Sparse matrices and graphs (Basics)

Grégoire Pichon, Bora Uçar & Frédéric Vivien

CNRS, INRIA, Université Lyon 1 & ENS Lyon

CR15: November 2022

gpichon.gitlabpages.inria.fr/m2if-numerical_algorithms/

Outline

CR15: What is and what is not?

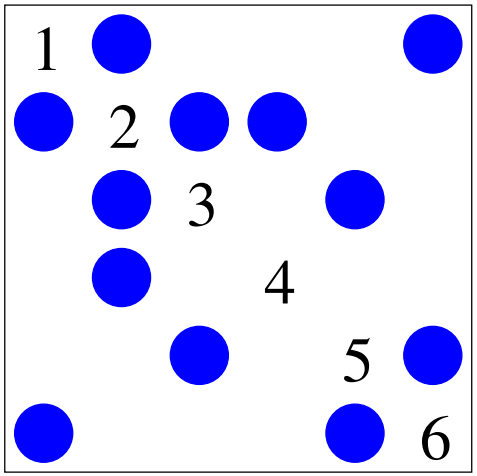
- CR15 is about sparse matrices and computations with them.

  Sparse matrix: It is a matrix with many zeros which are not stored.

  Sparse matrix computations: Mostly operate on the nonzero elements.

  Sparse matrices are everywhere: Google's PageRank, network analysis (social, biological,...), simulation science (circuits, numerical,...).

$$\mathbf{A} = \begin{bmatrix} 4 & -1 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & -1 & 3 & 0 & -1 & 0 \\ 0 & -1 & 0 & 2 & 0 & 0 \\ 0 & 0 & -1 & 0 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$



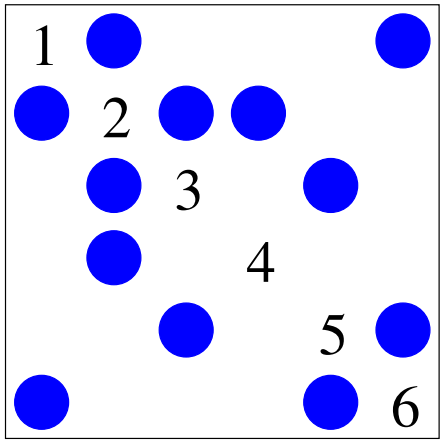
CR15: What is and what is not?

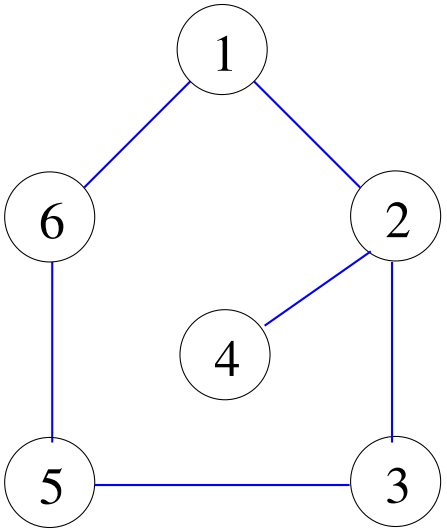
- CR15 is about graphs.

  Nonzeros of a matrix can been seen as the edges of a bipartite graph, a directed graph, an undirected graph, or a hypergraph.

- Computations on sparse matrices can be modeled using graphs.

  CR15 is about this correspondence in a broader sense.





CR15: What is and what is not?

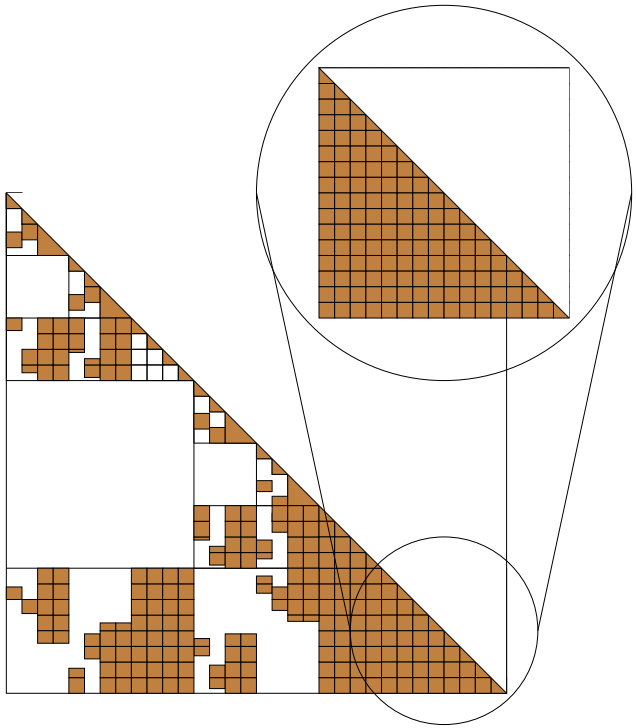
- ...not a scientific computing/linear algebra course.
- ...not a full-fledged combinatorial optimization course.
- ...not a parallel computing course.

We will see some exciting problems having to do with topics from these three domains.

Prerequisites: Graphs and algorithms. A little bit of linear algebra for comfort and seing beyond what we discuss (but not required)

CR15: Sparse direct solvers

- Solve large systems (> 1M unknowns)
- On top of parallel architectures

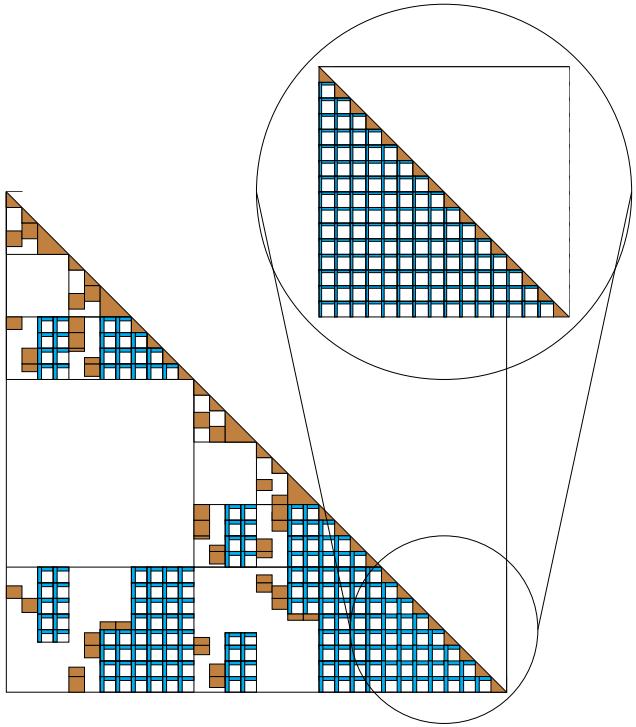


- Work on the structure of the factorized matrix
- Efficiency related to block's granularity
- Complexity depending on the underlying graph properties
- Implementation on modern hardware (MPI, GPUs...)

We will investigate how to build efficient generic sparse direct solvers.

CR15: Sparse direct solvers

- Low-rank compression
- Recent approach ($<$ 15 years)
- Reduce the complexities



- Compress a block C in the form $u_C v_C^t$ such that $\frac{||C - u_C v_C^t||}{||C||} \leq \tau$
- Different compression schemes ($\mathcal{H}$ matrices, BLR)
- Potential compression depends on the underlying operator

We will investigate how the use of low-rank compression impacts sparse direct solvers (basic operations, complexity).

CR 15: Scheduling under memory constraints

Underlying question: how can we optimize memory usage?

- What is the minimum memory needed to perform some processing?
  - The pebble game model
  - Lower bounds
- Under memory shortage: what is the minimum amount of I/Os required to perform the processing?
  - I/O complexity
- Communication-avoiding algorithms
- Memory-aware DAG scheduling
  - Space-time tradeoffs

A question before we go on

[Types of questions that we will have in homework/self study]

(Class:)
Let A and B be two lower triangular matrices.
Show that $C = AB$ is lower triangular.

CR15: A sample problem

- Let A be an $n \times n$ symmetric positive definite matrix
- The LL^T factorization can be described by the equation:

$$A = H_0 = \begin{pmatrix} d_1 & v_1^T \\ v_1 & \overline{H_1} \end{pmatrix}$$

$$= \begin{pmatrix} \sqrt{d_1} & 0 \\ \frac{v_1}{\sqrt{d_1}} & I_{n-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & H_1 \end{pmatrix} \begin{pmatrix} \sqrt{d_1} & \frac{v_1^T}{\sqrt{d_1}} \\ 0 & I_{n-1} \end{pmatrix}$$

$= L_1 A_1 L_1^T$, where

$$H_1 = \overline{H_1} - \frac{v_1 v_1^T}{d_1}$$

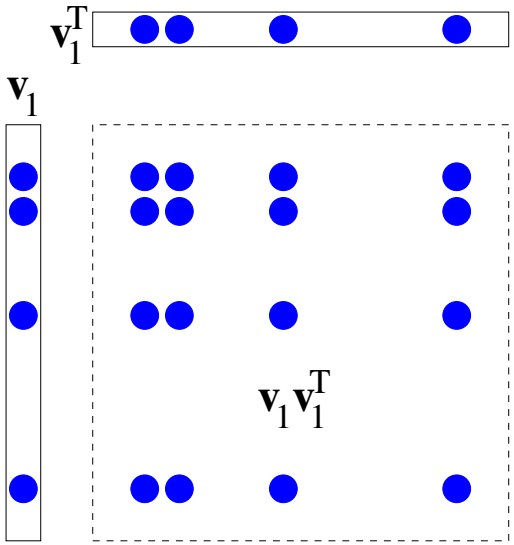
- The basic step is applied on $H_1, H_2, \ldots$ to obtain :

$$A = (L_1 L_2 \cdots L_{n-1})\, I_n\, (L_{n-1}^T \cdots L_2^T L_1^T) \quad = \quad LL^T$$

The basic step: $H_1 = \overline{H_1} - \frac{v_1 v_1^T}{d_1}$

Consider the graph of A.

What is $v_1 v_1^T$ in this graph (or in terms of structure)?



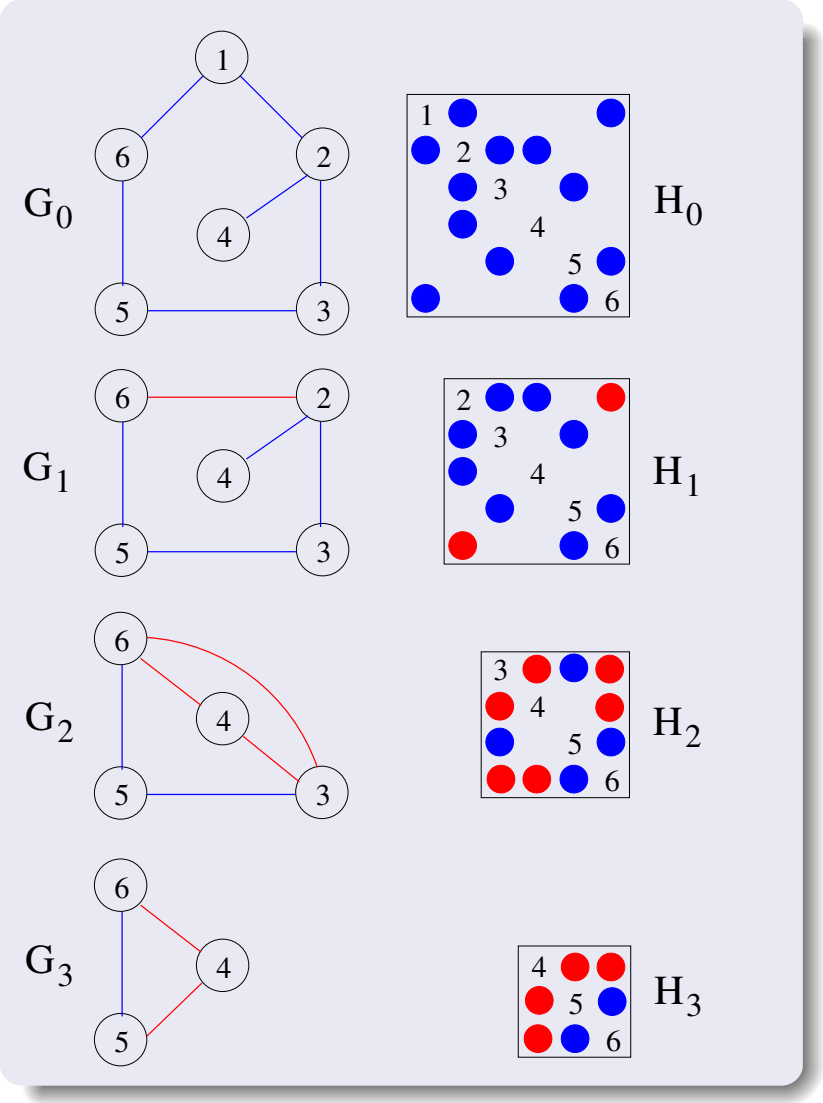
$v_1 v_1^T$: a dense sub-matrix in H_1.

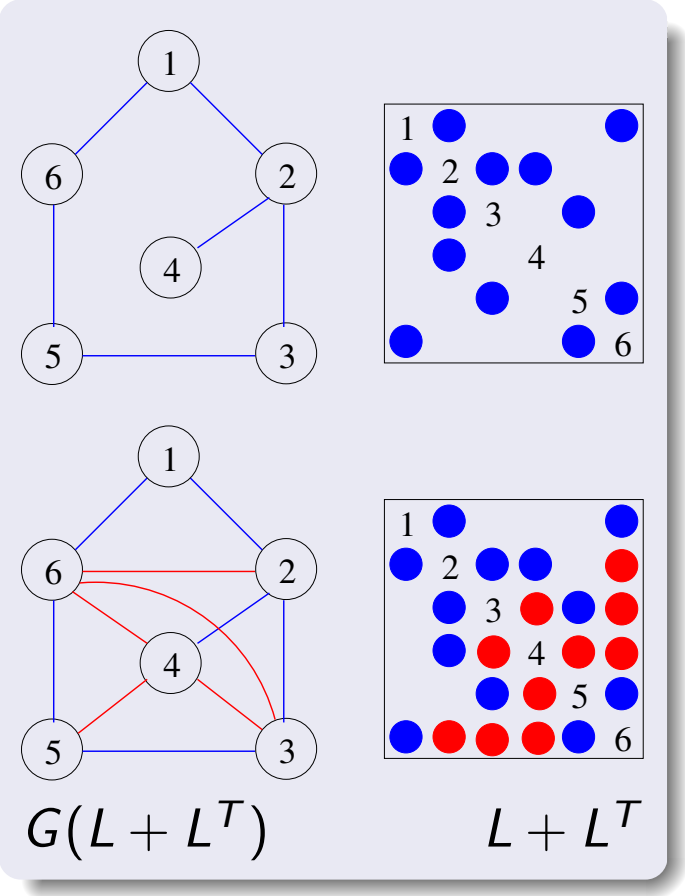
Recall $A = \begin{pmatrix} d_1 & v_1^T \\ v_1 & \overline{H_1} \end{pmatrix}$; that is v_1 is a column of A, hence the neighbors of the corresponding vertex.

The neighbors of the corresponding vertex form a clique.

Basic step modeled using a graph: Pick the vertex, form a clique from its neighbors, and remove the vertex.

CR15: A sample problem





What can we say about the final matrix by just looking at the original graph?

CR15: Planning

Instructors: Lecture notes (slides) and problems.

Students: Participation to solving problems;

Evaluation: homework (short questions), participation, and (projects/presentations). We will decide (projects/presentations) according to the number of participants and/or requests.

Dates: Nov: 14, 17, 21, 24, 28; Dec: 1, 5, 8, 12, 15; Jan: 2, 5, 9, 12, 16, 19, 23, 27.

Offices: M7.208, M7.215: Bora, Frédéric, Grégoire

Contact: `gregoire.pichon@ens-lyon.fr`
`bora.ucar@ens-lyon.fr`
`frederic.vivien@inria.fr`

Course web: https://gpichon.gitlabpages.inria.fr/m2if-numerical_algorithms/ Slides will be posted (more likely just before the class).

Outline

Sparse matrices

Harry Markowitz: "I got a Nobel Prize..."
Nobel in 1990 (work '52).

When we started putting these models together they became very large as compared to linear programming capability at that time... It struck me that our matrices were mostly full of zeros, and if you have a set of simultaneous equations that are mostly zeros, if you pick your pivots right, you could just solve it by hand. Then I thought, well, maybe we could get the computer to do the same thing. This led to "Sparse Matrices." As far as I know, I coined the word "Sparse Matrix."

I published [in this area, refers to a 1957 paper on selecting pivots for Gaussian elimination] and then forgot about it...[In a] decade, it started to catch on and expanded very fast. I was completely oblivious to what was going on and to what extent.

Sparse matrices: Coordinate format

There are many ways to store a sparse matrix.

We will look at three standard representations which store only the nonzero entries. We use τ to denote the number of nnzs.

$$\begin{bmatrix} 1.1 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 2.2 & 0.0 & 2.4 & 0.0 \\ 3.1 & 0.0 & 3.3 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 4.4 & 0.0 \\ 0.0 & 5.2 & 0.0 & 5.4 & 5.5 \end{bmatrix}$$

Coordinate (COO) format
MatrixMarket format

Two integer arrays (irn, jcn) and a double array `A`, each of size τ:

```
irn = [  1   2   2   3   3   4   5   5   5 ]

jcn = [  1   2   4   1   3   4   2   4   5 ]

A   = [1.1 2.2 2.4 3.1 3.3 4.4 5.2 5.4 5.5 ]
```

The kth entry a_{ij} is stored as `irn[k]` $= i$, `jcn[k]` $= j$, `a[k]` $= a_{ij}$.

The storage is 2τ integers and τ doubles (or single or complex). In general, $\tau = \mathcal{O}(m + n)$.

Remark: Nonzeros can be stored in any order.

Sparse matrices: Coordinate format

Question: Let A be an $m \times n$ matrix. R and C be two diagonal matrices of size $m \times m$ and $n \times n$. A can be scaled as $\hat{A} = RAC$.

Write a function/routine that does this operation for matrices stored in the coordinate format. R and C are stored as arrays $r[1, \ldots, m]$ and $c[1, \ldots, n]$. Create new arrays for $\hat{A}$.

Sparse matrices: Compressed storage by rows

There are many ways to store a sparse matrix.

$$\begin{bmatrix} 1.1 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 2.2 & 0.0 & 2.4 & 0.0 \\ 3.1 & 0.0 & 3.3 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 4.4 & 0.0 \\ 0.0 & 5.2 & 0.0 & 5.4 & 5.5 \end{bmatrix}$$

Compressed storage by rows (CSR)

Two integer arrays (ia, jcn) and a double array `A`:

```
ia  = [  1   2       4       6   7        10 ]

jcn = [  1   2   4   1   3   4   2   4   5 ]

A   = [1.1 2.2 2.4 3.1 3.3 4.4 5.2 5.4 5.5 ]
```

The nonzeros of the ith row are stored at the `ia[i]`$\cdots$ `ia[i+1]-1` positions of `jcn` and `A`.

For example the 3rd row: starts at `ia[3] = 4` and finishes at `ia[3+1]-1 = 5`. The column indices are therefore `jcn[4,5] = 1 3` and values are `A[4,5] = 3.1 3.3`.

Sparse matrices: Compressed storage by rows

There are many ways to store a sparse matrix.

$$\begin{bmatrix} 1.1 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 2.2 & 0.0 & 2.4 & 0.0 \\ 3.1 & 0.0 & 3.3 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 4.4 & 0.0 \\ 0.0 & 5.2 & 0.0 & 5.4 & 5.5 \end{bmatrix}$$

Compressed storage by rows (CSR)

Two integer arrays (ia, jcn) and a double array `A`:

```
ia  = [  1   2       4       6   7        10 ]

jcn = [  1   2   4   1   3   4   2   4   5 ]

A   = [1.1 2.2 2.4 3.1 3.3 4.4 5.2 5.4 5.5 ]
```

The nonzeros of the ith row are stored at the `ia[i]`$\cdots$`ia[i+1]-1` positions of `jcn` and `A`.

Let matrix be of size $m \times n$, and τ be the number of nonzeros, then the storage is $m + 1 + \tau$ integer and τ double (or single or complex).

Remark: Nonzeros in a row can be stored in any order.

Sparse matrices: Compressed storage by rows

Question: Let A be an $m \times n$ matrix stored in the CSR format. What is the number of nonzeros?

Question : Let A be an $m \times n$ matrix stored in the CSR format. Write a function/routine that displays A.

Let A be an $m \times n$ matrix. R and C be two diagonal matrices of size $m \times m$ and $n \times n$. A can be scaled as $\hat{A} = RAC$.

Question : Write a function/routine that does this operation for matrices stored in the CSR format.

Sparse matrices: Compressed storage by columns

There are many ways to store a sparse matrix.

$$\begin{bmatrix} 1.1 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 2.2 & 0.0 & 2.4 & 0.0 \\ 3.1 & 0.0 & 3.3 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 4.4 & 0.0 \\ 0.0 & 5.2 & 0.0 & 5.4 & 5.5 \end{bmatrix}$$

Compressed storage by columns (CSC)

Two integer arrays (`irn, ja`) and a double array `A`:

```
ja  = [  1       3       5   6           9  10 ]

irn = [  1   3   2   5   3   2   4   5   5 ]

A   = [1.1 3.1 2.2 5.2 3.3 2.4 4.4 5.4 5.5 ]
```

The nonzeros of the jth column are stored at the `ja[j]`$\cdots$`ja[j+1]-1` positions of `irn` and `A`.

For example the 2nd col: starts at `ja[2] = 3` and finishes at `ja[2+1]-1 = 4`. The row indices are therefore `irn[3,4] = 2 5` and values are `A[3,4] = 2.2 5.2`.

Sparse matrices: Compressed storage by columns

There are many ways to store a sparse matrix.

$$\begin{bmatrix} 1.1 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 2.2 & 0.0 & 2.4 & 0.0 \\ 3.1 & 0.0 & 3.3 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 4.4 & 0.0 \\ 0.0 & 5.2 & 0.0 & 5.4 & 5.5 \end{bmatrix}$$

Compressed storage by columns

Two integer arrays (`irn, ja`) and a double array `A`:

```
ja  = [  1       3       5   6           9  10 ]

irn = [  1   3   2   5   3   2   4   5   5 ]

A   = [1.1 3.1 2.2 5.2 3.3 2.4 4.4 5.4 5.5 ]
```

The nonzeros of the jth column are stored at the `ja[j]`$\cdots$`ja[j+1]-1` positions of `irn` and `A`.

Let matrix be of size $m \times n$, and τ be the number of nonzeros, then the storage is $n + 1 + \tau$ integer and τ double (or single or complex).

Remark: Nonzeros in a column can be stored in any order.

Sparse matrices: Compressed storage by columns

Question: Let A be an $m \times n$ matrix stored in the compressed storage by columns format. What is the number of nonzeros?

Question: (HW1-Q1 for 24 nov) Let A be an $m \times n$ matrix stored in the coordinate format, with τ nonzeros (assume that m, n and τ are given). Design an algorithm that creates A in csc format (`ja`, `irn`, and `A`). What is the run time? We expect a pseudo-code.

Reminder: Dense matrix vector multiplication

Need to compute $y \leftarrow Ax$ for an $m \times n$ dense matrix A and suitable dense vectors y and x.

Row-major order

for $i = 1$ **to** m **do**
  $y[i] \leftarrow 0.0$
  for $j = 1$ **to** n **do**
    $y[i] \leftarrow y[i] + A[i,j] \times x[j]$

Column-major order

for $i = 1$ **to** m **do**
  $y[i] \leftarrow 0.0$

for $j = 1$ **to** n **do**
  for $i = 1$ **to** m **do**
    $y[i] \leftarrow y[i] + A[i,j] \times x[j]$

Sparse matrices: Sparse matrix vector multiplies

Need to compute $y \leftarrow Ax$ for an $m \times n$ sparse matrix A and suitable dense vectors y and x.

Algorithm when A is stored in the coordinate format.

Sparse matrices: Sparse matrix vector multiplies

Need to compute $y \leftarrow Ax$ for an $m \times n$ sparse matrix A and suitable dense vectors y and x.

Algorithm when A is stored in the coordinate format.

Coordinate format with τ nonzeros (`irn, jcn, A`)

for $i = 1$ **to** m **do**
  $y[i] \leftarrow 0.0$

for $k = 1$ **to** τ **do**
  $y[irn[k]] \leftarrow y[irn[k]] + A[k] \times x[jcn[k]]$

Sparse matrices: Sparse matrix vector multiplies

Need to compute $y \leftarrow Ax$ for an $m \times n$ sparse matrix A and suitable dense vectors y and x.

Algorithm when A is stored in the CSR format.

Sparse matrices: Sparse matrix vector multiplies

Need to compute $y \leftarrow Ax$ for an $m \times n$ sparse matrix A and suitable dense vectors y and x.

Algorithm when A is stored in the CSR format.

Compressed storage by rows
(ia, jcn, A)

for $i = 1$ **to** m **do**
  $val \leftarrow 0.0$
  for $k = ia[i]$ **to** $ia[i+1] - 1$ **do**
    $val \leftarrow val + A[k] \times x[jcn[k]]$
  $y[i] \leftarrow val$

Sparse matrices: Sparse matrix vector multiplies

Need to compute $y \leftarrow Ax$ for an $m \times n$ sparse matrix A and suitable dense vectors y and x.

Algorithm when A is stored in the CSC format.

Sparse matrices: Sparse matrix vector multiplies

Need to compute $y \leftarrow Ax$ for an $m \times n$ sparse matrix A and suitable dense vectors y and x.

Algorithm when A is stored in the CSC format.

Compressed storage by columns (`ja, irn, A`)

for $i = 1$ **to** m **do**
  $y[i] \leftarrow 0.0$

for $j = 1$ **to** n **do**
  $xval \leftarrow x[j]$
  for $k = ja[j]$ **to** $ja[j+1] - 1$ **do**
    $y[irn[k]] \leftarrow y[irn[k]] + A[k] \times xval$

Sparse matrices: Sparse matrix vector multiplies

- Characterizes a wide range of applications with irregular computational dependency.
  - reduction operation from inputs (here entries of x) to outputs (here entries of y), e.g., get the max in a set, for many different sets. How?
- A fine grain computation: each nnz is read/operated on once. Guaranteeing efficiency will guarantee efficiency in applications with a coarser grain computation.

Sparse matrices: Sparse matrix vector multiplies

SpMxV's of the form $y \leftarrow Ax$ are the computational kernel of many scientific computations

- Solvers for linear systems, linear programs, eigensystems, least squares problems.
- Repeated SpMxV with the same large sparse matrix A.
- The matrix A can be symmetric, unsymmetric, rectangular.
- Sometimes multiplies are of the form $y \leftarrow ADA^T z$ with a diagonal D (in interior point methods for linear programs).
  - computation proceeds (why?) as $w \leftarrow A^T z$, then $x \leftarrow Dw$, then $y \leftarrow Ax$
- Sometimes we have multiplies with A and A^T independent; $y \leftarrow Ax$ and $w \leftarrow A^T z$ (QMR, CGNE, and CGNR methods with square unsymmetric A; rectangular A in Lanczos method).

Further questions

Given a real matrix A of size $m \times n$, with τ nonzeros.

Question: Write a function to create its transpose (three times: A is stored in the COO, CSR, and CSC formats, the transpose should be stored in the same format).

Question: (HW1-Q2 for 24 nov) Design an algorithm (in pseudocode) to test if A is pattern-wise symmetric. Assume A is stored in the CSC format (we need to access `ja`, `irn`, but not to `A`). What is the run time?

Outline

Graph notations and definitions

A graph $G = (V, E)$ consists of a finite set V, called the vertex set and a finite, binary relation E on V, called the edge set.

Three standard graph models

Undirected graph: The edges are unordered pair of vertices, i.e., $\{u, v\} \in E$ for some $u, v \in V$.

Directed graph: The edges are ordered pair of vertices, that is, (u, v) and (v, u) are two different edges.

Bipartite graph: $G = (U \cup V, E)$ consists of two disjoint vertex sets U and V such that for each edge $(u, v) \in E$, $u \in U$ and $v \in V$.

An ordering or labelling of $G = (V, E)$ having n vertices, i.e., $|V| = n$, is a mapping of V onto $1, 2, \ldots, n$.

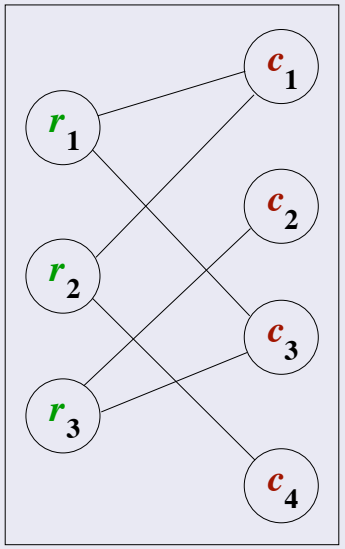
Matrices and graphs: Rectangular matrices

The rows/columns and nonzeros of a given sparse matrix correspond (with the natural labelling) to the vertices and edges, respectively, of a graph.

Rectangular matrices

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} \times & & \times & \\ \times & & & \times \\ & \times & \times & \end{pmatrix} \end{array}$$

Bipartite graph



The set of rows corresponds to R, the set of columns corresponds to C such that for each $a_{ij} \neq 0$, (r_i, c_j) is an edge.

Matrices and graphs: Square unsymmetric pattern

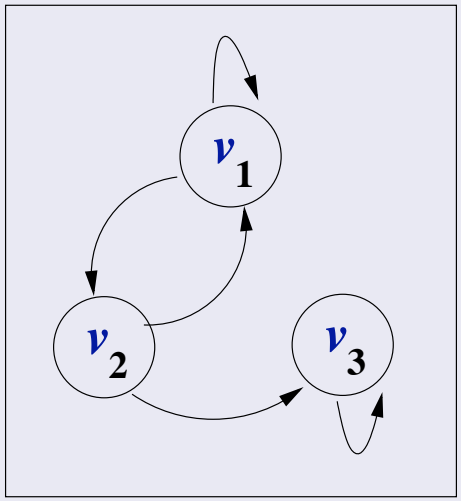
The rows/columns and nonzeros of a given sparse matrix correspond (with natural labelling) to the vertices and edges, respectively, of a graph.

Square unsymmetric pattern matrices

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} \times & \times & \\ \times & & \times \\ & & \times \end{pmatrix} \end{array}$$

Graph models

- Bipartite graph as before.
- Directed graph



The set of rows/cols corresponds the vertex set V such that for each $a_{ij} \neq 0$, (v_i, v_j) is an edge. Transposed view possible too, i.e., the edge (v_i, v_j) directed from column i to row j. Usually self-loops are omitted.

Matrices and graphs: Square unsymmetric pattern

A special subclass

Directed acyclic graphs (DAG): A directed graphs with no loops (maybe except for self-loops).

DAGs

We can sort the vertices such that if (u, v) is an edge, then u appears before v in the ordering.

Question: (Class) What kind of square matrices have a DAG? To be precise: Assume we created the directed graph. When it will be DAG?

Note: if we permute the matrix first unsymmetrically, then create the graph, the answer will be different. [See Fertin, Rusu, and Vialette, `https://doi.org/10.1007/978-3-662-48971-0_15`]

Matrices and graphs: Symmetric pattern

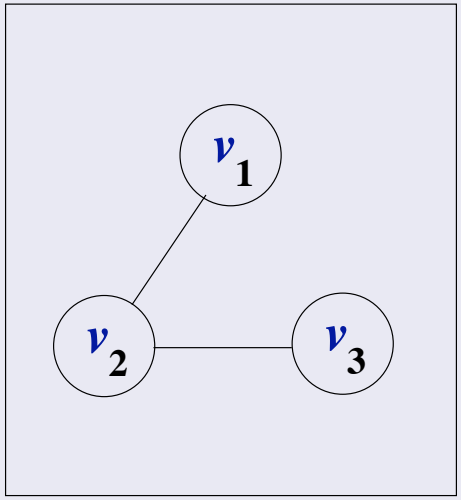
The rows/columns and nonzeros of a given sparse matrix correspond (with natural labelling) to the vertices and edges, respectively, of a graph.

Square symmetric pattern matrices

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} & \times & \\ \times & \times & \times \\ & \times & \times \end{pmatrix} \end{array}$$

Graph models

- Bipartite and directed graphs as before.
- Undirected graph



The set of rows/cols corresponds the vertex set V such that for each $a_{ij}, a_{ji} \neq 0$, $\{v_i, v_j\}$ is an edge. No self-loops; usually the main diagonal is assumed to be zero-free.

Matrices and graphs: How do we store a graph?

Vertex-edge adjacency matrix (sparse/dense)?

Vertex-vertex adjacency matrix (sparse/dense)?

Definitions: Edges, degrees, and paths

Many definitions for directed and undirected graphs are the same. We will use (u, v) to refer to an edge of an undirected or directed graph to avoid repeated definitions.

- An edge (u, v) is said to incident on the vertices u and v.
- For any vertex u, the set of vertices in $\mathrm{adj}(u) = \{v : (u, v) \in E\}$ are called the neighbors of u. The vertices in $\mathrm{adj}(u)$ are said to be adjacent to u.
- The degree of a vertex is the number of edges incident on it.
- A path p of length k is a sequence of vertices $\langle v_0, v_1, \ldots, v_k \rangle$ where $(v_{i-1}, v_i) \in E$ for $i = 1, \ldots, k$. The two end points v_0 and v_k are said to be connected by the path p, and the vertex v_k is said to be reachable from v_0.
- Simple path: has no vertex repetitions.

Definitions: Components

- An undirected graph is said to be connected if every pair of vertices is connected by a path.
- The connected components of an undirected graph are the equivalence classes of vertices under the "is reachable" from relation.

- A directed graph is said to be strongly connected if every pair of vertices are reachable from each other.
- The strongly connected components of a directed graph are the equivalence classes of vertices under the "are mutually reachable" relation.

Definitions: Trees and spanning trees

A tree is a connected, acyclic, undirected graph. If an undirected graph is acyclic but disconnected, then it is a forest.

Properties of trees

- Any two vertices are connected by a unique path.
- $|E| = |V| - 1$

A rooted tree is a tree with a distinguished vertex r, called the root.

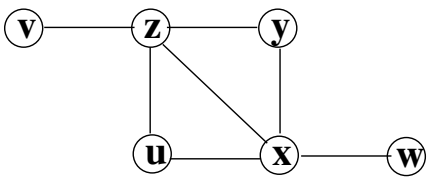
There is a unique path from the root r to every other vertex v. Any vertex y in that path is called an ancestor of v. If y is an ancestor of v, then v is a descendant of y.

The subtree rooted at v is the tree induced by the descendants of v, rooted at v.

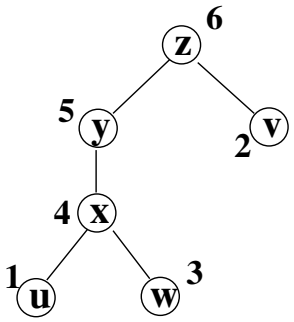
A spanning tree of a connected graph $G = (V, E)$ is a tree $T = (V, F)$, such that $F \subseteq E$.

Ordering of the vertices of a rooted tree

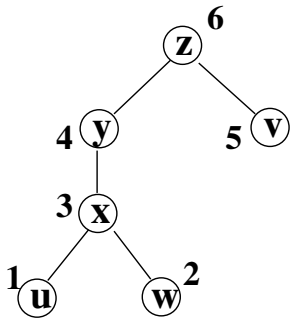
- A topological ordering of a rooted tree is an ordering that numbers children vertices before their parent.
- A postorder is a topological ordering which numbers the vertices in any subtree consecutively.



Connected graph G



Rooted spanning tree
with topological ordering



Rooted spanning tre
with postordering

Postordering the vertices of a rooted tree

```
[porder]=PostOrder(T, r)
  porder ← [·]
  seen(v) ← False for all v ∈ T
  seen(r) ← True
  Push(S, r)
  while NotEmpty(S) do
    v ←Top(S)
    if ∃ a child c of v with seen(c) = False then
      seen(c) ← True
      Push(S, c)
    else
      Pop(S)
      porder ← [porder, v]
```

Again, have to run for each root, if T is a forest.

The algorithm runs in $\mathcal{O}(n)$ time for a tree with n nodes.

Permutation matrices

A permutation matrix is a square $(0,1)$-matrix where each row and column has a single 1.

If P is a permutation matrix, $\mathsf{P}\mathsf{P}^T = \mathsf{I}$, i.e., it is an orthogonal matrix. Let,

$$\mathsf{A} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} \times & \times & \\ \times & & \times \\ \times & & \times \end{pmatrix} \end{array}$$

and suppose we want to permute columns as $[2,3,1]$. Define $p_{2,1} = 1$, $p_{3,2} = 1$, $p_{1,3} = 1$, and $\mathsf{B} = \mathsf{A}\mathsf{P}$ (if column j to be at position i, set $p_{ji} = 1$)

$$\mathsf{B} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 2 & 3 & 1 \end{array} \\ \begin{pmatrix} \times & & \times \\ & \times & \times \\ & \times & \times \end{pmatrix} \end{array} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} \times & \times & \\ \times & & \times \\ \times & & \times \end{pmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} & & 1 \\ 1 & & \\ & 1 & \end{pmatrix} \end{array}$$

Permutation matrices

Question:

Given a matrix A, we want to permute the rows according to a given permutation.

Let $\pi(i)$ denote the new position of row i. Write a simple for-loop to create the permutation matrix P so that the matrix PA is permuted according to π.

Verify your solution on the matrix in the previous slide.

Permutation matrices

Question (Class): For P and Q permutation matrices, what can be said about their multiplication PQ?

Question (Class) : Let A be a square matrix and P be a permutation matrix. How do you describe PAP^T verbally?

Question (Class): Let A be a square matrix and P be a permutation matrix. What can you say about the diagonal elements of PAP^T?

Question (Class): Let A be a square matrix and P be a permutation matrix. What can you say about the graphs of A and PAP^T?

Matching in bipartite graphs and permutations

A matching in a graph is a set of edges no two of which share a common vertex. We will be mostly dealing with matchings in bipartite graphs.

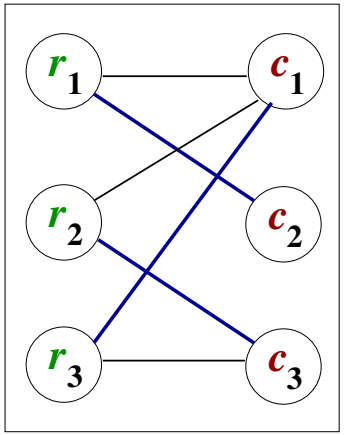
In matrix terms, a matching in the bipartite graph of a matrix corresponds to a set of nonzero entries no two of which are in the same row or column.

A vertex is said to be matched if there is an edge in the matching incident on the vertex, and to be unmatched otherwise. In a perfect matching, all vertices are matched.

The cardinality of a matching is the number of edges in it. A maximum cardinality matching or a maximum matching is a matching of maximum cardinality. Solvable in polynomial time.

Matching in bipartite graphs and permutations

Given a square matrix whose bipartite graph has a perfect matching, such a matching can be used to permute the matrix to have the matching entries are along the main diagonal.



$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \times & \bullet & \\ 2 & \times & & \bullet \\ 3 & \bullet & & \times \end{array}$$

$$\begin{array}{c|ccc} & 2 & 3 & 1 \\ \hline 1 & \bullet & & \times \\ 2 & & \bullet & \times \\ 3 & & \times & \bullet \end{array} = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \times & \bullet & \\ 2 & \times & & \bullet \\ 3 & \bullet & & \times \end{array} \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & & & 1 \\ 2 & 1 & & \\ 3 & & 1 & \end{array}$$

Perfect matchings and zero-free diagonals

Question:

Let A be an $n \times n$ sparse matrix having a perfect matching in its bipartite graph. Let $\mathcal{M}$ be such a matching, where `colmatch(j)=i` show that the column j is matched to row i in $\mathcal{M}$.

Write a pseudocode to create a permutation matrix Q such that AQ has the matching entries along its diagonal.

Write a pseudocode to create a permutation matrix P such that PA has the matching entries along its diagonal.

Definitions: Reducibility

Reducible matrix: An $n \times n$ square matrix A is reducible if there exists an $n \times n$ permutation matrix P such that

$$\mathsf{PAP}^T = \begin{pmatrix} \mathsf{A}_{11} & \mathsf{A}_{12} \\ O & \mathsf{A}_{22} \end{pmatrix},$$

where A_{11} is an $r \times r$ submatrix, A_{22} is an $(n-r) \times (n-r)$ submatrix, where $1 \leq r < n$.

Irreducible matrix: There is no such a permutation matrix.

Theorem: An $n \times n$ square matrix is irreducible iff its directed graph is strongly connected.

Proof: Follows by definition.

Why care irreducibility

Definition 1.15. For $n \geq 2$, an $n \times n$ complex matrix A is **reducible** if there exists an $n \times n$ permutation matrix[6] P such that

$$PAP^T = \begin{bmatrix} A_{1,1} & A_{1,2} \\ O & A_{2,2} \end{bmatrix}, \tag{1.39}$$

where $A_{1,1}$ is an $r \times r$ submatrix and $A_{2,2}$ is an $(n-r) \times (n-r)$ submatrix, where $1 \leq r < n$. If no such permutation matrix exists, then A is **irreducible**. If A is a 1×1 complex matrix, then A is irreducible if its single entry is nonzero, and reducible otherwise.

It is evident from Definition 1.15 that the matrix of (1.38) is reducible.

The term *irreducible* (*unzerlegbar*) was introduced by Frobenius (1912); it is also called *unreduced* and *indecomposable* in the literature.[7] The motivation for calling matrices such as those in (1.39) *reducible* is quite clear, for if we seek to solve the matrix equation $\tilde{A}\mathbf{x} = \mathbf{k}$, where $\tilde{A} := PAP^T$ is the partitioned matrix of (1.39), we can partition the vectors $\mathbf{x}$ and $\mathbf{k}$ similarly so that the matrix equation $\tilde{A}\mathbf{x} := \mathbf{k}$ can be written as

$$\begin{aligned} A_{1,1}\mathbf{x}_1 + A_{1,2}\mathbf{x}_2 &= \mathbf{k}_1, \\ A_{2,2}\mathbf{x}_2 &= \mathbf{k}_2. \end{aligned}$$

Thus, by solving the second equation for $\mathbf{x}_2$, and with this known solution for $\mathbf{x}_2$, solving the first equation for $\mathbf{x}_1$, we have *reduced* the solution of the original matrix equation to the solution of two *lower*-order matrix equations.

Matrix Iterative Analysis, by Richard S. Varga, Second (revised and expanded edition of Prentice-Hall, Englewood Cliffs, New Jersey, 1962), Springer, 2000.
(page 18 contains historical references to irreducibility)

More on reducibility

The matrix $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ is irreducible (we just do not have many alternatives to permute it symmetrically and obtain the required form).

Yet, clearly it is not hard to solve $Ax = b$; the solution is $x_2 = b_1$ and $x_1 = b_2$.

Here comes something more general than irreducibility.

Definitions: Fully indecomposability

Fully indecomposable matrix: There is no permutation matrices P and Q such that

$$PAQ = \begin{pmatrix} A_{11} & A_{12} \\ O & A_{22} \end{pmatrix},$$

with the same condition on the blocks and their sizes as above.

Theorem: An $n \times n$ square matrix A is fully indecomposable iff for some permutation matrix Q, the matrix AQ is irreducible and has a zero-free main diagonal.

Proof: Anyone wants to try? [We will see later.]

Definitions: Cliques and independent sets

Clique

In an undirected graph $G = (V, E)$, a set of vertices $S \subseteq V$ is a clique if for all $s, t \in S$, we have $(s, t) \in E$.

Maximum clique: A clique of maximum cardinality (finding a maximum clique in an undirected graph is NP-complete).

Maximal clique: A clique is a maximal clique, if it is not contained in another clique.

In a symmetric matrix A, a clique corresponds to a subset of rows R and the corresponding columns such that the matrix $A(R, R)$ is full.

Independent set

A set of vertices is an independent set if none of the vertices are adjacent to each other. Can we find the largest one in polynomial time?

In a symmetric matrix A, an independent set corresponds to a subset of rows R and the corresponding columns such that the matrix $A(R, R)$ is either zero, or diagonal.

Definitions: More on cliques

Clique: In an undirected graph $G = (V, E)$, a set of vertices $S \subseteq V$ is a clique if for all $s, t \in S$, we have $(s, t) \in E$.

In a symmetric matrix A corresponds to a subset of rows R and the corresponding columns such that the matrix $A(R, R)$ is full.

Cliques in bipartite graphs: Bi-cliques

In a bipartite graph $G = (U \cup V, E)$, a pair of sets $\langle R, C \rangle$ where $R \subseteq U$ and $C \subseteq V$ is a bi-clique if for all $a \in R$ and $b \in C$, we have $(a, b) \in E$.

In a matrix A, corresponds to a subset of rows R and a subset of columns C such that the matrix $A(R, C)$ is full.

The maximum node bi-clique problem asks for a bi-clique of maximum size (e.g., $|R| + |C|$), and it is polynomial time solvable, whereas maximum edge bi-clique problem (e.g., asks for a maximum $|R| \times |C|$) is NP-complete.

Definitions: Hypergraphs

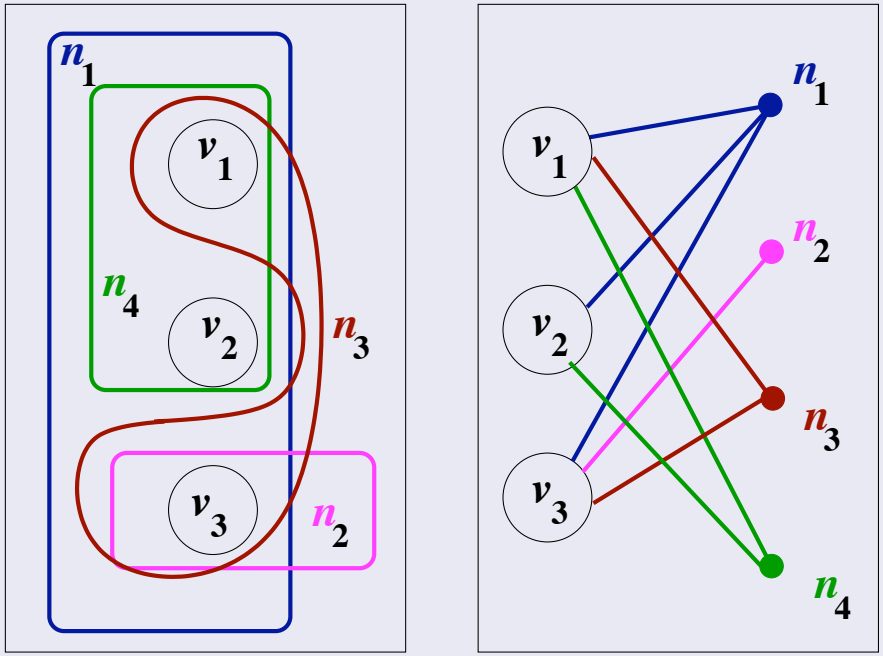
Hypergraph: A hypergraph $H = (V, N)$ consists of a finite set V called the vertex set and a set of non-empty subsets of vertices N called the hyperedge set or the net set. A generalization of graphs.

For a sparse matrix A, define a hypergraph whose vertices correspond to the rows and whose nets correspond to the columns such that vertex v_i is in net n_j iff $a_{ij} \neq 0$ (the column-net model).

A sample matrix

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} \times & & \times & \times \\ \times & & & \times \\ \times & \times & \times & \end{pmatrix} \end{array}$$

The column-net hypergraph model



Basic graph algorithms

We assume some familiarity with

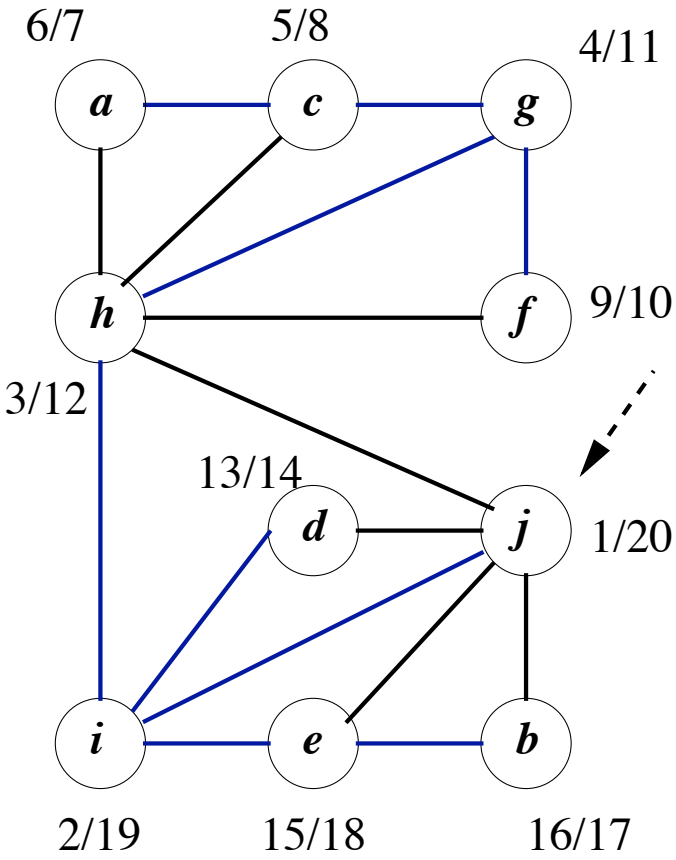
- Breadth-first search, BFS tree
- Depth-first search, DFS tree (in particular edge classifications)
- Topological sort of DAGs, finding connected components of undirected graphs and strongly connected components of directed graphs.
- Transitive closure or reduction of a directed graph.

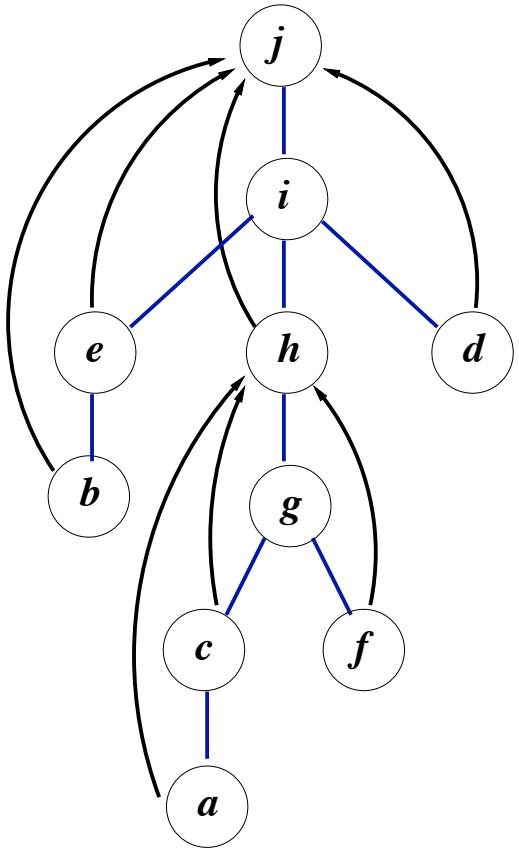
If not familiar, the book by Cormen, Leiserson, Rivest, and Stein is highly recommended.

Depth-first search: Undirected graphs

Edge classification

Any DFS on an undirected graph produces only Tree and Back edges.





BFS and matrix-vector multiplication

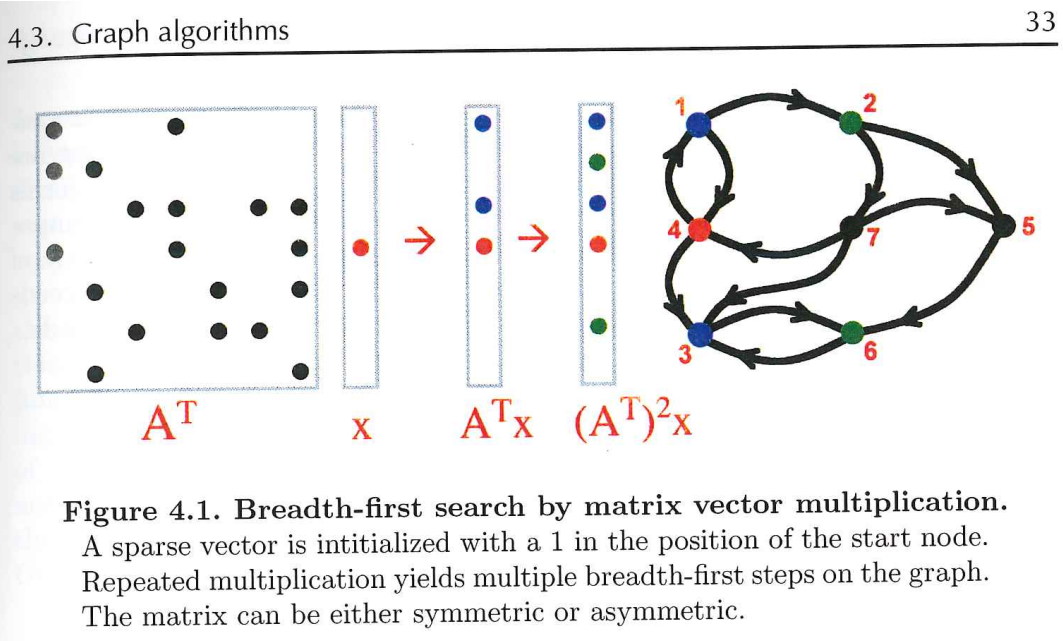
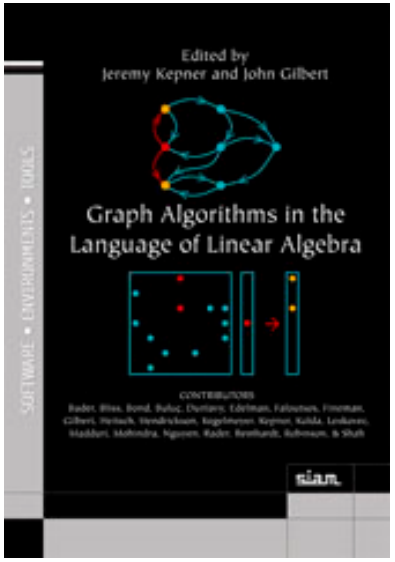


Figure 4.1. Breadth-first search by matrix vector multiplication. A sparse vector is intitialized with a 1 in the position of the start node. Repeated multiplication yields multiple breadth-first steps on the graph. The matrix can be either symmetric or asymmetric.



- BFS can be performed by sparse matrix vector multiplication (G and x)
- To search from i, begin with $x(i) = 1$ and $x(j) = 0$ for $i \neq j$.
- $G^T x$ picks out row i; $G^T(G^T x)$ gives vertices two steps away etc.
- With $A = G + I$ in place of G we select vertices at distance at most k in the kth step.

Could not get enough of it: Questions

How would you describe the following in the language of graphs

- the structure of PAP^T for a given square sparse matrix A and a permutation matrix P.
- the structure of PAQ for a given square sparse matrix A and two permutation matrices P and Q.
- the structure of A^k, for $k > 1$, where A has a zero diagonal. What about A^k when A has a zero-free diagonal?
- the structure of AA^T.
- the structure of the vector b, where $b = Ax$ for a given sparse matrix A, and a sparse vector x.

Could not get enough of it: Questions

Can you define:

- the row-net hypergraph model of a matrix.
- a matching in a hypergraph (is it a hard problem?).

Can you relate:

- the DFS or BFS on a tree to a topological ordering? postordering?