Adjacency Matrices, Incidence Matrices, Database Schemas, and Associative Arrays

Jeremy Kepner and Vijay Gadepally
MIT Lincoln Laboratory, Lexington, MA

Spreadsheets, databases, hash tables and dictionaries; these are the fundamental building blocks of big data storage, retrieval and processing. The MIT Dynamic Distributed Dimensional Data Model (D4M) is an interface to databases and spreadsheets that enble developers to write analytics in the language of linear algebra, significantly reducing the time and effort required. At the core of D4M are associative arrays that allow big data to be represented as large sparse matrices. This sparse matrix D4M schema has been adopted in a number of domains (most notably in the Apache Accumulo community). Linear algebraic graph algorithms work naturally with the D4M schema when such algorithms are cast in terms of edge incidence matrices.

An incidence matrix, **E**, stores each edge in a graph as a row and every vertex as a column. Setting $\mathbf{E}(i,v_1) = -1$ and $\mathbf{E}(i,v_2) = 1$ is one convention to indicate that edge i starts at vertex v_1 and ends at vertex v_2. Incidence matrices naturally represent partite-directed-weighted-multi-hyper graphs (i.e., graphs with many vertex classes, edge direction, weighted edges, multiple edges between vertices, and multiple vertices per edge). Incidence matrices naturally encompass the full richness of data that is found in many data sets (e.g., text documents, bioinformatics, network logs, weblinks, and health records).

In contrast, in an adjacency matrix, **A**, each row and column represent vertices in the graph and setting $\mathbf{A}(v_1,v_2) = 1$ denotes an edge from vertex v_1 to vertex v_2. In the above convention for **E**, the adjacency matrix and the incidence matrix are linked by the formula $\mathbf{A} = |\mathbf{E}^{\prime} < 0|\,|\mathbf{E} > 0|$. In other words, **A** is the cross-correlation of **E**. Any algorithm that can be written using **A** can also be written using **E** via the above formula. However, because **A** is a projection of **E**, information is always lost in constructing **A**, and there are algorithms that can be written using **E** that cannot be constructed using **A**. This talk will describe the interrelationships between adjacency matrices, incidence matrices, database schemas, and associative arrays in the context of specific examples drawn from a number of real-world applications.

This work is sponsored by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, recommendations and conclusions are those of the authors and are not necessarily endorsed by the United States Government