

Course overview

August 2, 2022

Vectors have some similarity with tuples learned in the framework of cartesian product in a discrete mathematics course, as well as relations in tables or relational databases.

Vectors can also be thought of as an aggregate of a number of scalars taken in a specific order. Scalars are often used as a term for a number.

We saw that a vector can be used to store some kind of inventory indicating quantities. For example (Sodium, Potassium, water, Chlorine) is the generic template and a weekly inventory in kgs might look like (7, 4, 11, 21) in the first week. If the consumption in manufacturing processes is (2,3,2,1) in the first week and at the end of the first week new stock is procured as (4,3,4,9) then the net stock at the beginning of week 2 is

$$(7, 4, 11, 21) - (2, 3, 2, 1) + (4, 3, 4, 9) = (7-2+4, 4-3+3, 11-2+4, 21-1+9) = (9, 4, 13, 29)$$

This was an example of vectors representing an aggregate of individual attributes as well as vector addition/subtraction.

We also discussed, briefly, kinds of numbers and their closure or otherwise, under various operations.

- Positive integers are not closed under subtraction, but the set of all integers are.
- Integers are not closed under division

Closure means applying the operation to any two elements in the set must produce an answer that is also in the same set.

Generation of integers by **integer linear combinations** of a set of integers, may or may not generate all possible integers. For example, 30 and 14 cannot generate all integers. Nor can 10 and 30. However, whatever is generated using 10 and 30 can be generated using 10 alone. In vector space terms, this is equivalent to linearly dependent. Whereas 30 and 14

are linearly independent. However, 15, 31 can generate all integers. Neither can on their own. Loosely speaking this is like a basis. These parallels are only loose and miss many attributes of vector spaces. Do not read too much into it.

We also briefly discussed what it means to depend linearly versus depend non-linearly.