Lecture 5: MATRICES AND LINEAR FUNCTIONS *

10-606 MATHEMATICAL FOUNDATIONS FOR MACHINE LEARNING

1 Functions on vector spaces

Previously, we've seen that we can define many different vector spaces, with many different properties. Given all of these vector spaces, it makes sense to look at functions that map between them. For example, we could define a function $r \in \mathbb{R}^2 \to \mathbb{R}^2$ that takes a 2d vector and rotates it 30° clockwise around the origin. Or, we could define a function $p \in \mathbb{R}^5 \to \mathbb{R}^2$ that extracts the first and fourth components of a vector (discarding the second, third, and fifth).

When dealing with functions like these, it's a good idea always to keep track of the types of expressions. Type-checking can catch lots of simple errors: for example, with the definitions above, $r(p(\mathbf{v}))$ makes sense as long as $\mathbf{v} \in \mathbb{R}^5$. But, the expression $p(r(\mathbf{v}))$ doesn't make sense, no matter what the type of $\mathbf{v}$ is.

1.1 Functionals and operators

There are two special names that are worth knowing for specific kinds of functions between vector spaces. First, if the value of the function is a real number, the function is called a *functional*. Second, a mapping from a vector space to itself is called an *operator*, e.g., the "rotate 30°" function above is an operator on $\mathbb{R}^2$. As another example, the differential $\frac{d}{dx}$ is an operator on a vector space of functions, mapping a function like x^2 to another function like $2x$. (we'll have a lot more to say on differentials next week so stay tuned).

2 Linear functions

A function between vector spaces $f \in U \to V$ is called *linear* if it satisfies

$$f(a\mathbf{x} + b\mathbf{y}) = af(\mathbf{x}) + bf(\mathbf{y})$$

for all vectors $\mathbf{x}, \mathbf{y} \in U$ and all scalars $a, b \in \mathbb{R}$. That is, we can take addition and scalar multiplication in the input space U and turn them into addition and scalar multiplication in the output space V.

Another way to say the same thing is that f is a *vector-space homomorphism*. A homomorphism is a function that preserves structure, in this case the behavior of the key vector-space operations of addition and scalar multiplication.

We've already seen some examples of linear functions, e.g., the differential $\frac{d}{dx}$ is a linear operator, since for instance $\frac{d}{dx}(f + g) = \frac{d}{dx}f + \frac{d}{dx}g$ (the sum rule). Another well-known example of a linear function is multiplication by a matrix: if $A \in \mathbb{R}^{n\times m}$, then the function $f \in \mathbb{R}^m \to \mathbb{R}^n$ defined by $f(\mathbf{x}) = A\mathbf{x}$ is linear.

If $f \in V \to \mathbb{R}$ is a linear functional, and if V is a complete inner product space, then there always exists some vector $\mathbf{g} \in V$ such that $f(\mathbf{x}) = \langle \mathbf{g}, \mathbf{x} \rangle$ for all $\mathbf{x} \in V$. This property, that every linear functional can be implemented as an inner product, is another useful property of $\mathbb{R}^n$ that extends to more-general spaces.

One common source of confusion is the difference between linear and *affine* functions. Consider the function $f(x) = 3x + 2$, where $x \in \mathbb{R}$. Its plot is a line, but if we check the above definition, f is not linear: for example, $f(3) = 11$, but $f(3 \cdot 2) = 20 \neq 11 \cdot 2$. Instead f is *affine*: it is a linear function plus a constant.

*Compiled on Wednesday 7th June, 2023 at 14:29

Knowledge check

1. **Select one**: Consider the identity function: $I(\mathbf{x}) = \mathbf{x} \; \forall \; \mathbf{x}$. Which of the following *best* describes I?
   a) Functional
   b) Linear functional
   c) Operator
   d) Linear operator
   e) None of the above
   - **Answer**: d, I takes inputs from some vector space and always returns values from that same vector space, making it an operator. To show that I is linear, consider

   $$I(a\mathbf{x} + b\mathbf{y}) = a\mathbf{x} + b\mathbf{y} = aI(\mathbf{x}) + bI(\mathbf{y})$$

2. **Select one**: Consider a *definite* integration function: specifically, this function takes as input functions f defined over $[0, 1]$ and returns $\int_0^1 f(x)dx$. Which of the following *best* describes this definite integration function?
   a) Functional
   b) Linear functional
   c) Operator
   d) Linear operator
   e) None of the above
   - **Hint**: First, think about what the type of $\int_0^1 f(x)dx$ is. Then, try to simplify $\int_0^1 (af(x) + bg(x))\, dx$.
   - **Answer**: b, the output of a definite integral is some real value, making the definite integration function a functional. To see that it is a linear functional, observe that $\int_0^1 (af(x) + bg(x))\, dx = \int_0^1 af(x)dx + \int_0^1 bg(x)dx = a\int_0^1 f(x)dx + b\int_0^1 g(x)dx$.

3 Matrices and linear functions

Recall that we can use a basis B for some vector space V to derive a coordinate representation for each vector $\mathbf{v} \in V$. We can also make a coordinate representation for linear functions out of a coordinate representation for vectors. Suppose we have a linear function $L \in U \to V$, a basis $\mathbf{b}_1 \dots \mathbf{b}_n$ for U, and a basis $\mathbf{c}_1 \dots \mathbf{c}_m$ for V. We can apply L to one of our vectors $\mathbf{b}_j \in U$, and expand the result $L\mathbf{b}_j$ in terms of our basis for V: we pick coefficients $\ell_{1j}, \ell_{2j}, \dots$ so that

$$L\mathbf{b}_j = \ell_{1j}\mathbf{c}_1 + \ell_{2j}\mathbf{c}_2 + \dots + \ell_{mj}\mathbf{c}_m$$

We can do the same for all of the other basis vectors in U, expanding each one's image under L in terms of our basis for V. The resulting coefficients ℓ_{ij} form a basis representation for L: we can map the abstract vector space of linear operators to the concrete vector space of mn-dimensional real vectors.

We typically write out the coordinates of L as a matrix:

$$L \in U \to V \quad \leftrightarrow \quad \begin{pmatrix} \ell_{11} & \dots & \ell_{1n} \\ \vdots & \ddots & \vdots \\ \ell_{m1} & \dots & \ell_{mn} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

That is, instead of writing our mn coordinates as one long vector in $\mathbb{R}^{mn}$, we write them as an $m \times n$ matrix.

This representation agrees with our usual idea of matrix-vector multiplication: take a vector $\mathbf{x}$ whose coordinate representation is $u_1\mathbf{b}_1 + u_2\mathbf{b}_2 + \dots + u_m\mathbf{b}_m$. Then by linearity,

$$L\mathbf{x} = L\sum_{i=1}^{m} u_j\mathbf{b}_j = \sum_{j=1}^{m} u_j L\mathbf{b}_j$$

Expanding each term $L\mathbf{b}_j$, we get

$$L\mathbf{x} = \sum_{j=1}^{m} u_j \sum_{i=1}^{n} \ell_{ij}\mathbf{c}_i = \sum_{i=1}^{n} \left[\sum_{j=1}^{m} u_j \ell_{ij} \right] \mathbf{c}_i$$

The ith coordinate of $L\mathbf{x}$ is the coefficient of $\mathbf{c}_i$ in the above expression, namely $v_i = \sum_{j=1}^{m} u_j \ell_{ij}$, which is exactly the ith entry of the vector we get by multiplying the matrix coordinate representation of L by the vector coordinate representation of $\mathbf{x}$.

As was the case for vectors, picking different bases will result in different coordinate representations for L. So, we can get two completely different matrices that represent the same linear function: as before, the function hasn't changed, just our representation of it.

It's helpful to think of the matrix $\ell = (\ell_{ij})_{ij}$ as a linear function in $\mathbb{R}^m \to \mathbb{R}^n$, as compared to L, which is a linear function in $U \to V$. Our coordinate representations for U, V, and $U \to V$ agree in such a way that we can perform corresponding operations in the abstract and concrete vector spaces:

$$L\mathbf{x} = \mathbf{y} \quad \leftrightarrow \quad \ell\mathbf{u} = \mathbf{v}$$

3.1 Range and nullspace

The range of a function $f \in U \rightarrow V$ is the set of all its possible outputs:

$$\text{range}(f) = \{f(\mathbf{x}) \mid \mathbf{x} \in U\} \subseteq V$$

Suppose that f is linear. Then its range is a subspace of V, possibly all of V. In fact, if we have a matrix L that represents f, then the range of f is equal to the span of the columns of L. This follows directly from the expression for matrix multiplication and the definition of the span: $L\mathbf{x}$ is a linear combination of the columns of L, with weights given by the entries of $\mathbf{x}$.

The *rank* of a linear function is defined as the dimension of its range. Similarly, the rank of a matrix is defined as the rank of its corresponding linear function. For example, if our function is

$$f\left[\begin{pmatrix} x \\ y \end{pmatrix}\right] = \begin{pmatrix} x + y \\ 2x + 2y \end{pmatrix}$$

then its range is the set of all vectors whose second component is twice the first, $\text{span}((1, 2)^T)$. This is a one-dimensional set, so the rank of f is 1.

Suppose now that f is an operator, so that $U = V$. If the rank of f is less than the dimension of V, then there will be some vectors $\mathbf{x} \in V$ such that $f(\mathbf{x}) = 0$. The set of all such $\mathbf{x}$ is the *nullspace* or *kernel* of f:

$$\text{null}(f) = \{\mathbf{x} \mid f(\mathbf{x}) = 0\}$$

The dimension of the nullspace is called the *nullity* of f. The nullspace and nullity of a matrix are defined similarly.

For example, for the function defined above, the vector

$$\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

is in the nullspace; in fact the nullspace is equal to the span of this vector, so the nullity of f is 1.

Knowledge check

1. **Short answer:** Suppose we have two linear operators f, g on a vector space V of dimension d. Suppose the rank of f is $r \leq d$ and the rank of g is $s \leq d$. What can we say about the rank of fg, the function that applies f to the result of applying g to $\mathbf{x} \in V$?
   - **Hint**: Break the problem into cases; start by considering what happens if $r < s$ and then, what happens if $s \leq r$.
   - **Answer**: The rank of fg is at most $\min(r, s)$. Intuitively, if either g "shrinks" our vector space to some smaller range i.e., one with lower dimension, than f cannot increase that range by say adding dimensions (a direct result from the fact that f is an operator). At best f can maintain the new range but it could also potentially decrease the dimensionality further.

4 Finding coordinate representations

Suppose we have an inner product space V with a basis $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots \mathbf{b}_n\} \subseteq V$. We noted above that any abstract vector $\mathbf{x} \in V$ has a unique coordinate representation in terms of B; this coordinate representation is a concrete vector $\mathbf{u} \in \mathbb{R}^n$. But how exactly can we compute $\mathbf{u}$?

Since $\mathbf{x}$ and B are vectors in an abstract inner product space, we can only use the abstract inner product space interface to interact with them: for example, we can add pairs of vectors or take inner products between them. We will use this ability to write down a system of linear equations that the coordinates $\mathbf{u}$ must satisfy. We can then solve this system to find the desired coordinates.

To get our system of linear equations, suppose we take the inner product between $\mathbf{x}$ and one of the basis vectors $\mathbf{b}_j$. By linearity we have

$$\langle \mathbf{x}, \mathbf{b}_j \rangle = \left\langle \sum_{i=1}^{n} u_i \mathbf{b}_i,\ \mathbf{b}_j \right\rangle = \sum_{i=1}^{n} u_i \langle \mathbf{b}_i, \mathbf{b}_j \rangle$$

Define $g_j = \langle \mathbf{x}, \mathbf{b}_j \rangle$ and $G_{ji} = \langle \mathbf{b}_i, \mathbf{b}_j \rangle$. The above equation then becomes

$$g_j = \sum_{i=1}^{n} G_{ji} u_i$$

or in matrix form

$$\mathbf{g} = G\mathbf{u}$$

So, we can compute G and $\mathbf{g}$ by taking inner products between pairs of vectors; then we can solve this linear system to find the coordinates $\mathbf{u}$.

The matrix G satisfies a couple of interesting properties: first, since the inner product is symmetric, the matrix G is symmetric, $G_{ij} = G_{ji}$. Second, it is *positive definite*, a property that we'll define later. These two properties guarantee that the above system of equations has a unique solution.

Once we have our system $G\mathbf{u} = \mathbf{g}$, there are a variety of algorithms for solving it to find $\mathbf{u}$. Typically, the best approach is to hand the system to an appropriate library function, e.g., `scipy.linalg.solve` in Python. If you need to solve a small system by hand, the best method is probably Gaussian elimination: repeatedly eliminate a variable by adding multiples of one equation to all of the others.

In general, you should *not* try to solve the system by inverting the matrix G i.e., computing $\mathbf{u} = G^{-1}\mathbf{g}$. While this works with exact arithmetic, it can cause all sorts of problems if we try to do it with the approximate arithmetic that happens in a CPU or GPU. (We'll give more details later.)

Knowledge check

1. **True or False:** Changing the basis B will change the coordinate representation of $\mathbf{x}$ but the matrix G and the vector $\mathbf{g}$ will be the same, regardless of what basis we use.
   - **Answer**: False, changing the basis B will cause at least one of the terms in G and $\mathbf{g}$ to change as the inner products $\langle \mathbf{x}, \mathbf{b}_j \rangle$ and $\langle \mathbf{b}_i, \mathbf{b}_j \rangle$ depend on the specific basis vectors. Another way to see this is that if G and $\mathbf{g}$ didn't change, then the solution to $\mathbf{g} = G\mathbf{u}$ would also stay constant, a contradiction to our assertion that the coordinate representation would change under a different basis.