MAT 226B Large Scale Matrix Computation
Homework 1

Ahmed Mahmoud

January, 30th 2020

Problem 1:

(a) Let $A = [a_{j,k}] \in \mathbb{R}^{n\times n} \succ 0$ and $L = [l_{j,k}]$ be its Cholesky factor. Using MATLAB notation, Algorithm 1 shows Cholesky factorization algorithm

Algorithm 1: Cholesky Factorization

Input: $A = [a_{j,k}] \in \mathbb{R}^{n\times n} \succ 0$

Output: $L = [l_{j,k}]$ such that $A = LL^T$

1 $l_{j,k} = a_{j,k}, \forall j \geq k$ and $j, k = 1, 2, \ldots, n$

2 **for** $k = 1, 2, \ldots, n$ **do**

3 $\quad l_{k,k} = \sqrt{l_{k,k}}$

4 $\quad l_{k+1:n,k} = \frac{1}{l_{k,k}} l_{k+1:n,k}$

5 $\quad$ **for** $j = k+1, k+2, \ldots, n$ **do**

6 $\quad\quad l_{j:n,j} = l_{j:n,j} - l_{j:n,k} l_{jk}$

7 $\quad$ **end**

8 **end**

Line 1 in Algorithm 1 is a memory copy and does not include any flops. Line 3 accounts for n square root operations. On iteration k, Line 4 will account for $n-k$ division operations. Since this loop goes from $k = 1, 2, \ldots, n$, we get $\sum_{i=1}^{n}(n-k) = \frac{1}{2}n(n-1)$ division operation.

Line 6 does two operations; subtraction and multiplication, each on a vector of length $(n - j + 1)$. Thus, the total cost of the inner loop is

$$\sum_{k=1}^{n} \sum_{j=k+1}^{n} 2(n-j+1) = \frac{1}{3}n(n^2-1)$$

Thus, the total cost of Algorithm 1 is

$$n + \frac{1}{2}n(n-1) + \frac{1}{3}n(n^2-1) \text{ flops}$$

(b) Let A be a banded $n \times n$ matrix with bandwidth $2p+1$, i.e., $a_{jk} = 0$ if $|j-k| > p$. To show that Cholesky factor L has lower bandwidth p, i.e., $l_{jk} = 0$ if $j - k > p$, we need to show the Cholesky factorization does not introduce any fill-in's. Line 3 and 4 in Algorithm 1 do not introduce any fill-in's.

At step k, the factor l_{jk} in Line 6 will be non-zero only for $k \leq j \leq k+p$. Thus, the only possible fill-in's for column j (i.e., at iteration j of the inner loop) is at the first p rows below the diagonal which are already non-zero since A is banded matrix with bandwidth $2p+1$ which means there is p non-zero rows below the diagonal already.

(c) Cholesky factorization can be re-written more efficiently for banded matrices since it is guaranteed to *not* introduce fill-in such that computation can be skipped for the zero elements. Algorithm 2 shows Cholesky factorization algorithm for banded matrices

Algorithm 2: Cholesky Factorization for Banded Matrices

Input: $A = [a_{j,k}] \in \mathbb{R}^{n\times n} \succ 0$ with bandwidth $2p+1$

Output: $L = [l_{j,k}]$ such that $A = LL^T$

1 $l_{j,k} = a_{j,k}, \forall j \geq k$ and $j, k = 1, 2, \dots, n$

2 **for** $k = 1, 2, \dots, n$ **do**

3 $\quad l_{k,k} = \sqrt{l_{k,k}}$

4 $\quad l_{k+1:k+p,k} = \frac{1}{l_{k,k}} l_{k+1:k+p,k}$

5 $\quad$ **for** $j = k+1, k+2, \dots, k+p$ **do**

6 $\quad\quad l_{j:k+p,j} = l_{j:k+p,j} - l_{j:k+p,k} \quad l_{j,k}$

7 $\quad$ **end**

8 **end**

In the algorithm above, we note the following

- Line 4 now only operates on the first p rows below the diagonal elements of the column k.
- Line 5 only goes through the first p columns after column k (during iteration k) since the factor l_{jk} (Line 6) will be zero for $j > k+p$
- Line 6 now only operates up to row $k+p$ since the rows below $k+p$ for column k (i.e., at iteration k) will contain zeros.

(d) Algorithm 2 requires n square root operations. Line 4 requires only p division. Since Line 4 runs for all k values, then the total number of division done by Line 4 is np.

Line 6 costs one subtraction and one division. Thus the total cost of the whole loop (Line 5-7) is

$$\sum_{k=1}^{n} \sum_{j=k+1}^{k+p} \sum_{i=j}^{p+k} 2 = np(p+1)$$

Thus, the total cost of Algorithm 2 is

$$n(p+1)^2$$

Problem 2:

(a) Taken an example for $m = 5$ to see the pattern of how the non-zero values arise in the Cholesky factor L

$$T_5 = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \Longrightarrow L^1 = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 \\ h_1 & 2-h_1^2 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

where $d_1 = \sqrt{2}$ and $h_1 = \frac{-1}{d1}$

$$L^2 = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 \\ h_1 & d_2 & 0 & 0 & 0 \\ 0 & h_2 & 2-h_2^2 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

where $d_2 = \sqrt{2-h_1^2}$ and $h_2 = \frac{-1}{d_2}$

$$L^3 = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 \\ h_1 & d_2 & 0 & 0 & 0 \\ 0 & h_2 & d_3 & 0 & 0 \\ 0 & 0 & h_3 & 2-h_3^2 & 0 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

where $d_3 = \sqrt{2-h_2^2}$ and $h_3 = \frac{-1}{d_3}$

Following until the final step, we can see that the diagonal elements of Cholesky factor L are $d_i = \sqrt{2-h_{i-1}^2}$ and the lower diagonal elements $h_i = \frac{-1}{d_i}$ where $h_0 = 0$.

(b)

Problem 4:

Figure 1 shows the associated graph $G(A)$ of matrix A along with the steps of the minimum degree algorithm. From these steps, the reordering of the nodes will be $2, 4, 5, 3, 6, 7, 1, 8, 9$

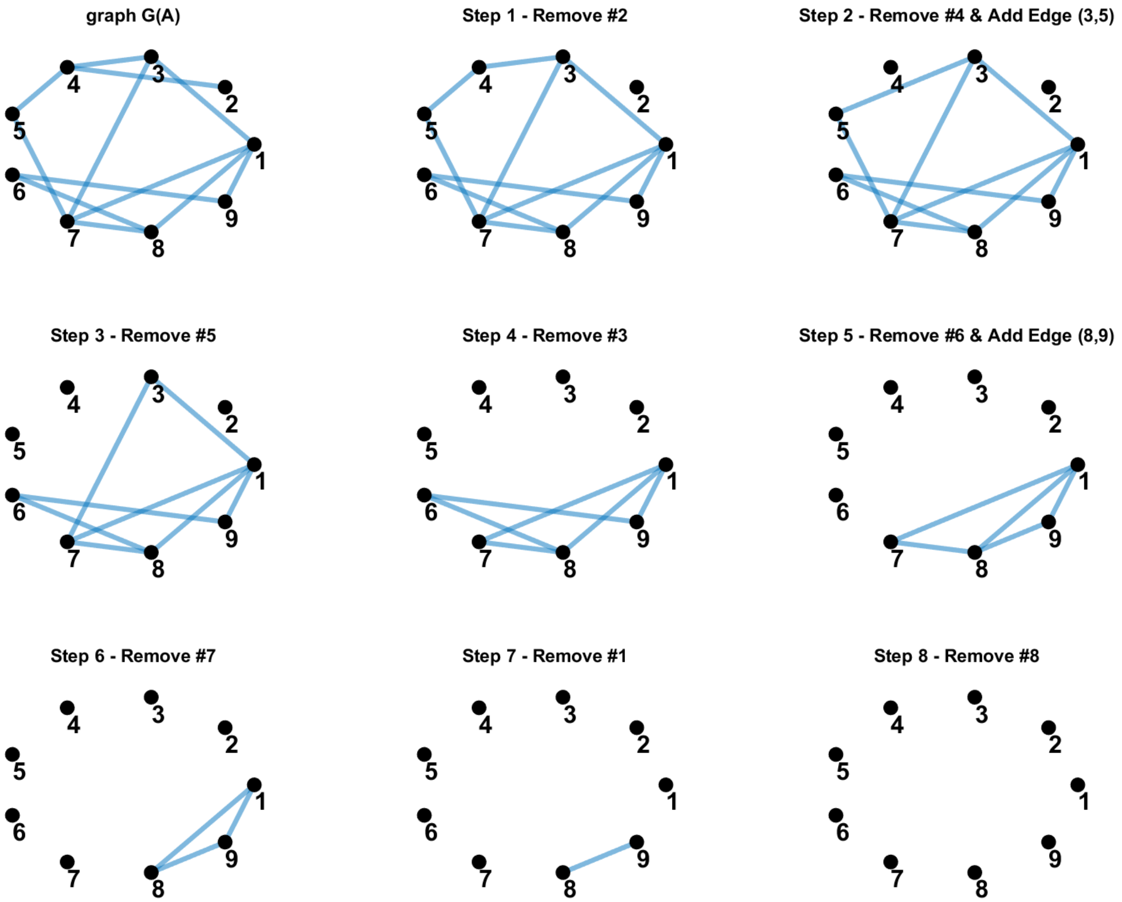

Figure 1: Graph $G(A)$ along with the 8 steps of the minimum degree algorithm applied on it.

From the reordering above, the permutation matrix can be constructed such that

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

From which, we can compute $P^T AP$ to be

$$P^T AP = \begin{bmatrix} * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & * & * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & * & 0 & * & * & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 & 0 & * & * \\ 0 & 0 & * & * & 0 & * & * & * & 0 \\ 0 & 0 & 0 & * & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * & * & 0 \\ 0 & 0 & 0 & 0 & * & 0 & * & 0 & * \end{bmatrix}$$

Applying Cholesky factorization to $P^T AP$ we get the following lower triangular matrix where the fill-in elements are shown with ${\color{red}+}$

$$L = \begin{bmatrix} * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & {\color{red}+} & * & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & * & 0 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & * & 0 & * & * & 0 & 0 \\ 0 & 0 & 0 & 0 & * & * & * & * & 0 \\ 0 & 0 & 0 & 0 & * & 0 & * & {\color{red}+} & * \end{bmatrix}$$

Problem 5:

(a) The number of non-zero entries in $L = 80$

Nonzero entries in L at column 4:

$$L(:,4) = \begin{bmatrix} (4,4) & 2.000000000000000e+00 \\ (5,4) & -5.000000000000000e-01 \\ (9,4) & -5.000000000000000e-01 \\ (15,4) & -5.000000000000000e-01 \\ (16,4) & -5.000000000000000e-01 \end{bmatrix}$$

Nonzero entries in L at column 7:

$$L(:,7) = \begin{bmatrix} (7,7) & 1.921537845661046e+00 \\ (8,7) & -4.003203845127178e-02 \\ (9,7) & -4.003203845127178e-02 \\ (11,7) & -8.006407690254357e-02 \\ (12,7) & -5.604485383178049e-01 \\ (15,7) & -8.006407690254357e-02 \\ (16,7) & -5.604485383178049e-01 \end{bmatrix}$$

Nonzero entries in L at column 10:

$$L(:,10) = \begin{bmatrix} (10,10) & 1.931795514549768e+00 \\ (11,10) & -5.581787456852771e-01 \\ (12,10) & -5.821358759653915e-03 \\ (13,10) & -2.238984138328429e-04 \\ (14,10) & -1.388170165763626e-01 \\ (15,10) & -4.097340973141025e-02 \\ (16,10) & -6.269155587319600e-03 \end{bmatrix}$$

Nonzero entries in L at column 13:

$$L(:,13) = \begin{bmatrix} (13,13) & 1.931485591744331e+00 \\ (14,13) & -5.180834914653560e-01 \\ (15,13) & -1.515180286251492e-01 \\ (16,13) & -2.197481923194983e-02 \end{bmatrix}$$

Nonzero entries in L at column 16:

$$L(:,16) = \begin{bmatrix} (16,16) & 1.717283221311613e+00 \end{bmatrix}$$

The relative error is

$$\frac{\| A - LL^T \|_2}{\| L \|_2} = 3.513060e-16$$