

Visualizing The Four Fundamental Subspaces in a Linear Programming Problem

Brooks Mershon

September 1, 2015

Please refer to Figure 1.6 on page 25 of *BT* throughout this exercise. Pages 21-25 cover all of section 1.4, which provides some useful techniques for visualizing standard form problems. Consider reading section 1.4 before continuing. I found section 1.4 helpful, but wanted to further explore the connection between a particular example contained within to some of the linear algebra prerequisites assumed for the course. In particular, I hope this exercise will help tie together the concepts of solving a system of linear equations with the Four Fundamental Subspaces in the context of a linear programming problem.

1 Constraints in \mathbb{R}^3

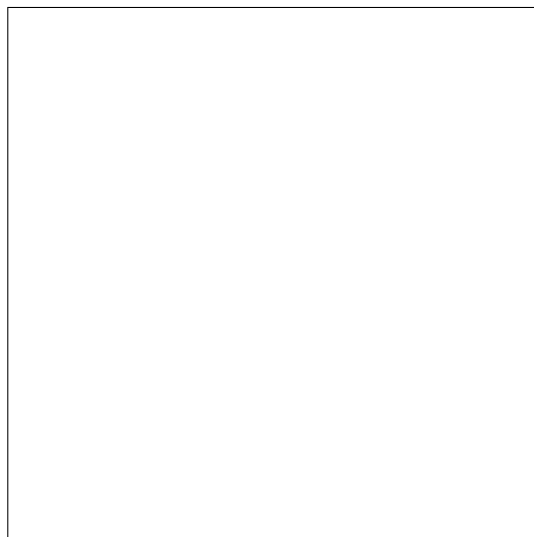
On page 25 of Bertsimas and Tsitsiklis (BT), we are instructed to visualize the feasible set in \mathbb{R}^3 defined by the constraints:

$$x_1 + x_2 + x_3 = 1$$

and

$$x_1, x_2, x_3 \geq 0$$

Try graphing the feasible set defined by this constraint in the empty box provided. You should get a view similar to Figure 1.6a from the textbook. The equation $x_1 + x_2 + x_3 = 1$ defines a hyperplane, which is in this case a plane in \mathbb{R}^3 .



Ignoring for a moment the constraint that x_1 , x_2 , and x_3 are all positive, we can represent the information contained in the hyperplane we visualized with the 3×1 matrix \mathbf{A} as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} 1 \end{bmatrix}$$

It is common for us to specify constraints in linear programming problems with the expression $\mathbf{A}\vec{x} = \vec{b}$. Here, \vec{x} is a decision variable, usually chosen such that $\mathbf{A}\vec{x} = \vec{b}$ holds while some objective function of \vec{x} is minimized. The graph you have just drawn shows a set in 3-dimensional space where the information contained in \mathbf{A} and \vec{b} has defined a hyperplane. Tasked with finding an assignment of x_1 , x_2 , and x_3 for which $\mathbf{A}\vec{x} = \vec{b}$ holds, you only need to produce a vector \vec{x} whose tip lies somewhere on the hyperplane (the feasible set). This feasible set is a triangle in \mathbb{R}^3 if $x_1, x_2, x_3 \geq 0$.

Go ahead and add a vector \vec{x} which stretches from the origin to a point somewhere in the feasible region of your drawing (the triangle in \mathbb{R}^3).

2 Resource allocation

It's often easier to think about abstract ideas by playing with concrete examples. The *hyperplane in three dimensions* example we just encountered can be made even more concrete by relating it to a “real-world” problem. Let us imagine a linear programming problem (LPP) that would be served well by the $\mathbf{A}\vec{x} = \vec{b}$ model of constraints that we just examined. This example will let us connect several linear algebra concepts to our new desire to solve all things that look like $\mathbf{A}\vec{x} = \vec{b}$.

Suppose you run a Bed & Breakfast in Vermont that specializes in local food. You keep chickens around so you always have lots of fresh eggs on hand, and you use these eggs to prepare, among other things, three fantastic breakfast dishes, each of which uses just one egg. You never know how many of these dishes you will serve on a particular day. Some days one of the three dishes (call them dish A, B, and C) is very popular while the others are rarely ordered. On other days guests have you cooking up all three dishes in similar quantities. You do, however, only have a set number of eggs on hand; perhaps 40 eggs on a good day and only 20 or so on a bad day. On a bad day in which you only have 24 fresh eggs, you will be able to serve 9 guests who order dish A, 4 guests order dish B, and 11 guests who order dish C, but you will not be able to serve 30 total orders for an egg dish. On that day, you would have to recommend other choices to those 6 who were late to come to breakfast!

If we wanted to visualize a vector space that told us something about the combinations of orders we could satisfy with b eggs on hand, we could imagine an 3×1 matrix \mathbf{A} that looked like this:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

And a vector \vec{x} specifying how many of each dish we serve:

$$\vec{x} = \begin{bmatrix} \# \text{ of dish A} \\ \# \text{ of dish B} \\ \# \text{ of dish C} \end{bmatrix}$$

$\mathbf{A}\vec{x}$ gives us the total number of eggs used in satisfying the orders vector \vec{x} . This is a scalar for all practical purposes, but we should still consider \vec{b} as a column vector of length 1:

$$\vec{b} = \begin{bmatrix} \# \text{ of eggs on hand} \end{bmatrix}$$

Go back to your graph and annotate the three axes, where x_1 is now dish A, x_2 is now dish B, and x_3 is now dish C. You have just created *dish space*. Any vector \vec{x} in *dish space*, where $x_1, x_2, x_3 \geq 0$, corresponds to some particular day's distribution of orders. For example, 10 of dish A, 3 of dish B, and no orders for dish C. The graph you have now annotated shows the combinations of orders you could satisfy if you had ONLY ONE EGG TO USE. If you relax the requirement that you cook

whole dishes, rather than fractions of a dish, then you can pick a vector \vec{x} anywhere in the feasible set you have drawn.

3 Column space in \mathbb{R}

The column space of a matrix \mathbf{A} is the span of the column vectors of \mathbf{A} . In this example, the three column vectors are not linearly independent. Intuitively, this is because the three dishes we make all use eggs. Viewed as a resource allocation problem, we have 3 products and one material. Each product (dish) uses one material (eggs). Our specific problem indicates that each product uses the same amount of material, but this could easily be changed so that dish A took 3 eggs, dish B took 2 eggs, and dish C required only 1 egg. So the column space in this case is really a one-dimensional “number line.” The vector \vec{b} lies in *egg space*, or *material space*. Given a vector \vec{x} in \mathbb{R}^3 , $\mathbf{A}\vec{x}$ gives us the number of eggs used, which is actually a vector in *egg space*. The *egg space* we have now visualized is in fact the *column space* of the matrix \mathbf{A} . And this column space is one-dimensional.

Go ahead and draw a number line in the box below. This is *egg space*. This is the *column space* of matrix \mathbf{A} . The matrix \mathbf{A} maps vectors from \mathbb{R}^3 to \mathbb{R} , or from *dish space* to *egg space*.



4 Null space in \mathbb{R}^3

The null space of a matrix \mathbf{A} is the set of vectors \vec{x} that satisfy $\mathbf{A}\vec{x} = \vec{0}$. If we restrict ourselves to positive numbers of dishes, then the only way to make dishes while using zero eggs is to make none of each dish. This corresponds to a plane through the origin in *dish space*. This plane can be shifted away from the origin in the direction of a normal to our plane by increasing the amount of eggs we have to use. This corresponds to picking a vector \vec{b} on our number line in *egg space* further to the right. The plane can be tilted if we change the entries of \mathbf{A} . Draw a “tilted” version of the feasible region by specifying a different matrix \mathbf{A} and corresponding graph in the box below.



Notice that a normal to our feasible set is orthogonal to our null space...

5 Row space in \mathbb{R}^3

The row space of a matrix \mathbf{A} is the span of the row vectors. In this case, we see that this is a line specified by scalar multiples of the normal vector to our hyperplane. The row space is the orthogonal complement to the null space, and it is spanned by a normal vector to our feasible set!. Any vector with positive components in \mathbb{R}^3 (*dish space*) can be thought of as some vector in the null space plus some vector in the row space of \mathbf{A} . Remember, the column space lives elsewhere, on the number line you have drawn, but the row space lives in the same space where you decide how many of each dish to make (the vector \vec{x}).

*So here's the really neat part: you can think of the row space as providing a direction to move if you want to require more or less eggs *in total*. If you want to reallocate your numbers of each type of dish so as to use the same number of eggs, then you should move around in the feasible set. How do you move around in the feasible set? Moving strictly along the row space keeps the number of each dish in proportion; you simply end up requiring more or less total eggs. If you take a vector \vec{x} that already satisfies $\mathbf{A}\vec{x} = \vec{b}$, and add to that vector some vector in the null space, you get another feasible combination of dishes. You are moving from one feasible set of orders in *dish space*, given a set number of eggs on hand, to another feasible set of orders in *dish space*.

6 Left null space in \mathbb{R}

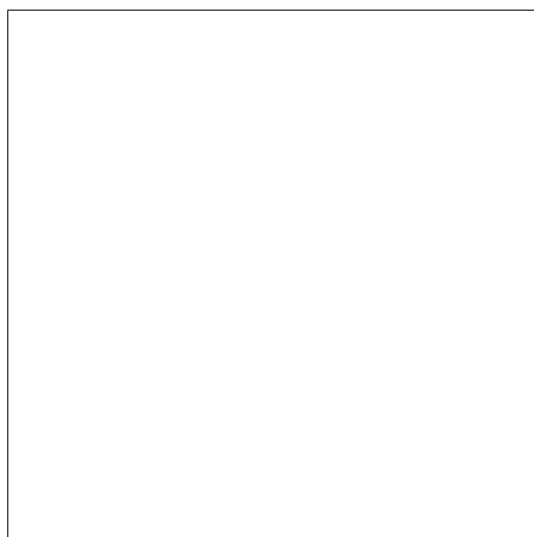
The left null space is the orthogonal complement to the column space. Since this subspace lives in \mathbb{R}^m , and $m = 1$, the dimension of this subspace is zero.

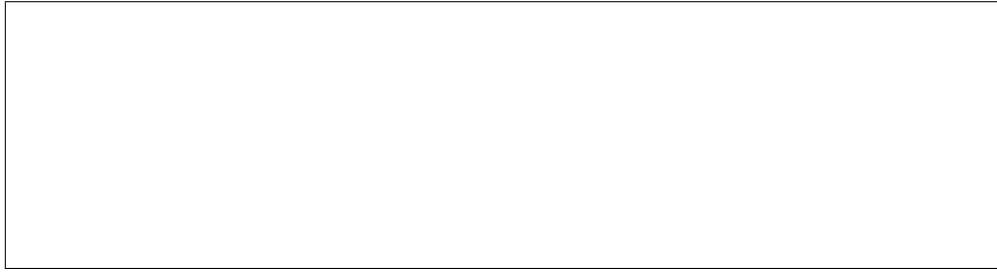
7 Where's the optimization?

Lastly, the linear programming formulation of the bed & breakfast problem requires some objective function to minimize (maximize). With just the feasible set that you have drawn, you might just stop at imagining the combinations of orders you will be able to satisfy as the guests show up and request various dishes, given the number of fresh eggs b that you have for the day. However, let us say each dish has a different cost, and the line chef gets bored if he doesn't get to experience some variety in the orders for the three dishes. With a price vector \vec{p} and an interest vector \vec{q} , we can aim to maximize the restaurant's success counterbalanced against the chef's fulfillment. Let p_i represent the price of dish i , and let q_i represent the chef's interest, normalized in a unit comparable to dollars, in cooking dish i . We wish to maximize:

$$\sum_{i=1}^n (p_i + q_i)x_i$$

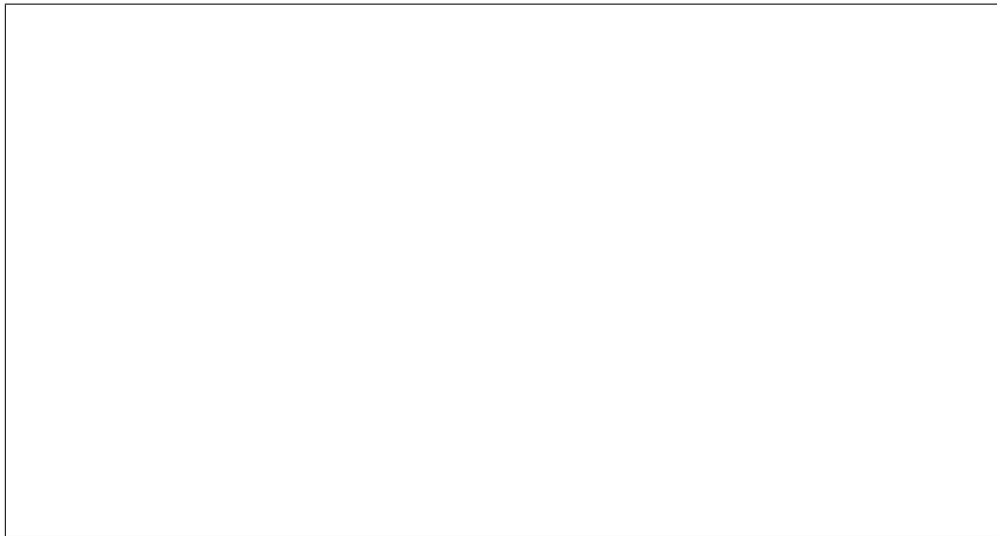
Look back at the graphs you have drawn and annotated. If things got messy, draw \mathbb{R}^3 and \mathbb{R} again in the boxes below. Try drawing and labeling the basis vectors corresponding to the column space (egg space) and row space.





I hope this helps. Writing this sure helped me better understand the connection between subspaces and solving constraints of the form $\mathbf{A}\vec{x} = \vec{b}$. The big box below is a good space to write down a diagram representing the Four Fundamental Subspaces. Gilbert Strang's (MIT) essay on the Four Fundamental Subspaces is a useful resource.

<http://web.mit.edu/18.06/www/Essays/newpaper_ver3.pdf>



For a refresher on the interaction between row space and null space, consider watching the following series of videos from Khan Academy:

<https://www.khanacademy.org/math/linear-algebra/alternate_bases>

Areas for further exploration:

- The left null space and quotient spaces and their relation to LPP's.
- Other “real-world” examples of LPP's visualized as linear transformations.
- Find interactive resources to put textbook theory into practice.