Compressed Sparse Column (CSC) Storage

Kalyani Gadgil

December 7, 2017

Need for Efficient Sparse Matrix Storage

Sparse matrices contain a lot of zeros and not all of them are useful information. The most common way is to use the CSR/CSC format.

Storage Complexity of Sparse Matrices for SUMMA

1. Blocks of size $(n/\sqrt{p}) \times (n/\sqrt{p})$
   where n - matrix dim and p - number of processors
2. Storing each of these submatrices in CSC format
   $O(n\sqrt{p} + +nnz)$
3. Storing whole matrix $O(n + nnz)$ on single processor
4. Storing matrix in DCSC format requires $O(nnz)$

CSC example - Column Major

Matrix A

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix} \end{array}$$

Column-major

$\text{nnz} = 4$

$\text{NUM} = [1, 2, 3, 4]$

$\text{row idx (IR)} = [0, 3, 2, 5]$

$\text{col idx} = [1, 1, 2, 5]$

CSC - Walkthrough

Matrix A

	0	1	2	3	4	5
0	0	1	0	0	0	0
1	0	0	0	0	0	0
2	0	0	3	0	0	0
3	0	2	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	4

JC stores indices of NUM where column changes

idx of NUM = [0]

NUM = [-]

IR = [-]

Column changed, so idx stored

JC = [0]

CSC example

Matrix A

	0	1	2	3	4	5
0	0	1	0	0	0	0
1	0	0	0	0	0	0
2	0	0	3	0	0	0
3	0	2	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	4

JC stores indices of NUM where column changes
idx of val = [0]

$$\text{NUM} = [\quad 1 \quad]$$

$$\text{IR} = [\quad 0 \quad]$$

Column changed, so idx stored

$$\text{JC} = [0, \quad 0 \quad]$$

CSC example

Matrix A

	0	1	2	3	4	5
0	0	1	0	0	0	0
1	0	0	0	0	0	0
2	0	0	3	0	0	0
3	0	2	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	4

JC stores indices of NUM where column changes

idx of val = [0, 1]

NUM = [1, 2]

IR = [0, 3]

Column not changed, so idx not stored

JC = [0, 0, ○]

CSC example

Matrix A

	0	1	2	3	4	5
0	0	1	0	0	0	0
1	0	0	0	0	0	0
2	0	0	3	0	0	0
3	0	2	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	4

JC stores indices of NUM where column changes

idx of val = [0, 1, 2]

NUM = [1, 2, 3]

IR = [0, 3, 2]

Column changed, so idx stored

JC = [0, 0, 2]

CSC example

Matrix A

	0	1	2	3	4	5
0	0	1	0	0	0	0
1	0	0	0	0	0	0
2	0	0	3	0	0	0
3	0	2	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	4

JC stores indices of NUM where column changes

idx of val = [0, 1, 2, 3]

NUM = [1, 2, 3, —]

IR = [0, 3, 2, —]

Column changed, so idx stored

JC = [0, 0, 2, 3]

CSC example

Matrix A

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & \boxed{4} & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix} \end{array}$$

JC stores indices of NUM where column changes

idx of val = [0, 1, 2, 3]

NUM = [1, 2, 3, $\boxed{-}$]

IR = [0, 3, 2, $\boxed{-}$]

Column changed, so idx stored

JC = [0, 0, 2, 3, $\boxed{3}$]

CSC example

Matrix A

	0	1	2	3	4	5
0	0	1	0	0	0	0
1	0	0	0	0	0	0
2	0	0	3	0	0	0
3	0	2	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	4

JC stores indices of NUM where column changes

idx of val = [0, 1, 2, 3]

NUM = [1, 2, 3, 4]

IR = [0, 3, 2, 5]

Column changed, so idx stored

JC = [0, 0, 2, 3, 3, 3]

CSC example

Matrix A

	0	1	2	3	4	5
0	0	1	0	0	0	0
1	0	0	0	0	0	0
2	0	0	3	0	0	0
3	0	2	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	4

JC stores indices of NUM where column changes

idx of val = [0, 1, 2, 3]

NUM = [1, 2, 3, 4]

IR = [0, 3, 2, 5]

NNZ stored as the n+1 element

JC = [0, 0, 2, 3, 3, 3, 4]

CSC example - Column Major

1. $O(n + nnz)$ comes from JC array of size $n + 1$

[0] [1] Buluc, A., & Gilbert, J. R. (2008). On the Representation and Multiplication of Hypersparse Matrices. https://doi.org/10.1109/IPDPS.2008.4536313

References

[1] Buluc, A., & Gilbert, J. R. (2008). On the Representation and Multiplication of Hypersparse Matrices. https://doi.org/10.1109/IPDPS 2008.4536313