

Revision Notes on Linear Algebra for Undergraduate Engineers

Pete Bunch

Lent Term 2012

1 Introduction

A matrix is more than just a grid full of numbers. Whatever sort of engineering you specialise in, a basic grounding in linear algebra is likely to be useful. Lets look at some examples of how matrix equations arise.

Add examples from structures (truss), electrical (resistor grid), mechanics (kinematics), fluids (some sort of pressure, height, velocity-square problem)

All of these problems lead us to the same equation,

$$\mathbf{Ax} = \mathbf{b} \quad (1)$$

We can think of \mathbf{x} as a position vectors in an “input space” or “solution space” and \mathbf{b} as the corresponding position vector in the “output space” or “constraint space” respectively. The coordinates in these spaces are just the values of the individual components of \mathbf{x} and \mathbf{b} . \mathbf{A} is just the linear transformation that takes us from one space to the other.

2 Fundamental Matrix Sub-Spaces

Consider an $m \times n$ matrix \mathbf{A} , which maps a point \mathbf{x} in the “input space” to \mathbf{y} in the “output space”. i.e.

$$\mathbf{y} = \mathbf{Ax} \quad (2)$$

What can we say about the input and output spaces? Firstly, \mathbf{x} must be $n \times 1$ so the input space is n -dimensional (nD). \mathbf{y} must be $m \times 1$ so the output space is m -dimensional (mD).

2.1 Output Side

Let’s consider the output space. We can write \mathbf{A} as a set of column vectors.

$$\mathbf{A} = [\mathbf{c}_1 \quad \mathbf{c}_2 \quad \dots \quad \mathbf{c}_n] \quad (3)$$

Now when we multiply \mathbf{A} by \mathbf{x} , we can write it as,

$$\mathbf{y} = \mathbf{Ax} = x_1\mathbf{c}_1 + x_2\mathbf{c}_2 + \dots + x_n\mathbf{c}_n. \quad (4)$$

So the output, \mathbf{y} , is just a weighted some of the columns of \mathbf{A} . This set of column vectors defines a new vector subspace, the “column space”, which contains all the valid \mathbf{y} ’s for the equation $\mathbf{y} = \mathbf{A}\mathbf{x}$.

The output space is always mD , but the column space won’t always “fill” the whole of it. If there are only r independent columns of \mathbf{A} , then the column space is an rD subspace. For example, if the matrix has 3 rows, but the three column vectors are co-planar, then the column space is a plane in a 3D space. r is called the *rank* of the matrix.

Note that r cannot be larger than n (i.e. if all the columns are independent), so if n is smaller than m , then the column space will definitely not fill the entire output space. In the parlance of simultaneous equations, this is because you have more equations than variables.

If the column space doesn’t fill the whole output space, what happens in the rest of it? We can split our output point up into a component in the column space and a perpendicular component that isn’t,

$$\mathbf{y} = \mathbf{y}_C + \mathbf{y}_{LN} \quad (5)$$

The perpendicular component lies in the orthogonal subspace to the column space. This is called the “left-nullspace”. (Its got to be orthogonal or we could write it (or part of it) in terms of column space vectors, which would mean that it (or part of it) was actually in the column space.) If we’re in an mD output space, and the column space is rD , then there are $m - r$ direction perpendicular to the column space, so the left-nullspace is $(m - r)D$.

So, for our equation $\mathbf{y} = \mathbf{A}\mathbf{x}$ to have a solution, we must have $\mathbf{y}_{LN} = 0$. What does this mean? Well, for our structures example, a component in the left-nullspace represents a violation of the geometry constraints, i.e. one of the bars must have changed length. In the electric example, it means that Kirchoff’s laws have been broken, and in the mechanics example it means Newton’s laws have been broken. Left-nullspace outputs are impossible.

2.2 Input Side

For the input space, we’re going to expand \mathbf{A} in a different way, in terms of its rows.

$$\mathbf{A} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \vdots \\ \mathbf{r}_m^T \end{bmatrix} \quad (6)$$

Now we can write the multiplication like this,

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{r}_1 \cdot \mathbf{x} \\ \mathbf{r}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{r}_m \cdot \mathbf{x} \end{bmatrix}. \quad (7)$$

Taking the dot product of the input point with each of the row vectors gives us the “contribution” of that point in each of the row directions. This defines another interesting subspace, the “row space”, which contains all the useful

inputs which have an effect on our system. As before, we can write each input point in terms of a component in the row space and a perpendicular component,

$$\mathbf{x} = \mathbf{x}_R + \mathbf{x}_N \quad (8)$$

Now, there's nothing wrong with having an \mathbf{x}_N component. We're talking about the input to the system here, and we can choose that to be whatever we damn-well please. However, if \mathbf{x}_N is perpendicular to the row space component, then it must also be perpendicular to all the individual rows of \mathbf{A} . Perpendicular means that the dot product is zero, which leads us to the fact that,

$$\mathbf{A}\mathbf{x}_N = \mathbf{0}. \quad (9)$$

We call this orthogonal subspace the “nullspace”. As we can see, an input (or a part of it) in the nullspace has no effect on the output.

The input space is always nD , but the row space won't always fill the whole of it. The number of row space dimensions is given by the number of independent row vectors. Curiously, this is always r (i.e. the rank, the same as the column space). We'll see why when we do LU decompositions. The nullspace then has $(n - r)$ dimensions.

2.3 Transposing

If we transpose the matrix, then everything flips around. The row space of \mathbf{A} is the column space of \mathbf{A}^T , and vice-versa. The nullspace of \mathbf{A} is the left-nullspace of \mathbf{A}^T and vice-versa. This should be pretty obvious, because the transpose operation turns the rows into columns and the columns into rows. It also gives us a nice way to define the left-nullspace, as the nullspace of \mathbf{A}^T . This means that $\mathbf{A}^T \mathbf{y}_{LN} = \mathbf{0}$.

2.4 Summary

$$\mathbf{y} = \mathbf{A}\mathbf{x} \quad (10)$$

- \mathbf{x} is in the input space, which is divided into the *row space* and the *nullspace*. The nullspace maps onto $\mathbf{0}$ in the output space. The row space maps onto the column space.
- \mathbf{y} is in the output space, which is divided into the *column space* and the *left-nullspace*. If \mathbf{y} is in the column space then there is a solution for \mathbf{x} (in the row space). If \mathbf{y} is in the left-nullspace, there is no solution for \mathbf{x} .

3 Solving Matrix Equations

We're now going to consider solving equations of the form,

$$\mathbf{y} = \mathbf{A}\mathbf{x} \quad (11)$$

where \mathbf{A} and \mathbf{y} are known and fixed and we want to find \mathbf{x} . Back in the simple land of scalars, there was always 1 solution to this equation (unless \mathbf{A} was 0). Here in multi-dimensional space there could be one, zero, or an infinite

number of solution. We've already considered when no solution exists — its when $\mathbf{y}_{LN} \neq 0$.

It turns out that there will be one solution if $r = n$, and many (infinite) if $r < n$. To see why, let's think about the nullspace. If $r < n$, then the matrix has a nullspace (because it has $n - r > 0$ dimensions), and it's defined by $\mathbf{A}\mathbf{x}_N = \mathbf{0}$. If \mathbf{x}_0 is a solution to the equation, i.e. $\mathbf{y} = \mathbf{A}\mathbf{x}_0$, then we could just add some of \mathbf{x}_N onto the original solution and we'd still have a valid solution.

$$\mathbf{A}(\mathbf{x}_0 + \lambda\mathbf{x}_N) = \mathbf{A}\mathbf{x}_0 + \lambda\mathbf{A}\mathbf{x}_N \quad (12)$$

$$= \mathbf{y} + \mathbf{0} \quad (13)$$

So there will be an infinite number of solutions when the nullspace exists. Otherwise, there's just one.

We'll get onto how we actually find \mathbf{x}_0 and \mathbf{x}_N when we look at decompositions.

3.1 Least Squares Solutions

We said that we can't solve a matrix equation when $\mathbf{y}_{LN} \neq 0$. However, all is not lost. We can still find a best-fit/minimum-error solution. Remember that the left-nullspace is characterised by $\mathbf{A}^T\mathbf{y}_{LN} = \mathbf{0}$. So if we pre-multiply the output (i.e. the constraints) of the equation by \mathbf{A}^T , then we can make that awkward left-nullspace bit disappear.

$$\mathbf{A}^T(\mathbf{y}_C + \mathbf{y}_{LN}) = \mathbf{A}^T\mathbf{y}_C + \mathbf{A}^T\mathbf{y}_{LN} \quad (14)$$

$$= \mathbf{A}^T\mathbf{y}_C + \mathbf{0} \quad (15)$$

So if we take the original solution-less equation and pre-multiply by \mathbf{A}^T , we get a new equation which does have a solution.

$$\mathbf{A}^T\mathbf{y} = \mathbf{A}^T\mathbf{A}\mathbf{x} \quad (16)$$

This solution results in an output of \mathbf{y}_C , which is as close as we can get to \mathbf{y} .

4 LU Decompositions

Structure. Geometric interpretation. Rank equality proof.

5 QR Decompositions

Structure. Geometric interpretation. Least squares problems.

6 Eigenvalues and Singular Values

Definitions. Row/column space interpretation.

7 Eigenvalue and Singular Value Decompositions

Structure. Derivations. Interpretations (rotation, scale, rotation).