Row and column operations

It is often very useful to apply row and column operations to a matrix. Let us list what operations we're going to be using.

We'll illustrate these using the example matrix $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$.

Row/column operations

• Adding a multiple of one row to another, e.g., $r_1 \mapsto r_1 + \alpha r_2$: $\begin{pmatrix} 1+4\alpha & 2+5\alpha & 3+6\alpha \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$.

• Scaling a row by some non-zero factor, e.g., $r_2 \mapsto \lambda r_2$: $\begin{pmatrix} 1 & 2 & 3 \\ 4\lambda & 5\lambda & 6\lambda \\ 7 & 8 & 9 \end{pmatrix}$.

• Swapping two rows, e.g., $r_2 \leftrightarrow r_3$: $\begin{pmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 4 & 5 & 6 \end{pmatrix}$.

There are equivalent operations for columns, resulting in changes such as:

$$\begin{pmatrix} 1 & 2+\alpha & 3 \\ 4 & 5+4\alpha & 6 \\ 7 & 8+7\alpha & 9 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2\lambda & 3 \\ 4 & 5\lambda & 6 \\ 7 & 8\lambda & 9 \end{pmatrix}, \quad \begin{pmatrix} 1 & 3 & 2 \\ 4 & 6 & 5 \\ 7 & 9 & 8 \end{pmatrix}$$

Row (and column) operations can make a matrix 'nice'

A matrix has a row-reduced form (and a column-reduced form, but let's study rows), which we obtain by row operations to make it as simple as possible. This form is such that:

• each non-zero row starts with some number of 0s, then an initial 1, then other entries;
• in each column which features some row's initial 1, the other entries are 0 (since with row operations we may use that 1 to clear the rest of the column);
• columns which do not feature some row's initial 1 may have other exciting entries (since we cannot clear them).
• the resulting non-zero rows are linearly independent (as the positions of the initial 1s shows)

There is an algorithm to transform a matrix into row-reduced form.

1. Start with the left-hand column. If it is entirely 0, we move to the next column. Otherwise, pick some non-zero entry x. Scale the row x is in by $1/x$ and swap it to the top, giving us a 1 as the first non-zero entry in the top row. Then add suitable multiples of this row to the other rows to make the rest of the column x was in 0.

   So now the first non-zero entry in the top row is 1, and all entries beneath it are 0. This row is now locked in place: we might add things to it, but we'll never scale or swap it again.

2. Move to the next column. Ignoring the locked top row, are there any non-zero entries in this column? If not, move on. Otherwise, pick some non-zero entry y. Scale the row y is in by $1/y$ and swap it to the second row, giving us a 1 as the first non-zero entry in the second row. (Note that it occurs in a column to the right of the 1 in the top row.)

   Then add suitable multiples of this second row to the other rows (including the top row) to make the rest of the column y was in 0. The second row is then locked in place.

3. Move to the next column, and repeat.

We illustrate it here by an example. Let $A = \begin{pmatrix} 0 & 2 & 6 & 0 & 0 \\ 3 & 0 & 6 & 0 & 6 \\ 2 & 0 & 4 & 0 & 7 \\ 1 & 1 & 5 & 0 & 6 \end{pmatrix}$

We'll pick the 3 in the first column. We scale row 2 by $\frac{1}{3}$ and swap the first and second rows.

$$\xrightarrow{r_2 \mapsto \frac{1}{3} r_2} \begin{pmatrix} 0 & 2 & 6 & 0 & 0 \\ 1 & 0 & 2 & 0 & 2 \\ 2 & 0 & 4 & 0 & 7 \\ 1 & 1 & 5 & 0 & 6 \end{pmatrix} \xrightarrow{r_1 \leftrightarrow r_2} \begin{pmatrix} 1 & 0 & 2 & 0 & 2 \\ 0 & 2 & 6 & 0 & 0 \\ 2 & 0 & 4 & 0 & 7 \\ 1 & 1 & 5 & 0 & 6 \end{pmatrix}$$

We subtract multiples of row 1 from rows 3 and 4

$$\xrightarrow{r_3 \mapsto r_3 - 2r_1} \begin{pmatrix} 1 & 0 & 2 & 0 & 2 \\ 0 & 2 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 1 & 1 & 5 & 0 & 6 \end{pmatrix} \xrightarrow{r_4 \mapsto r_4 - r_1} \begin{pmatrix} 1 & 0 & 2 & 0 & 2 \\ 0 & 2 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 1 & 3 & 0 & 4 \end{pmatrix}$$

We like the 1 in the second column, fourth row, so we swap rows 2 and 4, then subtract twice row 2 from row 4. We then skip over the third column (as its only non-zero entries are locked), and skip over the empty fourth column.

$$\xrightarrow{r_2 \leftrightarrow r_4} \begin{pmatrix} 1 & 0 & 2 & 0 & 2 \\ 0 & 1 & 3 & 0 & 4 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 2 & 6 & 0 & 0 \end{pmatrix} \xrightarrow{r_4 \mapsto r_4 - 2r_2} \begin{pmatrix} 1 & 0 & 2 & 0 & 2 \\ 0 & 1 & 3 & 0 & 4 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & -8 \end{pmatrix}$$

We decide (fairly arbitrarily) to pick the -8. We scale row 4 and swap it with row 3.

$$\xrightarrow{r_4 \mapsto -\frac{1}{8} r_4} \begin{pmatrix} 1 & 0 & 2 & 0 & 2 \\ 0 & 1 & 3 & 0 & 4 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \xrightarrow{r_3 \leftrightarrow r_4} \begin{pmatrix} 1 & 0 & 2 & 0 & 2 \\ 0 & 1 & 3 & 0 & 4 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

We clear the final column, using the third row.

$$\xrightarrow[r_4 \mapsto r_4 - 3r_3]{\substack{r_1 \mapsto r_1 - 2r_3 \\ r_2 \mapsto r_2 - 4r_3}} \begin{pmatrix} 1 & 0 & 2 & 0 & 0 \\ 0 & 1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

We cannot clear the third column, as there is no available initial 1 in the lower rows.

There is a similar procedure for column reduction, doing exactly the same thing, only to the columns. If we use both row *and* column operations, then we can tidy the matrix even more. We could continue the example above as follows.

$$\xrightarrow[c_3 \mapsto c_3 - 3c_2]{c_3 \mapsto c_3 - 2c_1} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \xrightarrow{c_3 \leftrightarrow c_5} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

This is the fully-reduced form of A. Any matrix can be put into a form this simple, providing we're allowed (see next page) to use both row and column operations.

Given that, it is clear that an invertible (square) matrix has fully-reduced form equal to the identity matrix I. However, such a matrix in fact has its row-reduced (or column-reduced) form equal to I. In our row operations, we cannot end up with a row entirely of 0s, so each row must have an initial 1 in it, and no two initial 1s can appear in the same column, so we must have an initial 1 in each diagonal place. Then their columns are cleared, giving us I.

Column operations preserve the column span and image of a matrix

So, row and column operations can make a matrix nicer, but what do they do to the effect of A as a linear map?

Example. Let $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$.

To find the image, consider

$$A\mathbf{x} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x+2y+3z \\ 4x+5y+6z \\ 7x+8y+9z \end{pmatrix} = x\begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix} + y\begin{pmatrix} 2 \\ 5 \\ 8 \end{pmatrix} + z\begin{pmatrix} 3 \\ 6 \\ 9 \end{pmatrix}$$

As $\mathbf{x}$ varies over all of $\mathbb{R}^3$, the image consists of all, and only, combinations of the vectors formed by the columns of A.

In other words: *the image of a matrix is the span of its columns.*

Let's try a column operation: say, subtract column 1 from column 2. We get:

$$A'\mathbf{x} = \begin{pmatrix} 1 & 1 & 3 \\ 4 & 1 & 6 \\ 7 & 1 & 9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = x\begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix} + y\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + z\begin{pmatrix} 3 \\ 6 \\ 9 \end{pmatrix} = (x-y)\begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix} + y\begin{pmatrix} 2 \\ 5 \\ 8 \end{pmatrix} + z\begin{pmatrix} 3 \\ 6 \\ 9 \end{pmatrix}$$

Hence, as $\mathbf{x}$ varies, the image is still the span of the *original* columns.

In other words: *column operations preserve the column span.*

And therefore: *column operations preserve the image of the matrix.*

So, to find the image of a matrix, we can column-reduce it, as follows.

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \xrightarrow[c_3 \mapsto c_3 - 3c_1]{c_2 \mapsto c_2 - 2c_1} \begin{pmatrix} 1 & 0 & 0 \\ 4 & -3 & -6 \\ 7 & -6 & -12 \end{pmatrix} \xrightarrow{c_2 \mapsto -\frac{1}{3}c_2} \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & -6 \\ 7 & 2 & -12 \end{pmatrix} \xrightarrow[c_3 \mapsto c_3 + 6c_2]{c_1 \mapsto c_1 - 4c_2} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 2 & 0 \end{pmatrix}$$

Hence the image has basis $\left\{ \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \right\}$.

Warning. Row operations destroy the image!

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \xrightarrow[r_3 \mapsto r_3 - 7r_1]{r_2 \mapsto r_2 - 4r_1} \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -12 \end{pmatrix}$$

And the image certainly doesn't contain $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$.

So, if we care about the image of our matrix A, we must be careful only to use column operations.

Row operations preserve the row span and kernel of a matrix

To find the kernel of the same original matrix A, we want to solve

$$A\mathbf{x} = \begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix}\begin{pmatrix}x\\ y\\ z\end{pmatrix} = \begin{pmatrix}x+2y+3z\\ 4x+5y+6z\\ 7x+8y+9z\end{pmatrix} = \begin{pmatrix}0\\ 0\\ 0\end{pmatrix}$$

We could solve these three equations simultaneously, but the standard manoeuvres there, such as subtracting one equation from another, are clearly equivalent to row operations.

And this makes sense. We are trying to find the vectors $\mathbf{x}$ which dot (in the sense of the scalar product) with each row of the matrix to give 0. If a vector dots with each row to give 0, then it dots with any row combination to give 0. So we are seeking the vectors $\mathbf{x}$ which are orthogonal to the whole of the *span* of the rows.

In other words: *the kernel is the set of vectors orthogonal to the row span.*

And, just as with the columns before: *row operations preserve the row span.*

And therefore: *row operations preserve the kernel of the matrix.*

So, to find the kernel of a matrix, we can fully row-reduce it, as follows.

$$\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix} \xrightarrow[r_3\mapsto r_3-7r_1]{r_2\mapsto r_2-4r_1} \begin{pmatrix}1 & 2 & 3\\ 0 & -3 & -6\\ 0 & -6 & -12\end{pmatrix} \xrightarrow{r_2\mapsto -\frac{1}{3}r_2} \begin{pmatrix}1 & 2 & 3\\ 0 & 1 & 2\\ 0 & -6 & -12\end{pmatrix} \xrightarrow[r_3\mapsto r_3+6r_2]{r_1\mapsto r_1-2r_2} \begin{pmatrix}1 & 0 & -1\\ 0 & 1 & 2\\ 0 & 0 & 0\end{pmatrix}$$

Hence for the kernel, we need vectors $\mathbf{x}$ such that $x = z$ and $y = -2z$.

I.e., a basis for the kernel is $\left\{\begin{pmatrix}1\\ -2\\ 1\end{pmatrix}\right\}$.

Warning. Column operations destroy the kernel!

This should be even more clear than for the image: when solving simultaneous equations, we can't just go and move coefficients around within each equation.

So, if we care about the kernel of our matrix A, we must be careful only to use row operations.

However, if we care only about the rank or nullity of A, then we can perform full reduction.

Elementary matrices

Let's take an example. Let

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \quad E = \begin{pmatrix} 1 & \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

where $\lambda \neq 0$. We get the following behaviour.

1. Applying E on the left and the right of A.

$$EA = \begin{pmatrix} 1+4\alpha & 2+5\alpha & 3+6\alpha \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \quad AE = \begin{pmatrix} 1 & \alpha+2 & 3 \\ 4 & 4\alpha+5 & 6 \\ 7 & 7\alpha+8 & 9 \end{pmatrix}$$

   I.e., multiplying on the left by E adds α times row 2 to row 1, while multiplying on the right by E adds α times column 1 to column 2.

   More generally, if $i \neq j$, the matrix which is I plus an α in the (i, j) place is such that multiplying on the left by adds α times row j to row i, while multiplying on the right adds α times column i to column j.

2. Applying M on the left and the right of A.

$$MA = \begin{pmatrix} 1 & 2 & 3 \\ 4\lambda & 5\lambda & 6\lambda \\ 7 & 8 & 9 \end{pmatrix}, \quad AM = \begin{pmatrix} 1 & 2\lambda & 3 \\ 4 & 5\lambda & 6 \\ 7 & 8\lambda & 9 \end{pmatrix}$$

   I.e., multiplying on the left by M scales row 2 by λ, while multiplying on the right by M scales column 2 by λ.

   More generally, the matrix which is I except for $\lambda \neq 0$ in the (i, i) place is such that multiplying on the left scales row i by λ, while multiplying on the right scales column i by λ.

3. Applying S on the left and the right of A.

$$SA = \begin{pmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 4 & 5 & 6 \end{pmatrix}, \quad AS = \begin{pmatrix} 1 & 3 & 2 \\ 4 & 6 & 5 \\ 7 & 9 & 8 \end{pmatrix}$$

   I.e., multiplying on the left by S swaps rows 2 and 3, while multiplying on the right by S swaps columns 2 and 3.

   More generally, the matrix which is I except for having 0 in the (i, i) and (j, j) places, and 1 in the (i, j) and (j, i) places, is such that multiplying on the left swaps rows i and j, while multiplying on the right swaps columns i and j.

Therefore: *row operations can be achieved by matrix multiplication on the left, and column operations can be achieved by matrix multiplication on the right.*

This further explains the behaviour on pages 3 and 4.

Suppose we have the matrix A, and perform column operations by multiplying on the right by matrices C_1, C_2, C_3. We get $AC_1C_2C_3$. The C_i are invertible, so their image is all of $\mathbb{R}^n$, and then we apply A as the final step. Hence the overall image is that of A itself.

Suppose we instead perform row operations by multiplying on the left by matrices R_1, R_2, R_3. We get $R_3R_2R_1A$. Any vector in the kernel of A gets sent to 0 by the initial application of A, and stays there. Any vector not sent to 0 by A doesn't later get sent to 0 by the R_i, since they are invertible. Hence the overall kernel is that of A itself.

It also explains how the "wrong" operations break things: the C_i move other things into the way of A's kernel, and the R_i move A's image around.