

0.1 Linear Independence

We have now described several subspaces of \mathbb{R}^n as spans. Sometimes, the choice of spanning set is not optimal. It may be possible to remove a few vectors from the spanning set, and yet still span the same subspace. How can we deal with such a redundancy? Our next concept captures this phenomenon and gives us language for dealing with it.

Definition 1 (Linear Independence, Linear Dependence). Suppose that $\{v_1, \dots, v_k\}$ is a set of vectors, all chosen from some \mathbb{R}^n . This set of vectors is called *linearly independent* when the only solution to the linear combination of vectors equation

$$x_1v_1 + x_2v_2 + \dots + x_kv_k = 0 \quad (1)$$

is the *trivial solution* $x_1 = x_2 = \dots = x_k = 0$.

The set of vectors is called *linearly dependent* otherwise. That is, we say that the set is linearly dependent when there exists some collection of scalars x_1, x_2, \dots, x_k which are not all zero and are a solution to the equation (1).

Note that if some collection of vectors is linearly dependent, then at least one of the vectors can be written as a linear combination of the others. This is why that vector would be redundant when making a subspace as a span. For example, the equation

$$2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 3 \begin{pmatrix} -2 \\ -1 \end{pmatrix} - 2 \begin{pmatrix} -2 \\ -1/2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

shows that the three vectors on the left form a linearly dependent set. This equation can be rewritten as

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{3}{2} \begin{pmatrix} -2 \\ -1 \end{pmatrix} = \begin{pmatrix} -2 \\ -1/2 \end{pmatrix},$$

which shows that the third vector can be written as a linear combination of the first two. So

$$\text{Span} \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -2 \\ -1 \end{pmatrix}, \begin{pmatrix} -2 \\ -1/2 \end{pmatrix} \right\} = \text{Span} \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -2 \\ -1 \end{pmatrix} \right\}$$

This definition can be translated into our matrix language, too. See the exercises!

There are two standard tests to determine if a set of vectors is linearly dependent or linearly independent. Each has advantages depending on the situation.

A Test for Linear Independence: Casting Out

Given a set of vectors $\{v_1, \dots, v_k\}$, the *Casting Out* algorithm tells us which of the v_i 's we should keep to form a “smallest” linearly independent set which spans the same subspace. It can also be used as a simple test for linear independence or linear dependence.

Step One: Form the matrix C which has the v_i 's as columns:

$$C = \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_k \\ | & | & & | \end{pmatrix}$$

Step Two: Use Gauss-Jordan Elimination to put the matrix C into reduced row echelon form, R . (Actually, we mostly care about which columns are pivot columns and which are free. So you only need to do the forward pass, if you are pressed for time.)

Step Three: From the original set of vectors v_i , keep any of those that correspond to pivot columns, and “cast out” those that come from free columns.

Theorem 2. If the set $\{v_1, \dots, v_k\}$ is linearly dependent, then the matrix C will have at least one free column, and the casting out algorithm will indicate at least one vector to remove.

If the set $\{v_1, \dots, v_k\}$ is linearly independent, then the matrix C will have each of its columns as a pivot column, so the casting out algorithm will tell us to keep all of the v_i 's.

A Test for Linear Independence: Row Algorithm

Given a set of vectors $\{v_1, \dots, v_k\}$, the *Row* algorithm helps us to form a “smallest” linearly independent set $\{w_1, \dots, w_j\}$ which spans the same subspace. It can also be used as a simple test for linear dependence or linear independence.

Step One: Form the matrix A which has the v_i 's as rows:

$$A = \begin{pmatrix} - & v_1 & - \\ & \vdots & \\ - & v_k & - \end{pmatrix}$$

Step Two: Use Gauss-Jordan Elimination to put the matrix A into reduced row echelon form, R .

Step Three: Take the non-zero rows of the RREF, R , and make vectors w_1, \dots, w_j out of them. Often, the matrix R will have rows without pivots (rows of all zeros). In this case, $j < k$, so there are fewer w_i 's than v_i 's.

Theorem 3. If the set $\{v_1, \dots, v_k\}$ is linearly dependent, then the RREF of matrix A will have at least one row of all zeros (a row without a pivot).

If the set $\{v_1, \dots, v_k\}$ is linearly independent, then the RREF of matrix A will have a pivot in each of its rows. Note that the algorithm still usually produces a different set of vectors, but this time $j = k$, that is, there are exactly as many w_i 's as v_i 's.

Which Algorithm?

Take a moment to think over the two algorithms. If all you want is to test for linear independence or linear dependence, it does not matter much which you use.

Can you think of a situation in which you might prefer one algorithm or the other?

Exercises

Exercise 1. Rewrite the definition of a linearly independent set of vectors by translating the linear combination of vectors equation into a matrix-vector equation.

Exercise 2. Rewrite the definition of a linearly dependent set of vectors by translating the linear combination of vectors equation into a matrix-vector equation.

Exercise 3. Make an example of a set of linearly independent vectors in \mathbb{R}^3 . How many vectors can you have?

Exercise 4. Make a collection of 5 different vectors in \mathbb{R}^4 . Use the casting out algorithm to figure out if this set is linearly independent. If it is not, can you find a subset of your example which is linearly independent?

Exercise 5. Use the same example you made in the last exercise, but now use the row algorithm to produce a set of vectors which is linearly independent but has the same span.

Exercise 6. Make an example of a set of three vectors in \mathbb{R}^4 which is linearly independent, or explain why this is not possible.

Exercise 7. Make an example of a set of 5 linearly dependent vectors in \mathbb{R}^5 , or explain why this is not possible.

Challenge 8. Why does the Casting Out algorithm work? Can you explain why Theorem 2 makes sense?

Challenge 9. Why does the row algorithm work? Can you explain why Theorem 3 makes sense?