Sparse matrices: Basics

Bora Uçar

RO:MA, LIP, ENS Lyon, France

CR-08: Combinatorial scientific computing, September 2014
http://perso.ens-lyon.fr/bora.ucar/CR08/
Outline

1. Course presentation
2. Sparse matrices
CR08 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,...).

\[
A = \begin{bmatrix}
4 & -1 & 0 & 0 & 0 & -1 \\
-1 & 3 & -1 & -1 & 0 & 0 \\
0 & -1 & 3 & 0 & -1 & 0 \\
0 & -1 & 0 & 2 & 0 & 0 \\
0 & 0 & -1 & 0 & 2 & -1 \\
-1 & 0 & 0 & 0 & -1 & 2
\end{bmatrix}
\]
CR08: What is and what is not?

- CR08 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.

  CR08 is about this correspondence in a broader sense.
CR08: 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.
CR08: A sample problem

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

\[
A = H_0 = \begin{pmatrix} d_1 & v_1^T \\ v_1 & H_1 \end{pmatrix} \\
= \begin{pmatrix} \sqrt{d_1} & 0 \\ \frac{v_1}{\sqrt{d_1}} & I_{n-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & H_1 \end{pmatrix} \begin{pmatrix} \sqrt{d_1} & v_1^T \\ 0 & \frac{1}{\sqrt{d_1}} \end{pmatrix} \\
= L_1 A_1 L_1^T, \text{ where}
\]

\[
H_1 = H_1 - \frac{v_1 v_1^T}{d_1}
\]

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

\[
A = (L_1 L_2 \cdots L_{n-1}) I_n (L_{n-1}^T \cdots L_2^T L_1^T) = LL^T
\]
The basic step: \[ H_1 = \overline{H_1} - \frac{v_1v_1^T}{d_1} \]

What is \( v_1v_1^T \) in terms of structure?

\( v_1v_1^T \): a dense sub-matrix in \( H_1 \).

\( v_1 \) 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.
CR08: A sample problem

What can we say about the final matrix by just looking at the original graph?
Instructor: Lecture notes (references to articles, books); problems.

Students: Participation to solving problems;

Presentations (can be a survey, of type expository, or about a research problem). We will decide the form according to the number of participants.

Classes to catch up: BU is away 10–14 November. We will decide the dates later for the course of that week.

Office: GN1 Nord 332 (LIP)
Contact: bora.ucar@ens-lyon.fr
Course web: http://perso.ens-lyon.fr/bora.ucar/CR08/
Lecture notes will be posted.
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.
There are many ways to store a sparse matrix. We will look at three standard representations which store only the nonzero entries. We use $\tau$ 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 (Triplet) format**

Two integer arrays (irn, jcn) and a double array $A$, each of size $\tau$:

- $\text{irn} = [1, 2, 2, 3, 3, 4, 5, 5, 5]$
- $\text{jcn} = [1, 2, 4, 1, 3, 4, 2, 4, 5]$
- $A = [1.1, 2.2, 2.4, 3.1, 3.3, 4.4, 5.2, 5.4, 5.5]$

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

The storage is $2\tau$ integers and $\tau$ doubles (or single or complex). In general, $\tau = \mathcal{O}(m + n)$. 
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.
There are many ways to store a sparse matrix.

We will look at three standard representations which store only the nonzero entries.

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

Compressed row storage

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 \(i\)th row are stored at the \(ia[i] \ldots ia[i+1]-1\) 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 row storage

There are many ways to store a sparse matrix.

We will look at three standard representations which store only the nonzero entries.

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

Compressed row format

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 \(i\)th row are stored at the \(ia[i] \ldots ia[i+1]-1\) positions of \(jcn\) and \(A\).

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

Question: Let $A$ be an $m \times n$ matrix stored in the compressed row storage format. Where is $\text{nnz}(A)$?

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

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 compressed row storage format.
Sparse matrices: Compressed column storage

There are many ways to store a sparse matrix.

We will look at three standard representations which store only the nonzero entries.

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

Compressed column storage

Two integer arrays \((\text{irn}, \text{ja})\) and a double array \(A\):

\[
\text{ja} = [1 \quad 3 \quad 5 \quad 6 \quad 9 \quad 10]
\]
\[
\text{irn} = [1 \quad 3 \quad 2 \quad 5 \quad 3 \quad 2 \quad 4 \quad 5 \quad 5]
\]
\[
A = [1.1 \quad 3.1 \quad 2.2 \quad 5.2 \quad 3.3 \quad 2.4 \quad 4.4 \quad 5.4 \quad 5.5]
\]

The nonzeros of the \(j\)th column are stored at the positions of \(\text{irn}\) and \(A\).

For example the 2nd col: starts at \(\text{ja}[2] = 3\) and finishes at \(\text{ja}[2+1]-1 = 4\). The row indices are therefore \(\text{irn}[3,4] = 2\ 5\) and values are \(A[3,4] = 2.2\ 5.2\).
There are many ways to store a sparse matrix.

We will look at three standard representations which store only the nonzero entries.

**Compressed column format**

Two integer arrays \((\text{irn}, \text{ja})\) and a double array \(A\):

\[
\text{ja} = [1 \ 3 \ 5 \ 6 \ 9 \ 10 ]
\]

\[
\text{irn} = [1 \ 3 \ 2 \ 5 \ 3 \ 2 \ 4 \ 5 \ 5 ]
\]

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

The nonzeros of the \(j\)th column are stored at the \(\text{ja}[j] \cdots \text{ja}[j+1]-1\) positions of \(\text{irn}\) and \(A\).

Let matrix be of size \(m \times n\), and \(\tau\) be the number of nonzeros, then the storage is \(n + 1 + \tau\) integer and \(\tau\) double (or single or complex).
Question: Let $A$ be an $m \times n$ matrix stored in the compressed column storage format. Where is $\text{nnz}(A)$?

Question: Let $A$ be an $m \times n$ matrix stored in the coordinate format. Write a function/routine that creates $A$ in csc format.
Reminder: Dense matrix vector multiplication

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

**Row-major order**

\[
\text{for } i = 1 \text{ to } m \text{ do} \\
y[i] \leftarrow 0.0 \\
\text{for } j = 1 \text{ to } n \text{ do} \\
y[i] \leftarrow y[i] + A[i, j] \ast x[j]
\]

**Column-major order**

\[
\text{for } i = 1 \text{ to } m \text{ do} \\
y[i] \leftarrow 0.0 \\
\text{for } j = 1 \text{ to } n \text{ do} \\
y[i] \leftarrow y[i] + A[i, j] \ast x[j] \\
\text{for } i = 1 \text{ to } m \text{ do} \\
y[i] \leftarrow y[i] + A[i, j] \ast x[j]
\]
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 $x$. $A$ is stored in the coordinate format.

```
Coordinate format with $\tau$ nonzeros (irn, jcn, A)

for $i = 1$ to $m$ do
  $y[i] \leftarrow 0.0$
for $k = 1$ to $\tau$ do
  $y[irn[k]] \leftarrow y[irn[k]] + A[k] \times jcn[k]]$
```
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 \( y \leftarrow A x \) 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 row storage

\((ia, jcn, A)\)

\[
\text{for } i = 1 \text{ to } m \text{ do} \\
\quad \text{val} \leftarrow 0.0 \\
\quad \text{for } k = ia[i] \text{ to } ia[i + 1] - 1 \text{ do} \\
\quad \quad \text{val} \leftarrow \text{val} + A[k] \times x[jcn[k]] \\
\quad y[i] \leftarrow \text{val}
\]
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.
Need to compute \( y \leftarrow A x \) 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 column storage 
\((ja, irn, A)\)

\[
\begin{align*}
\text{for } i &= 1 \text{ to } m \text{ do} \\
& \quad y[i] \leftarrow 0.0 \\
\text{for } j &= 1 \text{ to } n \text{ do} \\
& \quad xval \leftarrow x[j] \\
& \quad \text{for } k = ja[j] \text{ to } ja[j + 1] - 1 \text{ do} \\
& \quad \quad y[irn[k]] \leftarrow y[irn[k]] + A[k] \ast xval
\end{align*}
\]
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 $y$)
- A fine grain computation: each nnz is read/operated on once. Guaranteeing efficiency will guarantee efficiency in applications with a coarser grain computation.
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 matrix $D$ (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).
Given a real matrix $A$ of size $m \times n$, with $\tau$ nonzeros.

Question: Write a function to create its transpose (three times: stored in the COO, CSR, and CSC formats).

Question: Write a function to determine if $A$ is symmetric. Assume $A$ is stored in the CSC format.