Determinants

September 7, 2017

1 Determinants

One of the first things that most students learn about in linear algebra is the determinant of a matrix. Lots of useful formulas for 2×2 and 3×3 matrices can be expressed in terms of determinants, and determinants played a central role in linear algebra 100 years ago when most matrices were tiny.

Nowadays, determinants are much less useful as a practical tool, although they still occasionally show up. Determinant-related formulas are also useful in proving theorems in linear algebra. The basic computational problem, however, is that the determinant formulas don't scale — for a big matrix, there is almost always a better way of computing something than using explicit determinants, cofactors, Cramer's rule, and other tricks useful for small matrices.

Still, it is important to know what determinants are, and their basic properties. In 18.06, we mainly use determinants as a *conceptual* tool to help us understand eigenvalues via the characteristic polynomial — although, again, this is not a practical *computational* tool for eigenvalues, which are nowadays computed by very different methods.

2 Expectation: Singular = Zero determinant

The property that most students learn about determinants of 2×2 and 3×3 is this: **given a square matrix A, the determinant det(A) is some number that is zero if and only if the matrix is singular**.

For example, the following matrix is not singular, and its determinant (`det(A)` in Julia) is nonzero:

```
In [1]: A = [1 3
             2 4]
        det(A)

Out[1]: -2.0
```

(You may even remember the formula for the 2×2 determinant: $1 \times 4 - 3 \times 2 = -2$.

But this matrix is singular (the second column is twice the first), and so its determinant is zero:

```
In [2]: A = [1 2
             2 4]
        det(A)

Out[2]: 0.0
```

- By the way, many authors, including Strang's book, use the abbreviated notation $|A| = \det A$. I won't use this notation here, mainly because I don't think the determinant is important enough anymore to deserve its own punctuation. Anyway, $|A|$ looks too much like an absolute value, even though the determinant can have any sign.

2.1 A lucky guess for the determinant

In 18.06, we know have another way to check whether a matrix is zero: perform Gaussian elimination, and then **check whether any pivots (diagonal entries of U) are zero**.

But this gives us an obvious way to construct a single determinant-like number: **just multiply the pivots together**, and the result will be zero if and only if the matrix is singular.

In fact, this intuition turns out to be *almost* exactly the right guess:

- The **determinant is ± the product of the pivots**, with a minus sign if elimination involved an *odd* number of row swaps and a plus sign if there were an *even* number of swaps (including zero swaps).

We can check this for a random matrix:

```
In [3]: A = randn(5,5)
        det(A)
```

```
Out[3]: 1.4856724724450872
```

```
In [4]: L,U = lu(A, Val{false}) # LU without row swaps
        U
```

```
Out[4]: 5×5 Array{Float64,2}:
         -1.28889  0.853533  -0.590642  -0.844207   0.858544
          0.0      0.469495  -0.11632   -0.269368   0.549657
          0.0      0.0        0.962075   0.240636  -0.94799
          0.0      0.0        0.0        1.70485   -1.51598
          0.0      0.0        0.0        0.0       -1.49686
```

```
In [5]: prod(diag(U)) # the product of the diagonal elements of U
```

```
Out[5]: 1.4856724724450876
```

Note that this matches `det(A)` (up to roundoff errors in the last few digits).

This immediately gives you a hint of why the determinant is not such a useful computational tool as you might have thought:

- The most efficient way to compute a determinant, in general, is to do Gaussian elimination and then multiply the pivots together.
- Once you have done elimination, you *already* know whether the matrix is singular and you can *already* solve $Ax = b$ efficiently, so the determinant is mostly superfluous.

We'll discuss some actual determinant applications later.

Although we *could* use the "product of the pivots" as the definition of the determinant (at least for matrices), it is more typical to **build up the definition of the determinant from more basic properties**, and to get the product of the pivots as a *consequence*. We will do that now.

3 Defining properties of the determinant

The following three properties are actually sufficient to *uniquely define* the determinant of any matrix, and are taken from Strang's Introduction to Linear Algebra, section 5.1.

Therefore, we don't *derive* these properties: they are axioms that serve to define the determinant operation.

3.1 1. det(I) = 1

It is clear that the identity matrix I is not singular, and all its pivots are 1. A reasonable starting point for defining determinants, therefore, is to require:

- $\det I = 1$ for any $m \times m$ identity matrix I (any m).

For example:

```
In [6]: eye(5)
```

```
Out[6]: 5×5 Array{Float64,2}:
         1.0  0.0  0.0  0.0  0.0
         0.0  1.0  0.0  0.0  0.0
         0.0  0.0  1.0  0.0  0.0
         0.0  0.0  0.0  1.0  0.0
         0.0  0.0  0.0  0.0  1.0
```

```
In [7]: det(eye(5))
```

```
Out[7]: 1.0
```

3.2 2. Sign flips under row exchange

The second key property is:

- If you **swap two rows** in a matrix, the **determinant flips sign**.

For example, with a random 5×5 matrix A:

```
In [8]: A = rand(-3:3, 5,5)
```

```
Out[8]: 5×5 Array{Int64,2}:
         -1  3   2  -1  -2
         -3  3  -3   0  -2
          0  2   0  -2   2
          1  0  -3  -3  -3
          1  1  -1  -3  -2
```

Swapping the first two rows gives the matrix B:

```
In [9]: B = copy(A)
        B[1,:] = A[2,:]
        B[2,:] = A[1,:]
        B
```

```
Out[9]: 5×5 Array{Int64,2}:
         -3  3  -3   0  -2
         -1  3   2  -1  -2
          0  2   0  -2   2
          1  0  -3  -3  -3
          1  1  -1  -3  -2
```

Hence the determinants are equal and opposite:

```
In [10]: det(A), det(B)
```

```
Out[10]: (-27.99999999999998,27.99999999999999)
```

(Up to roundoff errors, of course.)

3.3 3. Linearity in any individual row

The determinant will *not* be a linear operation on the whole matrix: $\det(A+B) \neq \det A + \det B$!! But, we *would* like it to be linear with respect to **operations on individual rows**.

This means two things:

3.3.1 Scaling rows

- If we **multiply a row by a scalar** α, then the **determinant multiplies by** α.

This axiom actually makes a lot of sense if you think about the example of the identity matrix. Multiplying the first row of I by α leads to the matrix:

$$\begin{pmatrix} \alpha & 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

The determinant of this matrix is exactly α! As $\alpha \to 0$, this matrix becomes singular, and the determinant goes to zero at the same rate. It is also consistent with our "product of the pivots" intuitive guess above, because the pivots here are $(\alpha, 1, 1, \cdots)$.

We can also try this with our random matrix A from above. Let's multiply the second row by 2:

```
In [11]: C = copy(A)
         C[2,:] = 2*A[2,:]
         C

Out[11]: 5×5 Array{Int64,2}:
          -1  3   2  -1  -2
          -6  6  -6   0  -4
           0  2   0  -2   2
           1  0  -3  -3  -3
           1  1  -1  -3  -2

In [12]: det(A), det(C)

Out[12]: (-27.99999999999998,-55.99999999999996)
```

As expected, the determinant doubles.

As a consequence of this, if you multiply an *entire* $m \times m$ matrix A by α, we obtain:

- $\det(\alpha A) = \alpha^m \det A$

This is *not* an axiom, it is a *consequence* of the axiom above: we pick up a factor of α for each row that we scale.

For our 5×5 matrix A, this means that $\det(2A) = 2^5 \det A = 32 \det A$:

```
In [13]: det(2A) / det(A)

Out[13]: 32.0
```

3.3.2 Adding a row vector to a row

There is a second property of linearity, corresponding to vector addition:

- If we **add a row vector** r to a row of A, then the determinant becomes $\det(A) + \det(A')$, where A' is the matrix with that row **replaced by** r (with **other rows unchanged**).

This is easier to explain with an example:

$$\det \begin{pmatrix} a+a' & b+b' \\ c & d \end{pmatrix} = \det \begin{pmatrix} a & b \\ c & d \end{pmatrix} + \det \begin{pmatrix} a' & b' \\ c & d \end{pmatrix} .$$

Or, in terms of our matrix A from above, let's add $(1, 2, 3, 4, 5)$ to the first row:

```
In [14]: A

Out[14]: 5×5 Array{Int64,2}:
          -1  3   2  -1  -2
          -3  3  -3   0  -2
           0  2   0  -2   2
           1  0  -3  -3  -3
           1  1  -1  -3  -2

In [15]: [1,0,0,0,0] * [1 2 3 4 5] # = column * row = outer product

Out[15]: 5×5 Array{Int64,2}:
          1  2  3  4  5
          0  0  0  0  0
          0  0  0  0  0
          0  0  0  0  0
          0  0  0  0  0

In [16]: det(A + [1,0,0,0,0] * [1 2 3 4 5])

Out[16]: 307.99999999999994
```

This should be the same as $\det A$ plus the determinant of A with the first row replaced by $(1, 2, 3, 4, 5)$:

```
In [17]: A′ = copy(A)
         A′[1,:] = [1,2,3,4,5] # replace first row
         A′

Out[17]: 5×5 Array{Int64,2}:
           1  2   3   4   5
          -3  3  -3   0  -2
           0  2   0  -2   2
           1  0  -3  -3  -3
           1  1  -1  -3  -2

In [18]: det(A) + det(A′)

Out[18]: 307.99999999999994
```

Yup, it matches (up to roundoff errors, of course).

4 Additional properties of determinants

The following properties can be **derived from the above 3**, and are quite useful to know. Again, the numbering follows Strang, section 5.1:

4.1 4. If two rows are equal, det = 0

It's easy to see why this **follows from property 2**: if we swap two equal rows, the matrix doesn't change, but the determinant must flip sign. But this means:

$$\det A = -\det A \implies \det A = 0$$

For example:

```
In [19]: det([ 1 2 3
               4 5 6
               1 2 3 ])
```

```
Out[19]: 0.0
```

This property also makes sense if our expectation is that the determinant is zero for singular matrices: if two rows are equal, the matrix is singular.

4.2 5. Subtracting a multiple of one row from another doesn't change det

Suppose we take a matrix A, and subtract (or add) a multiple of one row from another. For example:

$$\det \begin{pmatrix} a & b \\ c-\alpha a & d-\alpha b \end{pmatrix} = \det \begin{pmatrix} a & b \\ c & d \end{pmatrix} - \alpha \det \begin{pmatrix} a & b \\ a & b \end{pmatrix} = \det \begin{pmatrix} a & b \\ c & d \end{pmatrix} + 0$$

Here, we applied axiom 3 (linearity), and then property 4 (repeated rows).

The same thing happens for *any* size of matrix.

But this is *precisely* the kind of operation that we perform during Gaussian elimination. It has the crucial implications:

- **Elimination operations** on rows **don't change the determinant**.
- **Gaussian elimination without row swaps doesn't change the determinant**.

And, by axiom 2:

- **Gaussian elimination with row swaps** gives the **same determinant** but with **flipped sign for each row swap**.

For example:

```
In [20]: L, U = lu(A, Val{false}) # elimination without row swaps
         U
```

```
Out[20]: 5×5 Array{Float64,2}:
          -1.0   3.0   2.0  -1.0        -2.0
           0.0  -6.0  -9.0   3.0         4.0
           0.0   0.0  -3.0  -1.0         3.33333
           0.0   0.0   0.0  -0.666667   -9.11111
           0.0   0.0   0.0   0.0        -2.33333
```

```
In [21]: det(A), det(U)
```

```
Out[21]: (-27.99999999999998,-27.99999999999998)
```

4.3 6. A matrix with a row of zeros has det = 0

This is easy to see from axiom 3 (linearity): if we multiply the row of zeros by zero, it doesn't change the matrix but multiplies the determinant by zero, hence:

$$0 \times \det A = \det A \implies \det A = 0$$

For example:

```
In [22]: det([1 2 3
              4 5 6
              0 0 0])

Out[22]: 0.0
```

4.4 7. If A is triangular then det(A) is the product of the diagonal entries

This is another incredibly useful property. To see this, suppose we have an upper-triangular matrix U. Then:

1. Eliminate "upward" above the pivots to get a diagonal matrix D. This doesn't change the determinant by property 5.
2. Pull out each diagonal element by axiom 3 (linearity) until you get the identity matrix I whose determinant is 1 by axiom 1:

$$\det \begin{pmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \alpha_3 & \\ & & & \ddots \end{pmatrix} = \alpha_1 \det \begin{pmatrix} 1 & & & \\ & \alpha_2 & & \\ & & \alpha_3 & \\ & & & \ddots \end{pmatrix} = \cdots = \alpha_1\alpha_2\alpha_3 \cdots \det I = \alpha_1\alpha_2\alpha_3 \cdots$$

   which is precisely the product of the diagonals.

If we have a zero diagonal entry, we can't eliminate upward above it (we can't divide by the diagonal "pivot"). But in that case we end up with a row of zeros after eliminating above the *other* diagonals, and by property 6 we get a zero determinant. So it still matches the product of the diagonal entries.

Similarly for a lower triangular matrix, except that we eliminate "downward".

We already saw an example of this earlier, but let's do it again. We got our U matrix from elimination on A:

```
In [23]: U

Out[23]: 5×5 Array{Float64,2}:
          -1.0   3.0   2.0  -1.0        -2.0
           0.0  -6.0  -9.0   3.0         4.0
           0.0   0.0  -3.0  -1.0         3.33333
           0.0   0.0   0.0  -0.666667  -9.11111
           0.0   0.0   0.0   0.0       -2.33333
```

Its diagonal entries are:

```
In [24]: diag(U)

Out[24]: 5-element Array{Float64,1}:
          -1.0
          -6.0
          -3.0
          -0.666667
          -2.33333
```

The product of these is:

```
In [25]: prod(diag(U))
```

```
Out[25]: -27.99999999999998
```

which matches $\det U$ (and $\det A$):

```
In [26]: det(U), det(A)
```

```
Out[26]: (-27.99999999999998,-27.99999999999998)
```

If we *do* need to compute the determinant, this gives us a very practical way to do it: **compute det(A) by taking the product of the pivots after elimination, with a sign flip for every row swap**.

This is, in fact *exactly* what the Julia `det` function does, as you can check by looking at the source code:

```
In [27]: @which det(A)
```

```
Out[27]: det{T}(A::AbstractArray{T,2}) at linalg/generic.jl:574
```

From the source code, this calls `det(lufact(A))`, which calls:

```
In [28]: @which det(lufact(A))
```

```
Out[28]: det{T,S}(A::Base.LinAlg.LU{T,S}) at linalg/lu.jl:196
```

4.5 8. det(A) = 0 if and only if A is singular

This follows from property 7. Since the determinant is ± the product of the pivots, we get zero if and only if there is a zero pivot, corresponding to a singular matrix.

4.6 9. det(AB) = det(A) det(B)

This is an amazing property of determinants, and probably the least obvious.

A nice way to show this (from Strang's book) is simply to check that

$$\det(AB)/\det(B)$$

satisfies axioms 1,2,3 for A. If it does, then it must be $\det A$, and we are done! Let's check:

1. Identity: If $A = I$, then $\det(AB)/\det(B) = \det(B)/\det(B) = 1$. `[U+2713]`
2. Swaps: If we swap two rows of A, we also swap the *same* two rows of AB, hence $\det(AB)/\det(B)$ flips sign. `[U+2713]`
3. Linearity:

- Scaling a row of A by α scales a row of AB by α, which scales $\det(AB)/\det(B)$ by α. `[U+2713]`
- Adding a row of A to a row of A' (with other rows the same) adds the same rows of AB and $A'B$, so it adds $\det(AB)/\det(B)$ and $\det(A'B)/\det(B)$. `[U+2713]`

Let's try it:

```
In [29]: B = rand(-3:3, 5,5)
```

```
Out[29]: 5×5 Array{Int64,2}:
          -3  -2  -3  -3  -2
          -3  -3  -1  -1  -1
          -3   1  -1  -2   1
           2  -1   3  -1   3
           2   2   1   0   3
```

```
In [30]: det(A), det(B)
```

```
Out[30]: (-27.99999999999998,-175.00000000000003)
```

```
In [31]: det(A*B), det(A)*det(B)
```

```
Out[31]: (4899.999999999959,4899.999999999997)
```

4.6.1 Matrix inverses

This rule has important consequences for matrix inverses. First:

$$\det(A^{-1}) = 1/\det(A)$$

Proof: $1 = \det(I) = \det(AA^{-1}) = \det(A)\det(A^{-1})$.

For example:

```
In [32]: det(inv(A)), 1/det(A)
```

```
Out[32]: (-0.03571428571428567,-0.03571428571428574)
```

Recall from last lecture that XAX^{-1} corresponds simply to a **change of basis** from A. (We will later call this a **similar matrix** to A). Now we know:

$$\det(XAX^{-1}) = \det(X)\det(A)\det(X^{-1}) = \det(A)\ .$$

That is, a **change of basis doesn't change the determinant**.

4.7 10. det(A[U+1D40]) = det(A)

This is another non-obvious, but very important, property of determinants. It is relatively easy to see from properties 7 and 9, however.

In particular, factorize $PA = LU$, or $A = P^T LU$. Then, from property 9:

$$\det(A) = \det(P^T)\det(L)\det(U) = \det(P^T)\det(U) = \det(P^T) \times (\text{product of pivots})\ ,$$

where we have used the fact that $\det L = 1$ since the diagonal entries of L are all 1's. But we also have:

$$\det(A^T) = \det(U^T L^T P) = \det(U^T)\det(L^T)\det(P) = \det(P) \times (\text{product of pivots})$$

where we have used the fact that U and U^T have the same diagonal entries and hence the same determionant (U is upper triangular so U^T is lower triangular), similarly for L.

So the only difference is $\det P$ vs $\det P^T$. But recall that P is a permutation matrix: just a re-ordering of the rows of I, so by axioms 1 and 2 we must have

$$\det P = \pm 1$$

(depending on how many row swaps there were). Furthermore, recall that $P^T P = I$ (the permution P is *orthogonal* since its rows/columns are orthogonal (a re-ordering of the rows/columns of I). By property 9, this means that

$$1 = \det I = \det(P^T P) = \det(P^T)\det(P)$$

Since $\det P = \pm 1$, this means that $\det(P^T) = \det(P)$.

It follows that

$$\det A = \det A^T = \pm(\text{product of pivots})$$

For example:

```
In [33]: det(A), det(A')
```

```
Out[33]: (-27.99999999999998,-27.999999999999968)
```

5 Useful applications of determinants

Ignoring formulas (e.g. Cramer's rule, a formula for A^{-1} — see Strang, section 5.3) that are mainly useful for tiny matrices, here are some examples of real usages of determinants **even for large matrices**:

- Understanding **eigenvalues**: determinants will turn eigenvalues into polynomial roots, and since we know about polynomial roots, that tells us a lot about eigenvalues. (This is *not* how eigenvalues are *computed* in practice, however!)

- There is also something called a nonlinear eigenproblem, arising in many science and engineering problems, in which the determinant plays a basic conceptual role. Again, however, computational methods typically avoid computing determinants explicitly except for tiny matrices.

- Proofs: Determinants show up in a lot of proofs in matrix theory, because they reduce matrices to numbers that have nice properties and are easy to reason about. One also often sees things like the adjugate matrix and the Cayley–Hamilton theorem, both related to determinants.

- That is, we often use determinants to help use understand and derive things in linear algebra, even if the final result doesn't require us to *compute* the determinant for any practical purpose.

- Jacobian factors: in multivariable calculus, a factor of $|\det J|$ arises when you perform a *change of variables* in integration, where J is a Jacobian matrix.

- The reason a determinant arises here is that, more generally, **det(A) is the volume of a parallelepiped** ("box") whose edges are given by the columns of A.

- Integration may sound like something that only happens in a few dimensions (= tiny matrices J), but extremely high dimensional (even infinite-dimensional) integrals appear in statistics, quantum field theory, bioinformatics, and other fields.

- High-dimensional Gaussian integrals often arise in **statistics** and related areas of science (e.g. quantum field theory), and the inverse of the square root of a determinant appears in the answer. Often, one wants the logarithm of the result, in which case what arises is the **log determinant** $\log\det A$, an important matrix function.

This is no doubt an incomplete list. Nevertheless, although determinants are a much more marginal topic in modern linear algebra than they were in the 19th century, they have hardly disappeared.

6 A "Simple" But Horrible Formula

You probably learned a neat formula for the determinant of a 2×2 matrix at some point:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

You might have even learned a formula for 3×3 matrices. You might be hoping, therefore, that there would be an extension of this "nice" formula (which seems a lot easier than doing elimination to get pivots) to arbitrary matrices. There is!

Here it is (see Strang, section 5.2):

$$\det A = \sum_{\text{permutations } p} \text{sign}(p) \times (\text{product of diagonals of } A \text{ with columns permuted by } p)$$

The important thing to know is that you have to consider **all permutations (re-orderings)** of $(1, 2, 3, \ldots, n)$. (The sign of the permutation corresponds to the number of swaps it involves.) There are $n! = n(n-1)(n-2)\cdots 1$ (*n factorial*) re-orderings.

That means that this formula requires $\sim n \times n!$ scalar operations, which is **worse than exponential** in n. This is **far more expensive than elimination** ($\sim n^3$), making this formula **computationally useless** for $n > 3$.

(There is also *another* computationally useless formula involving minors and cofactors; see Strang, section 5.2.)

The permutation formula is still sometimes useful *conceptually*, however.