# What is Sparse Matrix

Edris Safari

Bellevue University, Nebraska U.S.A.

Abstract

In data science and related fields, processing of data and more importantly the ability to process data effectively is of paramount importance. In this paper, we describe one such methodology to process large amount of data. This methodology comes from linear algebra and it is called space matrix.

*Keywords*:

Sparse, matrix

What is Sparse Matrix

**Sparse Matrix**

Data as we know it exists in various formats, in disparate locations, and contain different data for different purposes. The main task of data scientist is to turn the data into a manageable dataset that they can work with. This dataset is shaped into a structure which consist of rows and columns. The columns describe the features that exist in the dataset (i.e. Age, First Name, Last Name, Salary, etc.), and the rows are data or observations for each instance of a feature. Oftentimes, and specially in large datasets, there are missing values which would not contribute to the overall calculations that need to be done to reach a conclusion. Sparse Matrix addresses this situation.

In its simplest definition, sparse matrixes are those matrices which have more elements that are zeros than are non-zero values. For example, the matrix below has total of 16 elements and 10 of them are zeros.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

It becomes evident that for a larger data set with hundreds of columns and thousands to millions of rows, the storage and computation of this dataset become an issue. To alleviate this, we must remove the zeros from the matrix and represent it in such as way so only non-zero values are represented. There are two ways of representing this:

1. Array Representation
2. Linked list representation

The array representation is a two-dimensional array with three rows are described below:

**Row:** The zero-based index of row in the sparse matrix where a non-zero element is located. For example, in the matrix above, rows 0, 1, and 2 have non-zero elements with values of 1,4,1, and 6 respectively. Note that row 2 has two non-zero elements.

**Column:** The zero-based index of column in the sparse matrix where a non-zero element is located. For example, in the matrix above, columns 0,1, and3 have non-zero elements with values 4,1,1 and 6 respectively with column 3 having two non-zero elements with values 1 and 6.

**Value:** The value row shows the value of the non-zero element.

The array representation of the above matrix looks like this:

| Row | 0 | 1 | 2 | 2 |
|---|---|---|---|---|
| Column | 3 | 0 | 1 | 3 |
| Value | 1 | 4 | 1 | 6 |

So the array representation of the sparse matrix can essentially be used to perform the computation needed and also to revert back to the sparse matrix if need be.

The linked list representation is composing of linked list of nodes with four fields-Row, Column, Value with an additional field that links the nodes consecutively. It has a head node that contains the total number of rows, total number of columns, total number of non-zeros in the sparse matrix, and the address of the 1st row. For example, the link list representation of the above sparse matrix looks like this:

Head Node:

| 4 | 4 | 6 | Pointer to 1st row |
|---|---|---|---|

Node1 :

| 0 | 3 | 1 | Pointer to Node2 |
|---|---|---|---|

Node2 :

| | | | |
|---|---|---|---|
| 1 | 0 | 4 | Pointer to Node3 |

Node3 :

| | | | |
|---|---|---|---|
| 2 | 1 | 1 | Pointer to Node4 |

Node4 :

| | | | |
|---|---|---|---|
| 2 | 3 | 6 | No pointer |

## Conclusion

There are many methods to represent sparse matrices. SciPy's 2-D sparse matrix package for numeric data makes the following types available for use in python.

1. csc_matrix: Compressed Sparse Column format
2. csr_matrix: Compressed Sparse Row format
3. bsr_matrix: Block Sparse Row format
4. lil_matrix: List of Lists format
5. dok_matrix: Dictionary of Keys format
6. coo_matrix: COOrdinate format (aka IJV, triplet format)
7. dia_matrix: DIAgonal format

The **coo** and **csr** formats are by far the simplest format as shown above. The application of these transformations is unavoidable when dealing with large datasets with a lot of missing values. It addresses storage and speed of computation issues and allow analysis of meaningful data which will result in more accurate results.

## References

Sparse Matrix | Array representation | Data Structures | Lec-24 | Bhanu Priya - https://www.youtube.com/watch?v=V3TAtTtC4Xs

Sparse Matrix | Linked list representation | Data Structures | Lec-25 | Bhanu Priya - https://www.youtube.com/watch?v=NPnWPeLMo2s

Sparse matrices (scipy.sparse) - https://docs.scipy.org/doc/scipy/reference/sparse.html

Albon, Chris. Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning (p. 4). O'Reilly Media. Kindle Edition.