

# Linear Algebra

Grant Smith

Spring 2022

## 1 The Four Subspaces

First of all, we know that there is a row space and a column space. We also know that the column space of  $A$  is the row space of  $A^T$  and the row space of  $A$  is the column space of  $A^T$ . Next are questions about the null space. But first, I want to discuss how Gaussian Elimination changes (or doesn't change) the row space.

### 1.1 Gaussian Elimination

Gaussian Elimination does not change the row space. We'll prove this in a couple steps. First of all, we have to prove that Gaussian Elimination doesn't generate rows outside of the original row space, and then we have to prove that it doesn't remove anything from the original space. First, we show that it doesn't add anything to the space. This is because whenever Gaussian Elimination is being used, the rows are just being added and scaled. Which means all of the row vectors in the end are linear combinations of the original row vectors we had. So we know that it didn't add any row vectors that weren't there previously. Next we need to prove that it doesn't take anything away. And right now, I don't know how to prove this.

Also, just to be clear, Gaussian Elimination does indeed change the column space, but it doesn't change its dimension. We'll prove the part about it not changing the dimension, but the best way to see that it does change the column space is to think about a matrix:

$$\begin{bmatrix} 1 & 5 \\ 2 & 10 \end{bmatrix}$$

And with Gaussian Elimination, that is:

$$\begin{bmatrix} 1 & 5 \\ 0 & 0 \end{bmatrix}$$

Which certainly has a different column space, but the column space has the same dimension, which, again, we will show soon.

### 1.2 The Row Space and the Column Space dimensions are equal

First of all, let's start with an  $m \times n$  matrix called  $A$ .

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & & a_{m,n} \end{bmatrix}$$

Notice that the dimension of the row space and the column space are both at most  $\min(m, n)$ . This is because if  $m$  is bigger, then we have  $m$  rows, but they are only  $n$  wide, so they can't span a space with higher dimension than  $n$ . And for the columns, we only have  $n$  columns, so they can't span a space of dimension  $> n$ . The same argument would work for if  $n$  was bigger. This is nice because it sets a ceiling on the two dimensions, and they are the same for the rows and columns. So this is just a hint and a nice fact, but it doesn't prove that they're the same.

Next, let's say the column space is  $b$  dimensional, and  $b \leq \min(m, n)$ . Then there is a basis of  $b$  column vectors (each of which is  $m$ -dimensional) that can be used to create our matrix. So let's arrange those into a matrix  $B$ . And since those

column vectors span the row space of  $A$ , we can write  $A = BC$  where the columns of  $C$  are the linear weightings of  $B$ 's columns that make  $A$ 's columns.

$$A = \begin{bmatrix} a_{1,1} & \dots & a_{1,b} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{b,1} & \dots & a_{b,b} & \dots & a_{b,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m,1} & \dots & a_{m,b} & \dots & a_{m,n} \end{bmatrix} = \begin{bmatrix} b_{1,1} & \dots & b_{1,b} \\ \vdots & \ddots & \vdots \\ b_{b,1} & \dots & b_{b,b} \\ \vdots & \ddots & \vdots \\ b_{m,1} & \dots & b_{m,b} \end{bmatrix} \begin{bmatrix} c_{1,1} & \dots & c_{1,b} & \dots & c_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{b,1} & \dots & c_{b,b} & \dots & c_{b,n} \end{bmatrix}$$

But we could also view this as a statement about  $C$ 's rows spanning  $A$ 's rows. So that means that  $\dim(\text{row}(A)) \leq \dim(\text{col}A)$

Let's do a similar argument, but by spanning the row space initially instead of the column space. let's say the row space is  $e$  dimensional. Then we can write the rows of  $A$  as linear combinations of  $e$  row vectors, and let's put those into a matrix  $E$ . Now we can say that  $A = DE$  where  $D$  contains the weightings of  $E$ 's row vectors that make  $A$ 's row vectors. Thus, we have

$$A = \begin{bmatrix} a_{1,1} & \dots & a_{1,e} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{e,1} & \dots & a_{e,e} & \dots & a_{e,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m,1} & \dots & a_{m,e} & \dots & a_{m,n} \end{bmatrix} = \begin{bmatrix} d_{1,1} & \dots & d_{1,e} \\ \vdots & \ddots & \vdots \\ d_{e,1} & \dots & d_{e,e} \\ \vdots & \ddots & \vdots \\ d_{m,1} & \dots & d_{m,e} \end{bmatrix} \begin{bmatrix} e_{1,1} & \dots & e_{1,e} & \dots & e_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ e_{e,1} & \dots & e_{e,e} & \dots & e_{e,n} \end{bmatrix}$$

Which means that  $A$ 's columns can be spanned by  $e$  or fewer columns, where  $e$  was the dimension of the row space. This means that  $\dim(\text{row}(A)) \geq \dim(\text{col}A)$  but now we have both sides of the inequality, so the two dimensions are equal.

And this means that  $e = b$ , so the two factorizations above are the same dimension. They don't have to be the exact same because there are many available bases for the two spaces, but they are at least the same dimension.

Thanks to Sean Owen on Quora for this explanation.

### 1.3 Back To Gauss Jordan

Now that we know the dimension of the row space is equal to the dimension of the column space, We can combine that with what we know about Gaussian Elimination not changing the row space: if the row space dimension and the column space dimension of any matrix are equal, and Gaussian Elimination does not change the row space at all (let alone its dimension), then Gaussian Elimination doesn't change the column space dimension.

### 1.4 Back to the end of the Row Space and The Column Space Being Equal Section

At this point, we could make one of those two matrices orthogonal or orthonormal. That would be nice because orthogonal and orthonormal bases are great. The problem is we can only really do one at a time. For example, if we make  $C(or E)$  orthonormal (and there are infinite options of orthonormal bases), then that specifies the weightings on  $B(or D)$ . And vice versa if we chose to make the left matrix orthonormal, then that would change the numbers in the right matrix.

But! If you choose your orthonormal vectors just right on either side (i.e. choose the right one of the infinite options), then you can allow the vectors on the other side to be orthogonal as well, and you can factor out their sizes to make them orthonormal. And this is the SVD. If we choose just the right orthonormal basis of the row space, then we can actually make the column space orthonormal too. And same for the other way around. We'll talk more about how to do this, but for now just know that there is something coming.