mathbb{H}}) \leqslant c\rho(\mathbb{H})$$

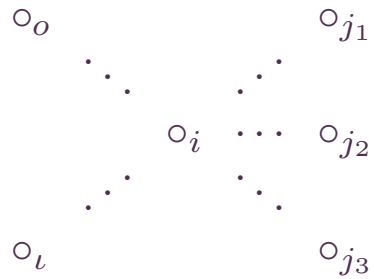
for **all** induced sub-graphs $\mathbb{H}$ of $\mathbb{G}$.

GIRS representation

- Produce an explicit (**inversion free**) representation of a GIRS matrix that can be chosen freely and yet satisfy the **rank** constraints of **all** Hankel blocks.
- This requires some fairly elaborate notation, so please bear with me.
  - The node of $\mathbb{G}$ associated with (x_i, b_i) is denoted merely as i. So the nodes of $\mathbb{G}$ are numbered from 1 to n.
  - A path in $\mathbb{G}$ from node k_1 to node k_l will be denoted simply as a string of numbers: $k_1\, k_2 \cdots k_l$. A **legal** path must be made up of a *finite* sequence of *distinct* adjacent nodes only. Other than that there are no constraints, and self-intersections are allowed.
  - To each path in $\mathbb{G}$ we are going to associate a matrix expression, but this requires some build up.
    - → To each of the n nodes of $\mathbb{G}$ we are going to associate a block matrix which we will call a **spinner** matrix. The spinner at node i will be denoted as W^i. Note that the superscript does not denote a matrix power and in general spinners are *not* square matrices.
    - → To each node i of $\mathbb{G}$ we add **two virtual** nodes simply denoted by the symbols ι (for input) and o (for output). These virtual nodes will only appear *inside* spinners and as *starting* and *ending* nodes of legal paths.

Spinners

- Suppose node i has 3 adjacent nodes: j_1, j_2, j_3:



  We also show the ι and o nodes for node i.

- Then the spinner for node i is partitioned as follows:

$$W^i = \begin{pmatrix} \text{☎} & j_1 & j_2 & j_3 & \iota \\ j_1 & w^i_{j_1,j_1} & w^i_{j_1,j_2} & w^i_{j_1,j_2} & w^i_{j_1,\iota} \\ j_2 & w^i_{j_2,j_1} & w^i_{j_2,j_2} & w^i_{j_2,j_3} & w^i_{j_2,\iota} \\ j_3 & w^i_{j_3,j_1} & w^i_{j_3,j_2} & w^i_{j_3,j_3} & w^i_{j_3,\iota} \\ o & w^i_{o,j_1} & w^i_{o,j_2} & w^i_{o,j_3} & w^i_{o,\iota} \end{pmatrix}$$

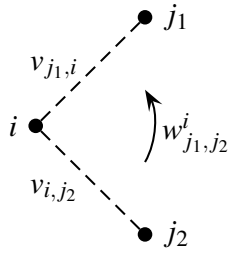
- The partition sizes will be described later.

Spinner components

- The block component $w^i_{j_1,j_2}$ for example of the spinner W^i will be referred to as an **edge-to-edge** operator.
  - If v_{i,j_2} is a column vector associated with the edge (i, j_2), then

$$v_{j_1,i} = w^i_{j_1,j_2} v_{i,j_2}$$

    is a column vector associated with the edge (j_1, i).



- Edge-to-edge operators of the form $w^i_{j,j}$ will be called **bounce-back** operators and appear on the block diagonal of W^i.
- The other block diagonal component of W^i is the edge-to-edge operator $w^i_{o,\iota}$. It is special as it involves the 2 virtual nodes at node i. It will actually turn out to be the block diagonal of the matrix $A_{i,i}$.

Path algebra

- To every path we associate a matrix expression built up from the spinner components along the path.

$$\begin{aligned} o\,i\,k_1\,k_2\cdots k_l\,j\,\iota &\equiv w^{i}_{o,k_1}\,w^{k_1}_{i,k_2}\,w^{k_2}_{k_1,k_3}\cdots w^{k_l}_{k_{l-1},j}\,w^{j}_{k_l,\iota} \\ k_1\,k_2\cdots k_l &\equiv w^{k_2}_{k_1,k_3}\,w^{k_3}_{k_2,k_4}\cdots w^{k_{l-1}}_{k_{l-2},k_l} \end{aligned}$$

  In fact we will freely use the path expressions on the left instead of the matrix expressions on the right. Note that the virtual nodes will always be those of the true nodes that are next to it.

- **Lemma**: For every legal path the matrix expression is legal.

- We will allow the concatenation of two paths if the last edge of the first path is the same as the first edge of the second path and associate that with matrix products:

$$o\,i\,k_1\,k_2\cdots k_{l-1}\,k_l \times k_{l-1}k_l\cdots k_p\,j \equiv w^{i}_{o,k_1}\,w^{k_1}_{i,k_2}\cdots w^{k_{l-1}}_{k_{l-2},k_l}w^{k_l}_{k_{l-1},k_{l+1}}\cdots w^{k_p}_{k_{p-1},j}.$$

- **Lemma**: Legal path products will yield legal matrix products.

Path sets

- Given two nodes i and j in $\mathbb{G}$ there can be multiple legals paths between them.
- We will need to make choices. Some common choices:
  - If $\mathbb{G}$ is a tree there is a unique legal non-self-intersecting path between 2 nodes.
    - $\rightarrow$ This choice will yield both **SSS** and **HSS** representations when they are applicable.
  - If $\mathbb{G}$ is a circle graph there are two distinct legal non-self-intersecting paths between 2 nodes.
    - $\rightarrow$ This will yield the circular semi-separable (**CSS**) representation, which has great theoretical interest.
  - For general graphs there are infinitely many legal paths between 2 nodes in general.
    - $\rightarrow$ This will yield the general **GIRS** representation.
- A set of legal paths between nodes i and j of $\mathbb{G}$ will be viewed as the sum of the corresponding matrix expressions:

$$\{oik_1\cdots k_l j\iota, oir_1\cdots r_m j\iota\} \equiv w^i_{o,k_1}\cdots w^j_{k_l,\iota} + w^i_{o,r_1}\cdots w^j_{r_m,\iota}.$$

- **Lemma**: All legal path sets for 2 distinct nodes will yield legal matrix additions.

A matrix representation

- We now have the notation to express a family of matrix representations that generalize both SSS and HSS to arbitrary graphs.
- Given the spinner components W^i for each node of $\mathbb{G}$ we can generate the block matrix entry $A_{i,j}$ as follows:
  - Choose a legal path set, P_{i_o,j_ι}, for the nodes i_o and j_ι.
  - Convert the path set into the corresponding matrix expression:

$$P_{i_o,j_\iota} = \bigcup_{oik_1\cdots k_l j\iota \in P_{i_o,j_\iota}} oik_1\cdots k_l j\iota \equiv \sum_{oik_1\cdots k_l j\iota \in P_{i_o,j_\iota}} w^i_{o,k_1} w^{k_1}_{i,k_2}\cdots w^j_{k_l,\iota} = A_{i,j}.$$

- The rank of the Hankel blocks will depend on the choice of path sets for each pair of nodes.
- If the sum is infinite there will be convergence issues. We ignore this for now and treat it purely formally.

Theorem 1. *If the path sets are chosen be the infinite set of all legal paths then the rank of the Hankel blocks will not exceed some constant times the associated border rank.*

- We will refer to the choice of the infinite all path sets as the **GIRS representation** of the matrix.

Corollary 2. *If $\mathbb{G}$ is a tree and we choose the path sets to be just the unique legal path between the two nodes, then the rank of the Hankel blocks will not exceed some constant times the associated border ranks.*

- The corollary falls out of the master theorem by just setting all the bounce-back operators to 0.
- We will refer to this representation for tree graphs as tree semi-separable (**TSS**) representation.
  - It includes both SSS and HSS as special cases.
  - It has a fast construction algorithm from the matrix entries, just like SSS and HSS.
  - It has fast matrix-vector multiply and fast solvers, just like SSS and HSS.
  - It is a strict generalization of SSS and HSS (better compression in general).

Prefix and suffix path sets

- We need some extra notation to describe the proof of the master theorem.
- P_{i_o,j_ι} will denote the set of all legal paths that start at the o node of node i and end at the ι node of node j, with all intermediate nodes being distinct adjacent nodes in $\mathbb{G}$.
- $P_{i_o,[k_1,k_2]}$ will denote the set of all legal paths that start at the o node of node i and end with the last two nodes being k_1 and k_2, with both of those being distinct adjacent nodes in $\mathbb{G}$, with **no** other occurrences of the edge $[k_1, k_2]$.
- $P_{[k_1,k_2],j_o}$ will denote the set of all legal paths with the first two nodes being k_1 and k_2, with both of those being distinct adjacent nodes in $\mathbb{G}$, and ending at node ι of node j in $\mathbb{G}$, with **no** other occurrences of the edge $[k_1, k_2]$.
- $P_{[k_1,k_2],[k_3,k_4]}$ will denote the set of all legal paths that start with the edge $[k_1, k_2]$ and end with the edge $[k_3, k_4]$. Both of those edges are allowed to be *repeated* inside the path.
- We denote the product of two path sets $P_{i_o,[k_1,k_2]}$ and $P_{[k_1,k_2],j_\iota}$, as $P_{i_o,[k_1,k_2]}\,P_{[k_1,k_2],j_\iota}$, and define it as the path set obtained by taking the cross product of $P_{i_o,[k_1,k_2]}$ and $P_{[k_1,k_2],j_\iota}$.
  - Therefore $P_{i_o,[k_1,k_2]}\,P_{[k_1,k_2],[k_3,k_4]}\,P_{[k_3,k_4],j_\iota}$ is a path set containing all possible products of paths with the first path being chosen from $P_{i_o,[k_1,k_2]}$, the second path from $P_{[k_1,k_2],[k_3,k_4]}$, and the third path from $P_{[k_3,k_4],j_\iota}$.

Path factorization

- **Lemma**: All matrix expressions in the product $P_{i_o,[k_1,k_2]}\, P_{[k_1,k_2],[k_3,k_4]}\, P_{[k_1,k_2],j_\iota}$ are legal.
- **Lemma**: $\bigcup_{[k_1,k_2],[k_3,k_4]\in E(\mathbb{G})} P_{i_o,[k_1,k_2]}\, P_{[k_1,k_2],[k_3,k_4]}\, P_{[k_1,k_2],j_\iota} = P_{i_o,j_\iota}$.
  - $E(\mathbb{G})$ denotes the set of all edges in $\mathbb{G}$.
  - This lemma is also true as a matrix expression (factorization).
    - → The difficulty in the proof is that set cross-products do not count multiple occurences of the same pairs, whereas block matrix multiplication does (via scalar multiplication),
    - → We circumvent this difficulty by using longest prefixes in $P_{i_o,[k_1,k_2]}$ that are free of the edge $[k_1,k_2]$, and longest suffixes in $P_{[k_3,k_4],j_\iota}$ that are free of the edge $[k_3,k_4]$.
    - → Without infinite path sets this factorization may be false.

Low rank path factorizations

- Consider a Hankel block using the all-path sets:

$$A_{\overline{\mathbb{H}},\mathbb{H}} \equiv \begin{pmatrix} \text{☎} & 4 & 5 \\ 1 & P_{1_o,4_\iota} & P_{1_o,5_\iota} \\ 2 & P_{2_o,4_\iota} & P_{2_o,5_\iota} \\ 3 & P_{3_o,4_\iota} & P_{3_o,5_\iota} \end{pmatrix}.$$

- Let $[1,4]$ and $[2,5]$ be the border edges of $\mathbb{H}$.
- Then we have the low rank path factorization for $A_{\overline{\mathbb{H}},\mathbb{H}}$:

$$\begin{pmatrix} \text{☎} & [1,4] & [2,5] \\ 1 & P_{1_o,[1,4]} & P_{1_o,[2,5]} \\ 2 & P_{2_o,[1,4]} & P_{2_o,[2,5]} \\ 3 & P_{3_o,[1,4]} & P_{3_o,[2,5]} \end{pmatrix} \begin{pmatrix} \text{☎} & [1,4] & [2,5] \\ [1,4] & P_{[1,4],[1,4]} & P_{[1,4],[2,5]} \\ [2,5] & P_{[2,5],[1,4]} & P_{[2,5],[2,5]} \end{pmatrix} \begin{pmatrix} \text{☎} & 4 & 5 \\ [1,4] & P_{[1,4],4_\iota} & P_{[1,4],5_\iota} \\ [2,5] & P_{[2,5],4_\iota} & P_{[2,5],5_\iota} \end{pmatrix}.$$

- This is essentially the proof of the master theorem.
  - All the notation was just to write this formula compactly and transparently.

Example: SSS

- $$A = \begin{pmatrix} D_1 & U_1 V_2 & U_1 W_2 V_3 \\ P_2 Q_1 & D_2 & U_2 V_3 \\ P_3 R_2 Q_1 & P_3 Q_2 & D_3 \end{pmatrix}$$
- $$\mathbb{G} = \circ_1 \; - \; \circ_2 \; - \; \circ_3$$
- $$W^1 = \begin{pmatrix} \text{☎} & 2 & \iota \\ 2 & 0 & Q_1 \\ o & U_1 & D_1 \end{pmatrix}$$
$$W^2 = \begin{pmatrix} \text{☎} & 1 & 3 & \iota \\ 1 & 0 & W_2 & V_2 \\ 3 & R_2 & 0 & Q_2 \\ o & P_2 & U_2 & D_2 \end{pmatrix}$$
$$W^3 = \begin{pmatrix} \text{☎} & 2 & \iota \\ 2 & 0 & V_3 \\ o & P_3 & D_3 \end{pmatrix}$$

Example: HSS

$$\mathbb{G} = \begin{array}{ccccc} \circ_1 & - & \circ_4 & - & \circ_5 \\ & & | & & | \\ & & \circ_2 & & \circ_3 \end{array}$$

$$A = \begin{pmatrix} \text{☎} & 1 & 2 & 3 \\ 1 & D_{2;1} & U_{2;1}B_{2;1,2}V_{2;2} & U_{2;1}R_{2;1}B_{1;1,2}V_{1;2} \\ 2 & U_{2;2}B_{2;2,1}V_{2;1} & D_{2;2} & U_{2;2}R_{2;2}\,B_{1;1,2}V_{1;2} \\ 3 & U_{1;2}\,B_{1;2,1}\,W_{2;1}\,V_{2;1} & U_{1;2}\,B_{1;2,1}\,W_{2;2}\,V_{2;2} & D_{1;2} \end{pmatrix}$$

$$W^1 = \begin{pmatrix} \text{☎} & 4 & \iota \\ 4 & 0 & V_{2;1} \\ o & U_{2;1} & D_{2;1} \end{pmatrix} \quad W^2 = \begin{pmatrix} \text{☎} & 4 & \iota \\ 4 & 0 & V_{2;2} \\ o & U_{2;2} & D_{2;2} \end{pmatrix} \quad W^3 = \begin{pmatrix} \text{☎} & 5 & \iota \\ 5 & 0 & V_{1;2} \\ o & U_{1;2} & D_{1;2} \end{pmatrix}$$

$$W^4 = \begin{pmatrix} \text{☎} & 1 & 2 & 5 & \iota \\ 1 & 0 & B_{2;1,2} & R_{2;1} & \cdot \\ 2 & B_{2;2,1} & 0 & R_{2;2} & \cdot \\ 5 & W_{2;1} & W_{2;2} & 0 & \cdot \\ o & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \quad W^5 = \begin{pmatrix} \text{☎} & 3 & 4 & \iota \\ 3 & 0 & B_{1;2,1} & \cdot \\ 4 & B_{1;1,2} & 0 & \cdot \\ o & \cdot & \cdot & \cdot \end{pmatrix}$$

Example: TSS

$$\mathbb{G} = \begin{array}{ccccc} \circ_1 & - & \circ_4 & - & \circ_3 \\ & & | & & \\ & & \circ_2 & & \end{array}$$

$$A = \begin{pmatrix} \text{☎} & 1 & 2 & 3 & 4 \\ 1 & w^1_{o,\iota} & w^1_{o,4}w^4_{1,2}w^2_{4,\iota} & w^1_{o,4}w^4_{1,3}w^3_{4,\iota} & w^1_{o,4}w^4_{1,\iota} \\ 2 & w^2_{o,4}w^4_{2,1}w^1_{4,\iota} & w^2_{o,\iota} & w^2_{o,4}w^4_{2,3}w^3_{4,\iota} & w^2_{o,4}w^4_{1,\iota} \\ 3 & w^3_{o,4}w^4_{3,1}w^1_{4,\iota} & w^3_{o,4}w^4_{3,2}w^2_{4,\iota} & w^3_{o,\iota} & w^3_{o,4}w^4_{1,\iota} \\ 4 & w^4_{o,1}w^1_{4,\iota} & w^4_{o,2}w^2_{4,\iota} & w^4_{o,3}w^3_{4,\iota} & w^4_{o,\iota} \end{pmatrix}$$

$$W^{j<4} = \begin{pmatrix} \text{☎} & 4 & \iota \\ 4 & 0 & w^j_{4,\iota} \\ o & w^j_{o,4} & w^1_{o,\iota} \end{pmatrix} \quad W^4 = \begin{pmatrix} \text{☎} & 1 & 2 & 3 & \iota \\ 1 & 0 & w^4_{1,2} & w^4_{1,3} & w^4_{1,\iota} \\ 2 & w^4_{2,1} & 0 & w^4_{2,3} & w^4_{2,\iota} \\ 3 & w^4_{3,1} & w^4_{3,2} & 0 & w^4_{3,\iota} \\ o & w^4_{o,1} & w^4_{o,2} & w^4_{o,3} & w^4_{o,\iota} \end{pmatrix}$$

Construction of GIRS

- Freely picking a GIRS representation (modulo norm constraints for convergence) will yield a matrix with GIRS structure.
- The converse is complicated:
  - For **any sparse matrix** the minimal GIRS representation is **trivial** to write down.
  - For any matrix a non-minimal GIRS representation is trivial via TSS using a tree cover.
  - Finding a **minimal** GIRS representation seems very difficult in general.

    Question 1. Does this exclude a general purpose fast exact sparse solver and support the Coifman conjecture?

  - Settle for approximate-minimal GIRS representations?

    $\rightarrow$ We will consider the case of a simple loop graph later.

Fast matrix-matrix multiply

- For the infinite GIRS representation fast matrix-vector multiplications seem to require **approximations**.
- Surprisingly it is possible to give an **exact** fast algorithm for $A = BC$.

$$w^k_{i,j}(BC) = \begin{pmatrix} \text{☎} & j(B) & j(C) \\ i(B) & w^k_{i,j}(B) & w^k_{i,\iota}(B)\, w^k_{o,j}(C) \\ i(C) & 0 & w^k_{i,j}(C) \end{pmatrix}$$

$$w^k_{i,\iota}(BC) = \begin{pmatrix} \text{☎} & \iota(C) \\ i(B) & w^{v_k}_{v_i,\iota}(B)\, w^{v_k}_{o,\iota}(C) \\ i(C) & w^{v_k}_{v_i,\iota}(C) \end{pmatrix}$$

$$w^k_{o,j}(BC) = \begin{pmatrix} \text{☎} & B & C \\ o(B) & w^k_{o,j}(B) & w^k_{o,\iota}(B) w^k_{o,j}(C) \end{pmatrix}.$$

  - Proof using infinite path sets.
- Fast addition is easy.

Circular Semi-Separable representation

- Given a dense matrix and a non-tree $\mathbb{G}$ can we construct a minimal GIRS representation?
- 2-path CSS is a remarkable simpler model problem to study first:

$$\mathbb{G} = \begin{array}{ccc} \circ_0 & - & \circ_1 \\ | & & | \\ \circ_3 & - & \circ_2 \end{array}$$

$$A \equiv \left(\begin{array}{c|cccc} \text{☎} & 0 & 1 & 2 & 3 \\ \hline 0 & o\iota & \begin{array}{c}01\\0321\end{array} & \begin{array}{c}012\\032\end{array} & \begin{array}{c}0123\\03\end{array} \\ 1 & \begin{array}{c}10\\1230\end{array} & o\iota & \begin{array}{c}12\\1032\end{array} & \begin{array}{c}123\\103\end{array} \\ 2 & \begin{array}{c}210\\230\end{array} & \begin{array}{c}21\\2301\end{array} & o\iota & \begin{array}{c}23\\2103\end{array} \\ 3 & \begin{array}{c}30\\3210\end{array} & \begin{array}{c}301\\321\end{array} & \begin{array}{c}32\\3012\end{array} & o\iota \end{array} \right)$$

Theorem 3. *CSS satisfies all graph-induced rank structure constraints.*

Periodic Tridiagonal

- CSS may be easier to construct than the infinte-path GIRS representation.
- CSS is closely tied to the inverse of a **periodic tridiagonal** matrix.
- The inverse of a **periodic bidiagonal** matrix:

$$\begin{pmatrix} I & -A_0 & & \\ & I & -A_1 & \\ & & I & -A_2 \\ -A_3 & & & I \end{pmatrix} \begin{pmatrix} I & A_0 & A_0A_1 & A_0A_1A_2 \\ A_1A_2A_3 & I & A_1 & A_1A_2 \\ A_2A_3 & A_2A_3A_0 & I & A_2 \\ A_3 & A_3A_0 & A_3A_0A_1 & I \end{pmatrix} =$$

$$\begin{pmatrix} I - A_0A_1A_2A_3 & & & \\ & I - A_1A_2A_3A_0 & & \\ & & I - A_2A_3A_0A_1 & \\ & & & I - A_3A_0A_1A_2 \end{pmatrix}$$

Lemma 4.

$$\begin{pmatrix} D_0 & A_0 & & B_3 \\ B_0 & D_1 & A_1 & \\ & B_1 & D_2 & A_2 \\ A_3 & & B_2 & D_3 \end{pmatrix} = \begin{pmatrix} I & & & L_3 \\ L_0 & I & & \\ & L_1 & I & \\ & & L_2 & I \end{pmatrix} \begin{pmatrix} U_0 & A_0 & & \\ & U_1 & A_1 & \\ & & U_2 & A_2 \\ A_3 & & & U_3 \end{pmatrix}$$

exists under the diagonally dominant condition $\|B_i D_i^{-1}\| + \|A_{i-1} D_i^{-1}\| \leqslant 1$.

- The proof is non-trivial since Gaussian elimination is of no use here.
- We depend on a fixed-point argument to establish the existence.
- Therefore we see that

$$(\text{Per. tri-diag})^{-1} = (\text{Per. upp. bi-diag})^{-1} + (\text{Per. low. bi-diag})^{-1}$$

is a reasonable conjecture. However the product leads to a non-square multilinear tensor equation in a **fixed** matrix size that seems difficult to handle.

Conclusions

Summary:

- GIRS: an efficient representation of sparse matrices and their inverse.
- TSS: a representation for fast exact algorithms (subsumes both SSS & HSS).
- Construction of GIRS for arbitrary graphs is hard.
  - Approximate constructions and fast algorithms have to be investigated

Future work:

- Apply GIRS to fast incomplete LU factorization of sparse matrices.

Acknowledgements:

- Patrick Dewilde
- Ethan Epperly
- Ming Gu

girsTssTalk.tm,v 1.1 2023/08/25 15:38:02 00shi Exp 00shi $