ECE 4960

Spring 2017

Lecture 10

Matrix Pivoting

Edwin C. Kan
School of Electrical and Computer Engineering
Cornell University

Example for Importance of Pivoting

Use the round-off rule to keep only 4 digits of precision for the following problem by Gaussian elimination from the first row:

$$\begin{cases} 0.003000x_1 + 59.14x_2 = 59.17 \\ 5.291x_1 - 6.130x_2 = 46.78 \end{cases}$$

x_1 = –10.00 and x_2 = 1.001. Substitute back to know accuracy!

$$\begin{cases} 59.14x_2 + 0.003000x_1 = 59.17 \\ -6.130x_2 + 5.291x_1 = 46.78 \end{cases}$$

x_1 = 10.00 and x_2 = 1.000. Substitute back to know accuracy!

Only column permutation is done!!!!

Pivoting for Precision Preservation

- Appropriate **pivoting** can be guided by highest **precision preservation** or by **minimal fill-ins** for the sparse matrix.
- These two criteria can be in **conflict**: the pivoting choice for precision preservation and the pivoting for minimal fill-in can cause severe degradation for each other!
- If only column vector permutations are considered when we choose the pivot element, it is called **partial** or **column pivoting**.
- If we consider both column and row permutations, it is then called **total**, **full** or **maximum pivoting**.
- The search of the best pivoting for precision preservation is an $O(n^3)$ operation, similar to the symbolic LU factorization for minimal fill-in, but we need to know the **numerical** elements of each element to choose pivoting for precision preservation.

Pivot-on-the-Fly

- To resolve the choices between minimum fill-in and maximum pivoting: **pivot-on-the-fly**
- A pivoting tolerance is set to invoke alternative pivoting choices from minimal fill-ins and take sacrifice of the increase in non-zero fill-in.
- Pivot-on-the-fly is expensive too, as choosing the pivot and changing the computing order in LU decomposition are O(n^3).
- For many practical matrices, we do not need maximum pivoting to control the round-off errors, such as when the matrix is **diagonally dominant**.

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \qquad \text{for all } i$$

Practical Heuristics for Pivot-on-the-Fly

$$|a_{ii}| > S_{th} \cdot \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \qquad \text{for all } i$$

- S_{th} is the scaling threshold for tolerance between 10^{-1} – 10^{-5}.
- As the minimal fill-in algorithm is performed at the symbolic step when the value of a_{ij} is not yet known, the pivot-on-the-fly algorithm will be triggered to alter the LU factorization operation with different pivoting choices only when Eq. (13) is violated.
  - Set S_{th} to 10^{-1} when the memory and computational time constraints are not stringent constraints.
  - Set S_{th} to 10^{-5} when we hope to keep most of the minimal fill-in pivoting choices to save memory and computational time.

Vector and Matrix Norms

- To estimate the conditioning of vectors and matrices, we often use "norms" to sum up the contribution from each element.
- Any "norm" needs to satisfy the three criteria below:
  1. $\|x\|$ is positive, and only equal to zero when all $x_i = 0$.
  2. $\|ax\| = |a| \cdot \|x\|$ where a is any complex scalar.
  3. $\|x + y\| \leq \|x\| + \|y\|$ (this is also referred as the "triangular rule")
- Popular vector norms:
  1. First norm: $\|x\|_1 = \sum_{i=1}^{n} |x_i|$
  2. Second: $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$
  3. Infinite: $\|x\|_\infty = \max_{i=1,n} |x_i|$

Matrix Norms

- Frobenius matrix norms: $\|A\|_{F2} = \sqrt{\sum_{i,j=1}^{n} a_{ij}^2}$

- Frobenius norm does not indicate the matrix properties in a useful manner. A better and more general definition is to view *A* as a transformation for the vector *x* that it applies to

$$\|A\| = \max_{\forall x \neq 0} \frac{\|Ax\|}{\|x\|}$$

$$\|A\|_1 \leq \max_k \sum_j |a_{jk}|$$ Maximum column vector sum

$$\|A\|_\infty \leq \max_j \sum_k |a_{jk}|$$ Maximum row vector sum

Hacker Practice

Calculate the upper bounds of $\|A\|_1$ and $\|A\|_\infty$ in the full-matrix and sparse-matrix formats. Preserve the function, as we will use the results later.

$$A = \begin{pmatrix} 1 & 2 & 0 & 0 & 3 \\ 4 & 5 & 6 & 0 & 0 \\ 0 & 7 & 8 & 0 & 9 \\ 0 & 0 & 0 & 10 & 0 \\ 11 & 0 & 0 & 0 & 12 \end{pmatrix}$$

Matrix and Its Inverse

- Matrix conditioning: how perturbation in matrix elements of a_{ij} will affect the solution of $Ax = b$.
- When A^{-1} exists and $AA^{-1} = I$, but it is often hard to compute A^{-1} directly, which is not sparse even if A has high sparsity.
- Cramer's rule for matrix inverse:

$$A_{ij}^{-1} = \frac{1}{\det(A)}\left(\det\left(subA_{ij}\right)\right)$$

$subA_{ij}$ is the matrix reduced from A by deleting the i-th row and j-th column

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \Rightarrow \quad A^{-1} = \frac{1}{(ad - bc)}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Matrix Conditioning

- Taking a delta of $Ax = b$.

$$\Delta A \cdot x + A\Delta x = 0$$

$$\|\Delta x\| \leq \left\|A^{-1}\right\| \cdot \|\Delta A\| \cdot \|x\|$$

$$\frac{\|\Delta x\|}{\|x\|} \leq \left\|A^{-1}\right\| \cdot \|A\| \cdot \frac{\|\Delta A\|}{\|A\|}$$

- Matrix conditioning of A: $\kappa_A \equiv \left\|A^{-1}\right\| \cdot \|A\|$

- However, it is usually too difficult to know $\|A^{-1}\|$!
- So, often we just use D^{-1} where $D = \text{diag}(A)$.

Column Pivoting and Diagonal Conditioning Example

- For the 4-digit precision, the previous ill-conditioned problem :

$$\begin{cases} x_1 + 19710 x_2 = 19720 \\ -0.8631 x_1 + x_2 = 7.631 \end{cases}$$

Column pivoting and permutation

$$\begin{bmatrix} 19710 & 1 \\ 1 & -0.8631 \end{bmatrix} \cdot \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} 19720 \\ 7.631 \end{bmatrix}$$

Matrix conditioning by diagonal normalization

$$\begin{bmatrix} 1 & 5.074 \times 10^{-5} \\ -11.59 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1.001 \\ -8.841 \end{bmatrix}$$