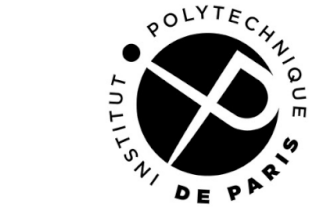
Graph Mining
Sparse Matrices

Clement Wang

2025 – 2026



INSTITUT
POLYTECHNIQUE
DE PARIS

Motivation

Large graphs are typically **sparse**

Dataset	#nodes	#edges	Density
Flights	2,939	30,500	$\approx 10^{-3}$
Amazon products	335k	925k	$\approx 10^{-5}$
Actors	382k	33M	$\approx 10^{-4}$
Wikipedia (en)	12M	378M	$\approx 10^{-6}$
Twitter	42M	1.5G	$\approx 10^{-6}$
Friendster	68M	2.5G	$\approx 10^{-7}$

Sparse matrices

$$\begin{bmatrix} 5 & 6 & 9 & 0 & 2 & 2 & 0 & 4 \\ 5 & 0 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 5 & 0 & 0 & 0 & 9 & 0 \end{bmatrix}$$

Efficient coding in Python

```
> from scipy import sparse
```

List of Lists

$$\begin{bmatrix} 5 & 6 & 9 & & 2 & 2 & & 4 \\ 5 & & & & 7 & & & \\ & & & & & & & \\ & & & & & & & \\ & 5 & & & & 3 & & \\ 6 & & & & & & 1 & 3 \\ & & 5 & & & & 9 & \end{bmatrix}$$

$$\text{data} = [[5, 6, 9, 2, 2, 4], [5, 7], [], [5, 3], [6, 1, 3], [5, 9]]$$

$$\text{rows} = [[0, 1, 2, 4, 5, 7], [0, 4], [], [1, 5], [0, 6, 7], [2, 6]]$$

Coordinate format

$$\begin{bmatrix} 5 & 6 & 9 & & 2 & 2 & & 4 \\ 5 & & & & 7 & & & \\ & & & & & & & \\ & 5 & & & & 3 & & \\ 6 & & & & & & 1 & 3 \\ & & 5 & & & & 9 & \end{bmatrix}$$

$$\text{shape} = (6, 8)$$

$$\begin{aligned} \text{data} &= (5, 6, 9, 2, 2, 4, 5, 7, 5, 3, 6, 1, 3, 5, 9) \\ \text{row} &= (0, 0, 0, 0, 0, 0, 1, 1, 3, 3, 4, 4, 4, 5, 5) \\ \text{col} &= (0, 1, 2, 4, 5, 7, 0, 4, 1, 5, 0, 6, 7, 2, 6) \end{aligned}$$

Compressed Sparse Row

$$\begin{bmatrix} 5 & 6 & 9 & & 2 & 2 & & 4 \\ 5 & & & & 7 & & & \\ & & & & & & & \\ & & & & & & & \\ & 5 & & & & 3 & & \\ 6 & & & & & & 1 & 3 \\ & & 5 & & & & 9 & \end{bmatrix}$$

$$\text{shape} = (6, 8)$$

$$\begin{aligned} \text{data} &= (5, 6, 9, 2, 2, 4, 5, 7, 5, 3, 6, 1, 3, 5, 9) \\ \text{indices} &= (0, 1, 2, 4, 5, 7, 0, 4, 1, 5, 0, 6, 7, 2, 6) \\ \text{indptr} &= (0, 6, 8, 8, 10, 13, 15) \end{aligned}$$

Compressed Sparse Row

$$\begin{bmatrix} 5 & 6 & 9 & & 2 & 2 & & 4 \\ 5 & & & & 7 & & & \\ & & & & & & & \\ & & & & & & & \\ & 5 & & & & 3 & & \\ 6 & & & & & & 1 & 3 \\ & & 5 & & & & 9 & \end{bmatrix}$$

$$\text{shape} = (6, 8)$$

$$\begin{aligned} \text{data} &= (5, 6, 9, 2, 2, 4, 5, 7, 5, 3, 6, 1, 3, 5, 9) \\ \text{indices} &= (0, 1, 2, 4, 5, 7, 0, 4, 1, 5, 0, 6, 7, 2, 6) \\ \text{indptr} &= (0, 6, 8, 8, 10, 13, 15) \end{aligned}$$

$$\text{row} = (0, 0, 0, 0, 0, 0, 1, 1, 3, 3, 4, 4, 4, 5, 5)$$

Matrix-vector multiplication (CSR format)

```
> matrix.dot(vector)
```

$$\begin{bmatrix} 5 & 6 & 9 & & 2 & 2 & & 4 \\ 5 & & & & 7 & & & \\ & & & & & & & \\ & & & & & & & \\ & 5 & & & & 3 & & \\ 6 & & & & & & 1 & 3 \\ & & 5 & & & & 9 & \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 1 \\ 2 \\ 0 \\ 3 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{aligned} \text{data} &= (5, 6, 9, 2, 2, 4, 5, 7, 5, 3, 6, 1, 3, 5, 9) \\ \text{indices} &= (0, 1, 2, 4, 5, 7, 0, 4, 1, 5, 0, 6, 7, 2, 6) \\ \text{indptr} &= (0, 6, 8, 8, 10, 13, 15) \end{aligned}$$

Matrix-vector multiplication (CSR format)

```
> [matrix.dot](http://matrix.dot)(vector)
```

$$\begin{bmatrix} 5 & 6 & 9 & & 2 & 2 & & 4 \\ 5 & & & & 7 & & & \\ & & & & & & & \\ & 5 & & & & 3 & & \\ 6 & & & & & & 1 & 3 \\ & & 5 & & & & 9 & \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 1 \\ 2 \\ 0 \\ 3 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \text{row } 1 \cdot \text{vector} \\ \text{row } 2 \cdot \text{vector} \\ \text{row } 3 \cdot \text{vector} \\ \text{row } 4 \cdot \text{vector} \\ \text{row } 5 \cdot \text{vector} \\ \text{row } 6 \cdot \text{vector} \end{bmatrix}$$

$$\begin{aligned} \text{data} &= (5, 6, 9, 2, 2, 4, 5, 7, 5, 3, 6, 1, 3, 5, 9) \\ \text{indices} &= (0, 1, 2, 4, 5, 7, 0, 4, 1, 5, 0, 6, 7, 2, 6) \\ \text{indptr} &= (0, 6, 8, 8, 10, 13, 15) \end{aligned}$$

Matrix-vector multiplication (CSR format)

$$\begin{bmatrix} 5 & 6 & 9 & & 2 & 2 & & 4 \\ 5 & & & & 7 & & & \\ & & & & & & & \\ & & & & & & & \\ & 5 & & & & 3 & & \\ 6 & & & & & & 1 & 3 \\ & & 5 & & & & 9 & \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 1 \\ 2 \\ 0 \\ 3 \\ 0 \\ 1 \end{bmatrix}$$

$$\text{vector[indices]} = (3, 6, 1, 0, 3, 1, 3, 0, 6, 3, 3, 0, 1, 1, 0)$$

$$\text{output}[i] = \sum \big(\text{data} * \text{vector[indices]}\big)[\text{indptr}[i] : \text{indptr}[i+1]]$$

$$\text{output}[0] = 5 \cdot 3 + 6 \cdot 6 + 9 \cdot 1 + 2 \cdot 0 + 2 \cdot 3 + 4 \cdot 1 = 70$$

Compressed Sparse Column

$$\begin{bmatrix} 5 & 6 & 9 & & 2 & 2 & & 4 \\ 5 & & & & 7 & & & \\ & & & & & & & \\ & & & & & & & \\ & 5 & & & & 3 & & \\ & & & & & & & \\ 6 & & & & & & 1 & 3 \\ & & 5 & & & & 9 & \end{bmatrix}$$

$$\text{shape} = (8, 6)$$

$$\begin{aligned} \text{data} &= (5, 5, 6, 6, 5, 9, 5, 2, 7, 2, 3, 1, 9, 4, 3) \\ \text{indices} &= (0, 1, 4, 0, 3, 0, 5, 0, 1, 0, 3, 4, 5, 0, 4) \\ \text{indptr} &= (0, 3, 5, 7, 7, 9, 11, 13, 15) \end{aligned}$$

Choosing the right format

	COO	CSR	CSC	LIL
Arithmetic		✓	✓	
Row slicing		✓		
Column slicing			✓	
Modification				✓
Loading	✓			

Loading a graph

Data

From	To
12	38
71	98
25	21
12	38
...	...

$\rightarrow$

Adjacency matrix
(COO format)

$$\text{row} = (12, 71, 25, 12, ...)$$
$$\text{col} = (38, 98, 21, 38, ...)$$
$$\text{data} = (1, 1, 1, 1, ...)$$

Coding in Python

```
> adjacency = sparse.coo_matrix((data, (row, col))
> adjacency.sum_duplicates()
```