

Adjacency Matrix Combination and its Types

Daniel Thomas*

Department of Applied Mathematics and Theoretical Physics, Addis Ababa University, Addis Ababa, Ethiopia

Commentary

Received: 03-Jun-2022,
Manuscript No. JSMS-22-66380;
Editor assigned: 06-Jun-2022,
Pre QC No. JSMS-22-66380
(PQ); **Reviewed:** 20-Jun-2022,
QC No. JSMS-22-66380;
Revised: 27- Jun-2022,
Manuscript No. JSMS-22-66380
(A); **Published:** 05-Jul-2022,
DOI: 10.4172/ J Stats Math
Sci.8.5.003.

***For Correspondence:**

Daniel Thomas,
Department of Applied Mathematics
and Theoretical Physics, Addis
Ababa University, Addis Ababa,
Ethiopia
E-mail: danielthomas@98.com

ABOUT THE STUDY

Adjacency matrices are square matrices used to represent finite graphs in graph theory and computer science. The members of the matrix indicate whether two vertices in the graph are adjacent.

The adjacency matrix is a (0,1) matrix with zeros on its diagonal in the special case of a finite simple graph. The matrix is symmetrical if the graph is undirected (i.e. all of its edges are bidirectional). In spectrum graph theory, the relationship between a graph and the Eigen functions and eigenvectors of its adjacency matrix is studied. A graph's eigenvector should be distinguished from its incidence matrix, which is a distinct methodology whose members indicate whether or not vertex-edge pairs are incident, as well as its degree composite, which provides information about each vertex's level. In computer applications for graph manipulation, the adjacency matrix can be used as a data structure for graph representation. The pairwise list is a popular alternative data structure that is also used in this scenario.

Because each item in the adjacency matrix only requires one bit, it may be written in a very compact manner, taking up just 8 bytes for a directed graph and approximately 16 bytes for an undirected graph (using a packed triangular format and only storing the lowest triangular section of the matrix). Although more concise representations are feasible, this method approaches the information-theoretic lower bound for the least amount of bits required to

RRJSMS | Volume 8 | Issue 5 | June, 2022

represent all n -vertex graphs. Fewer bits per byte can be utilized to ensure that all bytes are text characters when storing graphs in text files, such as by utilizing a base format. This compact encourages locality of reference in addition to eliminating wasted space. Adjacency lists, on the other hand, take up less storage space for a big sparse graph because they don't spend space representing edges that don't exist.

The numbers in each member of the matrix are substituted by pointers to edge objects (where edges are present) or null pointers in an alternative version of the adjacency matrix (which, however, requires more space) (when there is no edge). Edge weights can also be directly stored inside the elements of a data structure.

Apart from the space coast, the different data formats also enable various works easily. In an adjacency list, finding all vertices adjacent to specs is as easy as reading the list, and it takes time proportional to the number of peers. An adjacency matrix requires scanning an entire row, which takes longer and is proportional to the number of vertices in the entire graph. Testing whether there is an edge between two given vertices, on the other hand, may be done all at once using an adjacency matrix, whereas, with just an attribute list, it takes time proportional to the minimum degree of the two vertices.

The matrix combination of n types of A has an interesting interpretation if A is the set of edges of the directed or undirected graph G : the element l_j gives the number of (guided or undirected) walks of length n from vertex l to vertex j . n is the distance between vertex l and vertex j if n is the smallest nonnegative integer such that the element l_j of A^n is positive for some l_j . This implies that on an undirected graph G , the number of triangles is equivalent to the trace of A^3 divided by 6. The adjacency matrix can be used and see if the graph is connected or not.