# **The fundamental theorem of linear algebra.**

We can carry out matrix multiplication in different ways. For example, consider the product :

***A x b***

traditionally we carry out the multiplication as *rows* x *columns* and each element of ***b*** is obtained as a separate dot product:

However we can also consider ***b*** as a *linear combination* of the columns of ***A***:

Since the three columns of ***A*** are linearly independent they are a possible *basis* for the ℝ**3** space; in fact we say that they *span* the ℝ**3** space. By the same token, since ***b*** is necessarily a linear combination of the independent columns of ***A***, then ***b*** belongs to a vector space that we call the *column space* of ***A*** or **C(*A*)**. In the terminology of linear transformations, the *column space* is often called the *range* of ***A***.

We notice here that if instead of *matrix* x *vector* we had a *matrix* x *matrix* multiplication, the product would be obtained in the same way:

***A B C***

with the last 2 columns of the resulting matrix **C** originating from the linear combinations:

and

Thus, every column of ***C*** is in the *column space* of ***A***.

Matrix multiplication is not '*commutative*'; therefore in general

We can view this latter multiplication again as *rows* x *columns*:

***xT A c***

or as a *linear combination* of the *rows*:

Notice that in this case the resulting vector ***c*** is a *row* vector; since the three rows of ***A*** are linearly independent and each contain 3 elements, they also are a possible *basis* for the ℝ**3** space; in fact we say that they also *span* the ℝ**3** space. By the same token, since ***c*** is necessarily a linear combination of the independent rows of ***A***, then ***c*** belongs to a vector space that we call the *row space* of ***A*** or **R(*A*)**.

Notice that since ***A*** is a 3x3 matrix (square matrix) and all the columns/rows are linearly independent, the *dimensions* of the *column* and *row space* are 3: that is both the *column* and the *row* space correspond to the ℝ**3** space. However, we could have products involving a *rectangular* matrix of dimensions *m* x *n* (*m* rows x *n* columns):

***A*** (4x3) ***x*** (3x1) ***b*** (4x1)

The vector ***b*** is of *order* 4(it has 4 elements) and thus it belongs to the ℝ**4** (ℝ**m=4**) space (we write ), but since it is also derived from a linear combination of the columns of ***A***, it also belongs to the *column space* of ***A***. However, to represent ℝ**m=4** we need 4 basis vectors of 4 elements each. Thus, the *column space* of ***A***, while it belongs to ℝ**m** (, it can't represent its totality: in other words not all the vectors in ℝ**m** can be represented as a linear combination of the columns of ***A***: we say that **C(*A*)** is a *subspace* of ℝ**m**. In fact, the *dimensions* of **C(*A*)** are 3, while the *dimensions* of ℝ**m** are 4.

Notice that we could also have a *left side multiplication* as in:

***xT*** (1x4) ***A*** (4x3) ***c*** (1x3)

As we have seen before, the product ***c*** (a *row* vector) is a linear combination of the rows of ***A*,** and thus belongs to the *row space* of ***A***,(**R(*A*)**). We notice here something important: the rows of ***A*** are vectors of *order* 3 (each have 3 elements), and therefore belong to ℝ**3** (ℝ**n=3**). However we have 4 rows, and we need only 3 linearly independent vectors to define ℝ**n**. In fact, any one of the 4 rows of ***A*** can be obtained as a linear combination of the other 3: for example the 4th and 1st rows of ***A*** can be obtained, respectively, as:

A = [2 0 2;3 1 1;1 1 1]

b = [1 2 1]

y = b\*inv(A)

y = b/A

y\*A

A = [3 1 1;1 1 1;1 2 1]

b = [2 0 2]

y = b\*inv(A)

y = b/A

y\*A

Thus, only 3 (any 3) of the 4 row vectors of ***A*** are *linearly independent* and *span* **R(*A*)**, which means that the *dimensions* of **R(*A*)** are 3 (. We can summarize this result in the following table:

|  |  |  |
| --- | --- | --- |
| *subspace* |  | *dimensions* |
| **C(*A*)** | ℝ**m = 4** | **3** |
| **R(*A*)** | ℝ**n = 3** | **3** |

It's clear from this table that the *dimensions* of **C(*A*)** and **R(*A*)** are the same (=3). This is a general result valid for any matrix (*square* or *rectangular*), and the number of these dimensions represents the ***rank*** (***r***) of the matrix ***A***.

**The number of linearly independent columns in a matrix is always the same as the number of linearly independent rows, and it represents the matrix RANK, *r*.**

For example consider the matrix product:

***A*** *(6x4)* ***x*** *(4x1)* ***b*** *(6x1)*

The corresponding table for the *m* x *n* = 6x 4 matrix ***A*** is:

|  |  |  |
| --- | --- | --- |
| *subspace* |  | *dimensions* = RANK |
| **C(*A*)** | **Rm = 6** | **2** |
| **R(*A*)** | **Rn = 4** | **2** |

Only 2 of the 4 columns and 2 of the 6 rows of ***A*** are linearly independent.For example, columns 3 and 4 can be obtained as a linear combination of column 1 and 2:

and

A = [-6 2 0 4;-15 5 0 10;-7 3 1 3;-1 1 1 -1;1 1 2 -4;6 0 3 -9]

x = A(:,1:2)\A(:,3), x = A(:,1:2)\A(:,4)

Therefore, the *dimensions* of **C(*A*)** and **R(*A*)** are the same (= 2), and rank(***A***) = dim(**C(*A*)**) = dim(**R(*A*)**) = 2.

rank(A) rank of ***A***

CA = orth(A) orthogonal basis for the column space (range) of ***A***

RA = orth(A') orthogonal basis for the row space of ***A***

Now we ask the question: is there a more general strategy to find out if the columns of ***A*** are linearly independent? Since columns 3 and 4 can be obtained as a linear combination of columns 1 and 2, if we subtract those linear combinations from columns 3 and 4 we obtain a vector of all 0's. In fact, we can easily set up several linear combinations of the columns of ***A*** that sum up to 0. For example summing -0.5 of column 1 and -1.5 of column 2 to column 3, and taking none of column 4:

or taking 0 of the 3rd column and a linear combination of column 1 and 2 for column 4:

or even:

Thus, we can easily identify at least 3 vectors (different from the obvious **0** vector) that produce linear combinations of the columns of ***A*** that sum up to the **0** vector. In fact, if there are one or more solutions (different from the **0** vector) to the matrix equation (in other words, if there is any combination of the columns of ***A*** that sums up to **0**) it means that the columns of ***A*** are not all linearly independent.

However, it is important to notice that only some of all possible solutions to the matrix equation may be linearly independent themselves. For example:

All the possible ***x*** vectors that solve form a *vector space* called the *null space* of ***A*** (**N(*A*)**). In the language of *linear transformations* this is also called the *kernel* of ***A***. We notice the following:

1. Since any ***x*** vector has the same number of elements of any *row* vector of ***A***, then **N(*A*)** is a *subspace* of **Rn** :

2. The dimensions of **Rn** are *n*. Some of these dimensions were already taken by the *row space* of ***A***, which also belongs to **Rn**. Since dim(**R(*A*))** =*rank*=*r*, then the *dimensions* of the *null space* are *n-r* :

For example, for the ***A*** matrix (*m = 6* rows x *n = 4* columns) shown above, since *rank* = *r**=* 2 we have . This means that only 2 basis vectors are sufficient to represent the *null space* of ***A***. A possible *basis* for **N(*A*)** is represented by the two ***x*** vectors already found:

Notice that since the dot product of each *row* vector of ***A*** times any ***x*** vector in the *null space* is equal to 0 to fulfill the matrix equation , then any vector in the *null space* is perpendicular to any vector in the *row space*. For example:

N\_a = null(A,'r') % rational form

A\*N\_a

In this case the *basis*  for the *null space* does not consist of *unit* or *orthogonal* vectors:

norm(N\_a(:,1))

norm(N\_a(:,2))

N\_a'\*N\_a

However, we can also use the *null* function to choose orthonormal vectors:

N\_a = null(A)

norm(N\_a(:,1))

norm(N\_a(:,2))

N\_a'\*N\_a

A\*N\_a

For this reason we say that **R(*A*)** and **N(*A*)** are *orthogonal complement subspaces* of**Rn**, and their dimensions sum up to *dim*(**Rn**) = *n*.

We can ask the question: are there also one (or more) vectors (different from the **0** vector) that '*left*' multiplying ***A*** will give the **0** vector? This is the same as asking if if there are one or more solutions to the matrix equation (or, in other words, if there is any combinations of the rows of ***A*** that sums up to **0**):

***xT*** (*1x6*) ***A*** (*m*=*6 x n=4*)

If such solutions exist, they belong to a space that we call the *left nullspace* of ***A***, **LN(*A*)**. The basis vectors of **LN(*A*)** are vectors of *order 6*, and thus they belong to **Rm = 6**, like the vectors in the *column space* of ***A***, **C(*A*)**.

We recall here that the transpose (or inverse) of a product is the product of the transpose (or inverse) of the elements in reversed order. For example:

Thus, by taking the *transpose* of both sides we find that the matrix equation is completely equivalent to the matrix equation :

***AT*** (*4x6*) ***x***(*6x1*)

Thus, the *left null space* **LN(*A*)** of ***A*** is the same as the *null space* **N(*AT*)** of ***AT***.

Since taking the transpose does not change the rank *r* of ***A***, the dimensions of the *null space* of ***AT*** are 6 – *r* = 4, or equivalently, the dimensions of the left *null space* of ***A*** are *m* – *r* = 4. This result makes sense: in fact, since the *dimensions* of **Rm = 6** are 6, and 2 dimensions are already taken by the *column space* of ***A***, the remaining 4 dimensions of **Rm**, form the *basis* for **LN(*A*),** which isa *subspace* of **Rm**.

LN\_a = null(A')

norm(LN\_a(:,1))

norm(LN\_a(:,2))

norm(LN\_a(:,3))

norm(LN\_a(:,4))

LN\_a’\*LN\_a

LN\_a'\*A

Furthermore, since the dot product of each of the vectors in **LN(*A*)** with each of the columns of ***A*** is equal to 0, the four *basis vector*s of **LN(*A*)** are also orthogonalto any vector in the *column space* of ***A***. It follows that **C(*A*)** and **LN(*A*)** are *orthogonal complement subspaces of* **Rm**.

We have now completed the analysis of the ***four spaces***of a matrix: the existence and property of these four spaces represents the ***fundamental theorem of linear algebra***. This theorem applies to any matrix of any dimension (square or rectangular).

The following table summarizes the theorem for a general matrix ***A*** of dimensions *m* x *n* (*m* rows x *n* columns):

|  |  |  |  |
| --- | --- | --- | --- |
| *subspace* |  | *dimensions* | *orthogonal to* |
| **C(*A*)** | **Rm** | *r* | **LN(*A*) = N(*AT*)** |
| **R(*A*)** | **Rn** | *r* | **N(*A*)** |
| **N(*A*)** | **Rn** | *n-r* | **R(*A*)** |
| **LN(*A*) = N(*AT*)** | **Rm** | *m-r* | **C(*A*)** |

The *four spaces* of a matrix ***A*** can also be represented with the following diagram:



Every matrix product ***Ax*** *=* ***b***takes a vector ***x*** from **Rn** to a vector ***b*** in **Rm**. Any vector ***x*** can be thought ofas the sumof two vectors ***x*R(*A*)** and ***x*N(*A*)**, which are the *projections* of ***x*** onto the two subspaces **R(*A*)** and **N(*A*)** of **Rn**. If ***x*N(*A*) ≠** **0**, that component is taken to **0** in **Rm**. As a consequence ***Ax*R*(A)*** *=* ***b*** is the same as ***Ax*** *=* ***b***.

An important corollary of the *fundamental theorem* is that if ***x*** has a component in the *null space* of ***A***, that component will be converted to **0**.

**Once you go to 0 there is no way to come back!**

Any operation carried out on 0 will always give 0. This means that if ***A*** is a *square* matrix and if the *null space* of ***A*** is not empty, it is not possible to find another matrix that will yield ***x*** (if ***x*** has a component in **N(*A*)**) acting on ***b***: this is the same as saying that ***A*** is NOT invertible (there is no ***A-1*** matrix), and we say that ***A*** is ***SINGULAR***.

How do we know if the *null space* of a matrix ***A*** is not empty? We recall here that the dimensions of **N(*A*)** are *n-r*, and that the *rank r* of a matrix is the number of linearly independent columns or rows. Thus, if the *rank* of a square matrix is less than *n* (that is, if some of the columns are not linearly independent), it means that there are *n-r* vectors in the *null space*, and therefore the matrix is *singular*. By the same token, if all the columns of a square matrix ***A*** are linearly independent, it means that the *null space* of ***A*** is empty, and an inverse ***A-1*** of ***A*** exists.

Identification of the 4 *subspaces* of a matrix is the foundation of the matrix factorization known as ***Singular Value Decomposition*** or simply **SVD** (CHAPTER 11)**,** which finds several important applications in biology. This factorization is used internally in MATLAB each time we invoke the functions:

CA = orth(A)

RA = orth(A')

N\_a = null(A)

LN\_a = null(A')

r = rank(A);

As a short hand for the complete SVD factorization:

[U,S,V] = svd(A)

**PRACTICE**



**1.** Consider the matrix ***U***:

Its *column space* does not coincide with **R3** because only 2 of the columns are independent (for example C1 = C2+C3). Thus, the *basis* for **C(*U*)** is given by only two vectors of **R3**, and **C(*U*)** is a *subspace* of **R3**. It is immediately obvious that this *subspace* is a **plane** through the origin of **R3** containing the two vectors ***u1*** and ***u2***:

It's important to understand that while **C(*U*)** is itself a 2-dimensional space, its vectors belong to **R3** and not to **R2**, whose vectors would have only 2 components. Furthermore, any *space* or *subspace* must also always contain the 0 vector (the *origin*). Thus, a plane that does not go through the origin (also defined as an *affine plane*) is NOT a subspace of **R3**.

Find the *basis* for the *left* *null space* of ***U***. How would you represent this subspace of **R3**?

**2.** The traditional way of multiplying matrices is *rows* x *columns*. However there is also a way of doing it *columns* x *rows*. We recall here that the product of a column times a row is NOT a dot (***inner***) product ( = a *scalar*), but it is a matrix. For example:

As we mentioned, this type of product is also called an ***outer***product (or ***dyadic*** product, and represented with the symbol ).

Based on this information show how you would calculate the following product using a *columns* x *rows* multiplication:

In particular, show the individual *column* x *row* products that are added up to form the final matrix.

**3.** Consider the linear transformation represented by the matrix:

A = [-6 2 0 4;-15 5 0 10;-7 3 1 3;...

-1 1 1 -1;1 1 2 -4;6 0 3 -9];

Let's identify a possible choice of independent columns. We can accomplish this by doing Gaussian elimination on ***A*** until we achieve the *row reduced echelon form* of the matrix.

[R,jb] = rref(A)

Notice the following:

* The identity submatrix with 1's on the diagonal shows the *rank =* 2, andindicate there are 2 independent columns (column 1 and 2).
* The *indeces* of the independent columns are also listed in the ***jb*** vector.
* The columns outside the identity submatrix are the dependent columns: the elements of these columns adjacent to the identity submatrix show the linear combinations of the independent columns that generate the dependent columns. (e.g.: column 3 = 0.5 x column 1 + 1.5 x column 2).
* The rows of 0's below the identity submatrix indicate that there is a total of 4 linearly dependent rows, but do not give any information about the identity (*indeces*) of these rows.

Using this example as a guide:

a. Identify one possible choice of the independent rows.

b. Find a basis for the four subspaces of ***A***.

**4.** Vector ***u1*** = [3 7 5] is the 1st basis vector of an orthogonal basis ***U*** in **R3**. Using what you have learnt so far about the fundamental spaces of a matrix and about rotations:

a. Find two other possible vectors of an orthogonal basis ***U***.

b. Display the standard basis ***V*** and the new basis ***U***.

c. Find the transformation matrix ***Q*** that would allow the representation of any vector in the rotated frame ***U***.

**5.** Consider the matrix equation ***Ax = b***:

We recall that this equation is the same as:

-6x1 + 2x2 + 0x3 + 4x4 = 14

-15x1 + 5x2 + 0x3 + 10x4 = 35

-7x1 + 3x2 + 1x3 + 3x4 = 14

-1x1 + 1x2 + 1x3 - 1x4 = 0

1x1 + 1x2 + 2x3 - 4x4 = -7

6x1 + 0x2 + 3x3 - 9x4 = -21

This is a typical example of an overdetermined system of linear equations because there are more equations than unknowns (as opposed to an underdetermined system of linear equations in which there are more unknowns than equations). In cases like this one it can happen that an exact solution does not exist (e.g., ***b*** is not in the *column space* of ***A***), and only a *least-squares* solution is possible (see CHAPTER 6). But in this particular case matrix ***A*** is ***rank deficient*** (*r* < *n*), which produces an interesting situation:

1. As it turns out, one possible exact solution for ***x*** is [1 2 3 4]. Verify this is true.
2. Use mldivide to find another possible solution.
3. Use Gauss-Jordan elimination to find another possible solution.
4. Using the *null space* of ***A*** can you find two other exact solutions?
5. How many exact solutions exist for this system of linear equations?
6. Is it possible to find a solution that would have the shortest possible length (=smallest 2-norm)?