

LINEAR ALGEBRA: LECTURES 35–36

LEO GOLDBAKHER

1. FROM ABSTRACT TO CONCRETE

We've been thinking about abstract vector spaces the past few weeks. Last time we discussed how to represent an arbitrary linear map $T : V \rightarrow W$ as a matrix. One annoying aspect of this process is that we have to explicitly choose bases of V and W ; without doing this, it's impossible to deduce the transformation from the matrix (although even without knowing the bases, the matrix does tell you something – see below). The next result shows that any abstract vectors space can be viewed as a concrete one:

Theorem 1. *If $\dim V = n$, then $V \simeq \mathbb{R}^n$.*

Colloquial version. Every n -dimensional vector space V is \mathbb{R}^n , just using a different naming scheme for the elements.

Proof. We've already prove this: it is a special case of Proposition 2 from Lecture 33. □

Thus whenever we are working with an n -dimensional abstract vector space V , we can safely assume it's the concrete space \mathbb{R}^n . Among other advantages, \mathbb{R}^n comes with a standard basis: $\{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_n\}$ where

$$\vec{e}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \vec{e}_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \text{ etc.}$$

How do we translate between the languages of V and \mathbb{R}^n ? Well, by the above theorem, there exists an isomorphism $P : V \rightarrow \mathbb{R}^n$; we can use this as a dictionary to look up the new name for any particular element. We can also translate matrices from one language into the other. Given a linear map $T : V \rightarrow V$, let \tilde{T} denote the same map phrased in the language of \mathbb{R}^n , and consider the following diagram:

$$\begin{array}{ccc} V & \xrightarrow{T} & V \\ P \downarrow & & \downarrow P \\ \mathbb{R}^n & \xrightarrow{\tilde{T}} & \mathbb{R}^n \end{array}$$

This diagram shows that $\tilde{T} = PTP^{-1}$. This explains why we called it a change-of-basis earlier: we are literally changing the basis! More generally, one can show that given any linear transformation $V \rightarrow V$, any two matrices M and N which represent this linear transformation are similar (i.e. there exists P such that $M = P^{-1}NP$).

The conclusion from all the above is that we can restrict our attention to linear maps from \mathbb{R}^n to \mathbb{R}^m . Hence, from now on when we write a matrix we will not specify which bases we are taking, and assume that we are using the standard basis for both the source and the target space.

2. SOME TECHNICAL DETAILS

A bit of thought shows that an $m \times n$ matrix represents a linear map $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$; there are n columns (hence n basis elements we're plugging into f), and m rows (hence the outputs of f have m coordinates). This is a bit confusing, and is primarily due to the fact that we're used to writing $f(x)$ rather than $(x)f$. But there's no need to memorize this – it suffices to think through what it tells you about the matrix to have a certain number of rows or columns.

Next, we discussed composition of linear functions. For example, suppose $f : \mathbb{R}^5 \rightarrow \mathbb{R}^2$ and $g : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ are linear functions. Something happens which we didn't see in the two-dimensional case: the composition $f \circ g$ doesn't exist, since the output of g cannot be meaningfully plugged into f . By contrast, the composition $g \circ f$ does exist, and is a linear map from $\mathbb{R}^5 \rightarrow \mathbb{R}^3$. So, when studying compositions of linear functions, we must be careful to check that the dimensions line up.

Once we make sure that the composition is legitimate, how do we actually determine its matrix? Pretty much the same way as we're used to from earlier in the course: take the dot product of each row in the left matrix with a column in the right, and make the resulting number the appropriate entry in the matrix. For example,

$$\begin{pmatrix} 1 & -1 & 0 & 3 & 2 \\ * & * & * & * & * \end{pmatrix} \begin{pmatrix} * & * & 2 \\ * & * & 2 \\ * & * & -3 \\ * & * & 1 \\ * & * & 1 \end{pmatrix} = \begin{pmatrix} * & * & 5 \\ * & * & * \end{pmatrix}$$

where the asterisks represent some unknown entries. The i^{th} row multiplied by the j^{th} column produces the number in the i^{th} row and j^{th} column of the composition matrix (just as in the 2×2 case).

Note that a linear map from \mathbb{R}^m to \mathbb{R}^n is typically not invertible. Indeed, if $m > n$, then the target space is very small relative to the source (think of the plane versus 3-space), so there's not enough space to store all the information from the source space. If, on the other hand, $m < n$, then the target space is huge relative to the source space, so it cannot possibly be covered by the image of the source space. However, linear maps $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ can be invertible; the matrix of such an f is called a *square matrix*, since it's $n \times n$. Just as in the two-dimensional case, it can be shown that a square matrix is invertible if and only if its determinant is nonzero. But what is the determinant? And how do we find the inverse? The answers to these questions are more complicated than their two-dimensional cousins. We begin with the determinant.

3. DETERMINANTS OF LINEAR MAPS $\mathbb{R}^n \rightarrow \mathbb{R}^n$

Given a linear map $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, we can approach the determinant in much the same way as in the two-dimensional case: it captures what f does to a unit of volume. In particular, the determinant is the volume of the image of the unit cube under f , except that we allow the determinant to be negative if f flips the orientation of the cube. This time, there are more directions in which one can flip than in the two-dimensional case, and every time we flip space we multiply by another -1 . Actually, this happens in \mathbb{R}^2 as well: if a linear map flips the plane once, its determinant is negative; if it flips the plane twice, its determinant is positive, etc.

All of this is rather hard to picture, so, in analogy to the 2-d case, one might hope for an algebraic formula for $\det f$ in terms of the entries in the matrix of f . Somewhat amazingly, no such formula is known. Or rather, for any particular n one can determine a formula for the determinant of an $n \times n$ matrix, but these formulas quickly become overwhelmingly cumbersome, and no one's found a truly simple and efficient method for calculating the determinant.

Here we shall describe three methods for calculating the determinant. The first, called the *Laplace expansion* in honor of not its discoverer, is a terrible method for a typical matrix, but is useful for matrices with lots of zeros in them. The second approach, called *Dodgson condensation*, was devised by Lewis Carroll and is a relatively elegant method; the only downside is that this algorithm occasionally breaks. The final method we discuss, that of *elementary row operations*, is pretty straightforward to carry out and also gives a nice approach to calculating inverses. We consider these three methods in turn.

3.1. The Laplace Expansion (a.k.a. cofactor expansion). To make notation easier, we shall use vertical lines to denote the determinant of a matrix, e.g.

$$\begin{vmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{vmatrix} := \det \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}$$

The idea of the Laplace expansion is to express an $n \times n$ determinant as a linear combination of $(n-1) \times (n-1)$ determinants. Here's how it would work for 4×4 determinants:

$$\begin{aligned} \begin{vmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{vmatrix} &= \begin{pmatrix} a & & & \\ & \cdot & & \\ & & \begin{vmatrix} f & g & h \\ j & k & l \\ n & o & p \end{vmatrix} & \\ & & & \end{pmatrix} - \begin{pmatrix} & b & & \\ & \cdot & & \\ & & \begin{vmatrix} e & & g & h \\ i & & k & l \\ m & & o & p \end{vmatrix} & \\ & & & \end{pmatrix} \\ &+ \begin{pmatrix} & & c & \\ & & \cdot & \\ & & & \begin{vmatrix} e & f & & h \\ i & j & & l \\ m & n & & p \end{vmatrix} & \\ & & & & \end{pmatrix} - \begin{pmatrix} & & & d \\ & & & \cdot \\ & & & & \begin{vmatrix} e & f & g \\ i & j & k \\ m & n & o \end{vmatrix} & \\ & & & & \end{pmatrix} \\ &= a \begin{vmatrix} f & g & h \\ j & k & l \\ n & o & p \end{vmatrix} - b \begin{vmatrix} e & g & h \\ i & k & l \\ m & o & p \end{vmatrix} + c \begin{vmatrix} e & f & h \\ i & j & l \\ m & n & p \end{vmatrix} - d \begin{vmatrix} e & f & g \\ i & j & k \\ m & n & o \end{vmatrix} \end{aligned}$$

Thus, we have reduced a 4×4 determinant to calculating four 3×3 determinants. How do we calculate a 3×3 determinant? Same idea:

$$\begin{aligned} \begin{vmatrix} q & r & s \\ t & u & v \\ w & x & y \end{vmatrix} &= \begin{pmatrix} q & & \\ & \cdot & \\ & & \begin{vmatrix} u & v \\ x & y \end{vmatrix} \end{pmatrix} - \begin{pmatrix} & r & \\ & \cdot & \\ & & \begin{vmatrix} t & v \\ w & y \end{vmatrix} \end{pmatrix} + \begin{pmatrix} & & s \\ & & \cdot \\ & & & \begin{vmatrix} t & u \\ w & x \end{vmatrix} \end{pmatrix} \\ &= q \begin{vmatrix} u & v \\ x & y \end{vmatrix} - r \begin{vmatrix} t & v \\ w & y \end{vmatrix} + s \begin{vmatrix} t & u \\ w & x \end{vmatrix} \end{aligned}$$

Now, we can evaluate each of these 2×2 determinants to obtain a formula for a 3×3 determinant. Having done so, we could evaluate each of the 3×3 determinants above to obtain a formula for the original 4×4 determinant.

Needless to say, this is usually a painful process to carry out. However, for *sparse* matrices – i.e. matrices in which most entries are zero – this algorithm simplifies a good deal. A trivial example of this is the n -dimensional identity matrix

$$I = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix},$$

all of whose entries are zero aside from 1's along the main diagonal. Of course, you don't need any fancy algorithm to calculate the determinant – the identity maps every point to itself, in particular sending the unit cube to the unit cube – but it is a good first example to convince yourself that the Laplace algorithm is occasionally reasonable to use. A less obvious case in which the Laplace expansion is helpful is any matrix of the form

$$L := \begin{pmatrix} a_1 & & & \\ * & a_2 & & \\ * & * & \ddots & \\ * & * & \cdots & a_n \end{pmatrix}$$

where the empty spaces are all filled with zeros, the asterisks are arbitrary real numbers, and the a_i 's (also arbitrary real numbers) form the main diagonal of the matrix. Applying the Laplace expansion repeatedly, we see that the determinant of any such matrix (such matrices are usually called *lower triangular*) is simply the product of the diagonal entries:

$$\det L = a_1 a_2 \cdots a_n.$$

Nonetheless, it is clearly undesirable to carry out Laplace expansion on non-sparse matrices. The next algorithm is significantly nicer.¹

3.2. Lewis Carroll's method, a.k.a. Dodgson Condensation. Laplace expansion reduced the problem of evaluating an $n \times n$ determinant to that of evaluating a bunch of $(n - 1) \times (n - 1)$ determinants. Dodgson condensation – discovered by Charles Dodgson, better known under his pen name Lewis Carroll – computes an $n \times n$ determinant in terms of a bunch of 2×2 determinants. Here's how the process goes.

Given an $n \times n$ matrix, we will construct an $(n - 1) \times (n - 1)$ matrix which we picture as floating above our matrix. For example, given a 5×5 matrix, we will create a 4×4 matrix floating above the 5×5 , whose entries are located above the blue dots in the picture below:

$$\begin{pmatrix} a & b & c & d & e \\ & \bullet & & \bullet & \\ f & g & h & i & j \\ & \bullet & & \bullet & \\ k & l & m & n & o \\ & \bullet & & \bullet & \\ p & q & r & s & t \\ & \bullet & & \bullet & \\ u & v & w & x & y \end{pmatrix}$$

What are the entries of this 4×4 matrix? Each blue dot is at the center of a 2×2 submatrix; simply put the determinant of the 2×2 where the blue dot is. Easy!

Now we iterate the procedure, each time creating a matrix one dimension smaller than the previous one and floating above it in the spaces between entries. At every stage, each entry of the new matrix is created by evaluating the 2×2 determinant it's the center of. Repeating this process a bunch of times, we will eventually arrive at a 1×1 determinant. Sadly, this is **not** the determinant of the original matrix. However, with one more twist added to the algorithm, it will be. Here's the missing ingredient: every time a number is floating directly above another number, we divide the floating number by the one below it. Let's see an example of this.

Consider the 5×5 matrix

$$\begin{pmatrix} 1 & 2 & -3 & -1 & 2 \\ 0 & 2 & -1 & -1 & 3 \\ 3 & 1 & 1 & 2 & 0 \\ 1 & 1 & -1 & 2 & 1 \\ 2 & 3 & -1 & 2 & 3 \end{pmatrix}$$

Into this matrix we insert the 2×2 determinants, colored blue (ignore the circled numbers for now – they will play a role later):

$$\begin{pmatrix} 1 & 2 & -3 & -1 & 2 \\ & 2 & 4 & 2 & -1 \\ 0 & \textcircled{2} & & \textcircled{-1} & \textcircled{-1} & 3 \\ & -6 & 3 & -1 & -6 \\ 3 & \textcircled{1} & \textcircled{1} & \textcircled{2} & 0 \\ & 2 & -2 & 4 & 2 \\ 1 & \textcircled{1} & \textcircled{-1} & \textcircled{2} & 1 \\ & 1 & 2 & 0 & 4 \\ 2 & 3 & -1 & 2 & 3 \end{pmatrix}$$

¹Note: we didn't cover this method in class, and it will not be tested. But it's cool, and not well-known, so I thought I'd include it for fun.

None of these blue numbers are floating above any other numbers, so we obtain the following 4×4 matrix (ignore the boxed numbers):

$$\begin{pmatrix} 2 & 4 & 2 & -1 \\ -6 & \boxed{3} & \boxed{-1} & -6 \\ 2 & \boxed{-2} & \boxed{4} & 2 \\ 1 & 2 & 0 & 4 \end{pmatrix}$$

Next, we repeat the process with this 4×4 , again inserting the 2×2 determinants:

$$\begin{pmatrix} 2 & 4 & 2 & -1 \\ -6 & 30 & -10 & -13 \\ 2 & 6 & 10 & 22 \\ 1 & 6 & -8 & 16 \end{pmatrix}$$

Observe that this blue matrix is floating directly on top of the circled entries in the original 5×5 ! So, we divide each blue number by the circled number above which it floats, obtaining

$$\begin{pmatrix} 15 & 10 & 13 \\ 6 & 10 & 11 \\ 6 & 8 & 8 \end{pmatrix}$$

Repeating the process once more, we first insert the values of the 2×2 determinants in blue

$$\begin{pmatrix} 15 & 10 & 13 \\ 6 & 90 & -20 \\ 6 & -12 & -8 \\ 6 & 8 & 8 \end{pmatrix}$$

Since the blue matrix is floating directly above the entries in boxes from above, we divide and obtain

$$\begin{pmatrix} 30 & 20 \\ 6 & -2 \end{pmatrix}$$

Final iteration! First, we insert the determinant in the middle:

$$\begin{pmatrix} 30 & 20 \\ 6 & -180 \\ 6 & -2 \end{pmatrix}$$

This is floating directly above the 10 in the 3×3 matrix above, so we divide and find the 1×1 matrix (-18) . Thus, we conclude that

$$\det \begin{pmatrix} 1 & 2 & -3 & -1 & 2 \\ 0 & 2 & -1 & -1 & 3 \\ 3 & 1 & 1 & 2 & 0 \\ 1 & 1 & -1 & 2 & 1 \\ 2 & 3 & -1 & 2 & 3 \end{pmatrix} = -18$$

A few comments are in order. First, note that in the final step we divided by the underlying entry in the third-to-last step, and not by anything from the fifth-to-last. This is a general rule: at each step (other than the first), we divide by entries from two steps previous.

Second, notice that this process could potentially go very wrong – we might have to divide by zero at some intermediate stage! (For example, try applying the method to the identity map.) There's a way around this issue, but it's a bit technical so we won't discuss it right now. Suffice to say that this happens very rarely – it's a big coincidence if there happens to be a zero appearing at any stage of the process.

Third, it's *not* true that the determinants of the intermediate matrices are all the same. In particular, one cannot stop the method in the middle – when using this algorithm, you must go all the way to the 1×1 matrix.

3.3. Elementary row operations. There is yet another way to calculate the determinant. Unlike the previous two approaches, here we will evaluate the determinant directly from the entries of the matrix, rather than in terms of other affiliated determinants. Moreover, this method will inspire an approach to inverting a matrix.

One way to think about this third approach is as a game: the start of the game is some given matrix, and your goal is to make a sequence of modifications to the matrix until it looks like a triangular matrix:

$$\begin{pmatrix} * & * & * & * \\ & * & * & * \\ & & \ddots & \vdots \\ & & & * \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} * & & & \\ * & * & & \\ * & * & \ddots & \\ * & * & \cdots & * \end{pmatrix}$$

(As before, the asterisks denote any real number, and empty spaces denote entries which are zero.) The rules of the game are that you are only allowed to make the three following types of moves:

- (1) You may switch any two rows of the matrix; we call this operation $S_{i,j}$ (if the i^{th} and j^{th} row are switched). For example, $S_{2,3}$ switches the 2nd and 3rd rows.
- (2) You may multiply any row by any real constant; we call this $M_i(c)$ if the i^{th} row is multiplied by c .
- (3) You may add a multiple of any row to any other row; we call this $A_{i,j}(c)$ if c copies of the j^{th} row are added to the i^{th} .

These three types of moves are called the *elementary row operations*. Here's an example of this game. (Recall that a blank space means there's a zero there.)

$$\begin{aligned} \begin{pmatrix} 6 & 15 & 15 & 6 \\ -2 & -4 & 2 & -1 \\ 2 & 5 & 4 & 1 \\ 6 & 13 & 0 & 0 \end{pmatrix} &\xrightarrow{M_1(1/6)} \begin{pmatrix} 1 & 2.5 & 2.5 & 1 \\ -2 & -4 & 2 & -1 \\ 2 & 5 & 4 & 1 \\ 6 & 13 & 0 & 0 \end{pmatrix} \xrightarrow{A_{2,1}(2)} \begin{pmatrix} 1 & 2.5 & 2.5 & 1 \\ & 1 & 7 & 1 \\ 2 & 5 & 4 & 1 \\ 6 & 13 & 0 & 0 \end{pmatrix} \\ &\xrightarrow{A_{3,1}(-2)} \begin{pmatrix} 1 & 2.5 & 2.5 & 1 \\ & 1 & 7 & 1 \\ & & -1 & -1 \\ 6 & 13 & 0 & 0 \end{pmatrix} \xrightarrow{A_{4,1}(-6)} \begin{pmatrix} 1 & 2.5 & 2.5 & 1 \\ & 1 & 7 & 1 \\ & & -1 & -1 \\ & -2 & -15 & -6 \end{pmatrix} \\ &\xrightarrow{A_{4,2}(2)} \begin{pmatrix} 1 & 2.5 & 2.5 & 1 \\ & 1 & 7 & 1 \\ & & -1 & -1 \\ & & -1 & -4 \end{pmatrix} \xrightarrow{M_3(-1)} \begin{pmatrix} 1 & 2.5 & 2.5 & 1 \\ & 1 & 7 & 1 \\ & & 1 & 1 \\ & & -1 & -4 \end{pmatrix} \\ &\xrightarrow{A_{4,3}(1)} \begin{pmatrix} 1 & 2.5 & 2.5 & 1 \\ & 1 & 7 & 1 \\ & & 1 & 1 \\ & & & -3 \end{pmatrix} \end{aligned}$$

Since we've arrived at a triangular matrix, we win this round! It's worth noticing that there are many ways to win the game, and many different triangular matrices one might arrive at.

What on earth does this have to do with finding determinants? This becomes clear once we make two observations. The first one is that it's easy to evaluate the determinant of a triangular matrix:

Lemma 2. *The determinant of a triangular matrix is the product of the entries along its main diagonal.*

For example,

$$\begin{vmatrix} 1 & 2.5 & 2.5 & 1 \\ & 1 & 7 & 1 \\ & & 1 & 1 \\ & & & -3 \end{vmatrix} = -3.$$

We already saw this above for lower triangular matrices (i.e. those with nothing in the upper right corner). It turns out that a matrix and its transpose always have the same determinant, which guarantees that the lemma holds for upper triangular matrices as well.

Thus, our game above relates the original matrix to one whose determinant we can figure out easily. But what does this tell us about the determinant of the original matrix? This is where our second observation comes in.

Lemma 3. *There exist $n \times n$ matrices $S_{i,j}$, $M_i(c)$, and $A_{i,j}(c)$ such that for any $n \times n$ matrix f :*

- (a) $S_{i,j} \circ f$ exchanges the i^{th} and j^{th} rows of f ;
- (b) $M_i(c) \circ f$ multiplies the i^{th} row of f by c ; and
- (c) $A_{i,j}(c) \circ f$ adds c copies of the j^{th} row of f to the i^{th} row of f .

These three types of matrices ($S_{i,j}$, $M_i(c)$, and $A_{i,j}(c)$) are called the *elementary matrices*. Although the lemma only asserts their existence, it secretly tells us how to write down these matrices. For example, what is the 4×4 matrix $S_{2,4}$? By the lemma, no matter which matrix we apply it to, $S_{2,4}$ is supposed to switch the 2nd and 4th rows. In particular,

$$S_{2,4} = S_{2,4} \circ I = S_{2,4} \circ \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & & & 1 \\ & & 1 & \\ & 1 & & \end{pmatrix}$$

We can play the same game with all the elementary matrices. In other words, the elementary matrices are simply the result of applying the appropriate elementary row operation to the identity matrix I . As a consequence, all the elementary row matrices look pretty similar to I , and (in particular) are all sparse! This means we can easily calculate their determinants using the Laplace expansion; we obtain:

$$\det S_{i,j} = -1 \quad \det M_i(c) = c \quad \det A_{i,j}(c) = 1.$$

Putting this all together, we can now calculate the determinant of any matrix, so long as we can win the game of transforming it into a triangular using elementary row operations. Indeed, given some linear function $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, suppose we are able to find some sequence of elementary row operations which turns f into a triangular matrix T . Then there exists a sequence of elementary $n \times n$ matrices $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_k$ such that

$$\mathcal{E}_1 \mathcal{E}_2 \cdots \mathcal{E}_k f = T.$$

Taking determinants of both sides and using the fact that the determinant is multiplicative gives

$$(\det \mathcal{E}_1)(\det \mathcal{E}_2) \cdots (\det \mathcal{E}_k)(\det f) = \det T,$$

whence we deduce

$$\det f = \frac{\det T}{(\det \mathcal{E}_1)(\det \mathcal{E}_2) \cdots (\det \mathcal{E}_k)}$$

This is good news, because all the determinants on the right hand side are easy to calculate.

To see this in action, consider our example from above. We can rewrite our work in terms of elementary matrix actions:

$$A_{4,3}(1) \circ M_3(-1) \circ A_{4,2}(2) \circ A_{4,1}(-6) \circ A_{3,1}(-2) \circ A_{2,1}(2) \circ M_1(1/6) \circ \begin{pmatrix} 6 & 15 & 15 & 6 \\ -2 & -4 & 2 & -1 \\ 2 & 5 & 4 & 1 \\ 6 & 13 & 0 & 0 \end{pmatrix} \\ = \begin{pmatrix} 1 & 2.5 & 2.5 & 1 \\ & 1 & 7 & 1 \\ & & 1 & 1 \\ & & & -3 \end{pmatrix}$$

The determinant of the left side is

$$-\frac{1}{6} \begin{vmatrix} 6 & 15 & 15 & 6 \\ -2 & -4 & 2 & -1 \\ 2 & 5 & 4 & 1 \\ 6 & 13 & 0 & 0 \end{vmatrix},$$

while the determinant of the right side is -3 . This immediately tells us that

$$\begin{vmatrix} 6 & 15 & 15 & 6 \\ -2 & -4 & 2 & -1 \\ 2 & 5 & 4 & 1 \\ 6 & 13 & 0 & 0 \end{vmatrix} = 18.$$

4. INVERSES

In the third method described above, we stopped applying elementary matrices once we reached a triangular matrix, since it's easy to calculate the determinant at that point. If we want to, though, we can keep finding elementary row operations until we get down to the identity matrix I . Say we do this, ending up with

$$\mathcal{E}_1 \mathcal{E}_2 \cdots \mathcal{E}_k f = I$$

Right away, this tells you that you can solve for f in terms of the inverses of the \mathcal{E}_i . But we can do better. Take the inverse of both sides of the equation:

$$(\mathcal{E}_1 \mathcal{E}_2 \cdots \mathcal{E}_k f)^{-1} = I^{-1}$$

Recalling that $(AB)^{-1} = B^{-1}A^{-1}$ and simplifying, we see that

$$f^{-1}(\mathcal{E}_1 \mathcal{E}_2 \cdots \mathcal{E}_k)^{-1} = I$$

from which we deduce that

$$f^{-1} = \mathcal{E}_1 \mathcal{E}_2 \cdots \mathcal{E}_k.$$

This is already cool, but one more observation makes this really easy to evaluate:

$$\mathcal{E}_1 \mathcal{E}_2 \cdots \mathcal{E}_k = \mathcal{E}_1 \mathcal{E}_2 \cdots \mathcal{E}_k I.$$

The right hand side has a nice interpretation, as it's just applying a sequence of elementary row operations to I . This proves the following.

Proposition 4. *Suppose a sequence of elementary row operations transforms a matrix f into the identity matrix. Then the same sequence of elementary row operations applied to I produces f^{-1} .*

Before leaving the subject of elementary matrices, let me mention one final point. Through this discussion, we've been talking about elementary row operations. What about elementary column operations? It turns out that the *same* set of elementary matrices can be used, but rather than acting on the left (e.g. $\mathcal{E} \circ f$), we act on the right (e.g. $f \circ \mathcal{E}$). When acting on the right, an elementary matrix does exactly the same thing it did to f from the left, except that it does it to the columns of f rather than to the rows. For example, $f \circ S_{2,4}$ would swap the 2nd and 4th columns of f .

5. DECOMPOSITIONS

We've discussed inverses and determinants of linear maps $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$. What about the SVD and spectral decompositions? In two dimensions, these looked somewhat similar (both decompose T into a composition of three linear maps, the outer two of which have features in common), but there was one big difference – every nonsingular matrix has an SVD, but *not* every nonsingular matrix has a spectral decomposition. This distinction continues to hold in higher dimensions, but another important difference emerges: singular value decompositions can be made for arbitrary nonsingular matrices, while spectral decompositions can only be made for square matrices (and even then, not always). For this reason, the SVD is a significantly more potent tool in general than the spectral decomposition. However, both come up often in applications, so you should know what they are. Fortunately, both decompositions are fairly straightforward generalizations of the two-dimensional version.

5.1. Singular Value Decomposition. Given any nonsingular linear map $T : \mathbb{R}^m \rightarrow \mathbb{R}^n$. It turns out we can always find a decomposition

$$T = \mathcal{O}_n \circ D \circ \mathcal{O}_m$$

where

- \mathcal{O}_k denotes an *orthogonal* linear map $\mathcal{O}_k : \mathbb{R}^k \rightarrow \mathbb{R}^k$ (see below for details), and
- D denotes a diagonal linear map $D : \mathbb{R}^m \rightarrow \mathbb{R}^n$ (again, defined below).

Here an *orthogonal* map is any invertible linear map \mathcal{O} satisfying $\mathcal{O}^{-1} = \mathcal{O}^t$ (i.e. its inverse equals its transpose); note that \mathcal{O} must be a square matrix. Such maps have a number of nice properties:

Proposition 5. *If $\mathcal{O} : \mathbb{R}^k \rightarrow \mathbb{R}^k$ is orthogonal, then*

- (1) $\det \mathcal{O} = \pm 1$,
- (2) \mathcal{O} is distance-preserving, and
- (3) \mathcal{O} is a composition of rotations and reflections.

Next we turn to the notion of a *diagonal* map. We say D is diagonal iff for all pairs of distinct integers i, j , the entry in the i^{th} row and j^{th} column of D is 0.

Thus, this is very similar to the 2-dimensional case, the only real distinctions being (a) that the orthogonal transformations might be different dimensions from one another, (b) the diagonal matrix might not be a square, and (c) the orthogonal matrices might not be pure rotations, even if it has determinant 1.

What is the SVD useful for? I won't get into this too much, but here's the idea. Imagine you take a digital photo. Uncompressed, this is a lot of data: one might have an array of 768×1024 pixels, and each pixel is a combination of some amount of red, some amount of green, and some amount of blue. Let's separate these three colors out, so that we have one 768×1024 array of numbers keeping track of how much red goes into each pixel, a similar array keeping track of green, and another array keep track of blue. Decompose the red array into its SVD; it turns out this can be done relatively efficiently. The diagonal matrix D in this decomposition stores 768 singular values; we may as well line them up in decreasing order along the diagonal. We now create a new (and much smaller!) diagonal matrix: D' , whose diagonal consists of the ten largest singular values from the original D . Trimming rows of columns of the two orthogonal matrices accordingly, we have found a matrix which captures a lot of the behavior of the original, but uses **much** less space. Doing this for all three of the color arrays and putting together the result gives a compressed version of the image – slightly grainier, but still recognizable.

5.2. Spectral Decomposition. The idea of a spectral decomposition is exactly the same as in the 2-dimensional case: given a linear map $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$, the goal is to write it in the form

$$T = P \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix} P^{-1}$$

where $P : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is invertible. If one can accomplish this, the above expression is called the *spectral decomposition* of T , and gives a way to change basis into a nice one (consisting of *eigenvectors* of T) with respect to which T is simply a rescaling of the basis (each eigenvector being rescaled by a corresponding *eigenvalue* λ_i). More formally, an eigenvalue of T is any real number satisfying

$$T\vec{v} = \lambda\vec{v}$$

for some nonzero vector \vec{v} ; the \vec{v} in question is called the eigenvector of T corresponding to λ . It can be shown (similarly to what you did on a problem set) that if a spectral decomposition exists, then the set of numbers appearing along the diagonal in the diagonal matrix of the decomposition is a complete set of eigenvalues, and that P is the matrix mapping the standard basis \vec{e}_i to the eigenbasis elements \vec{v}_i . Put more simply, the columns of P are the eigenvectors.

Before reviewing the algorithm for finding the spectral decomposition, let me repeat two caveats from above:

- (1) if a matrix isn't square, it does *not* admit a spectral decomposition, and
- (2) *not* every square matrix admits a spectral decomposition.

The question of which square matrices do and don't admit such a decomposition is an interesting one, and you might like to look up the *Spectral Theorem* for a partial answer.

To (attempt to) determine the spectral decomposition of a linear map $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$, we follow the same three steps from before:

- (1) Find all solutions λ to the equation $\det(f - \lambda I) = 0$. These are the eigenvalues. (The left side of this equation is a polynomial in λ ; it is called the *characteristic polynomial*.)
- (2) For each eigenvalue, find all the corresponding eigenvectors. One can use a similar trick to the one we used before... rescale the mysterious eigenvector until the bottom coordinate is a 1, then solve for the unknowns.
- (3) Construct the change of basis matrix P by making the eigenvectors be the columns of P .

Of all these steps, the only one which is significantly more difficult than in the 2-dimensional case is the first step: finding the roots of the characteristic polynomial. For 2×2 matrices this was easy, since we have a machine (the quadratic formula) which solves any quadratic equation. But as the size of the matrix increases, finding the roots of the characteristic polynomial becomes harder and harder. Indeed, a famous result (discovered independently by Abel and Galois) asserts that there does not exist *any* formula in terms of standard mathematical operations which finds the roots of arbitrary quintic polynomials (same for degree larger than 5, too). Although people have devised very clever alternative methods of finding eigenvalues, dealing with large matrices remains a major headache, and an important problem in applications.