Some Notes on Advanced Calculus

Nathan Ratliff

April 22, 2014

Abstract

This document briefly discusses the pragmatic side of Advanced Calculus. We review notational conventions, some tricks of the trade, and some intuition behind the Inverse Function Theorem.

1 Derivative conventions

With functions $f : \mathbb{R}^n \to \mathbb{R}$ that take a multidimensional space to just a single scalar value there's some accounting we need to worry about then discussing derivatives. When we want to know how the function changes with changes to the input, we need to know how it changes for each individual dimension of that input. These calculations are the function's partial derivative, and by convention we often stack them up in column vector and call them collectively the *gradient*:

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}. \tag{1}$$

Notably, we stack the partial derivatives for this single output one on top of the other.

Since these partials are stacked one atop another for a *single* output, one might expect that for vector valued functions (functions with multiple outputs) we would generalize that idea and create a *matrix* of partials where each column corresponds to a different output. But that's not what we do.[1] And that change in convention can cause a good deal of confusion. This section discusses what *is* the most common convention these days and why that convention is actually quite convenient.

Suppose $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^m$ is a vector valued function from an n-dimensional space to an m-dimensional space, and denote its action as $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{y}$. We can

[1] Admittedly some texts do use matrices of column vector gradients, but we'll see why that's not really as nice. By far, the convention we describe here is the most common convention found throughout the machine learning and control literature.

view this vector-valued function as a set of m separate scalar valued functions $g_i : \mathbb{R}^n \to \mathbb{R}$ so that

$$\boldsymbol{g} = \begin{pmatrix} g_1(\boldsymbol{x}) \\ g_2(\boldsymbol{x}) \\ \vdots \\ g_m(\boldsymbol{x}) \end{pmatrix}. \tag{2}$$

Now, for every output function g_i, we have n partial derivatives, one for each input variable $x_1, \ldots, x_n$. Our convention in this setting is to shape the collection of all partial derivatives into an $m \times n$ matrix of the form

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial x_2} & \cdots & \frac{\partial g_m}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \nabla g_1^T \\ \nabla g_2^T \\ \vdots \\ \nabla g_m^T \end{pmatrix}. \tag{3}$$

We call this matrix the Jacobian and denote it using the notation $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$. Notice that it's actually the *rows* of this matrix that hold each output's *transposed* gradient. That seems to be a flip of convention. Why the transpose?

A matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is map transforming vectors in an n-dimensional space $\mathbb{R}^n$ to a vector in an m-dimensional space $\mathbb{R}^m$, and by convention we typically write $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ transforming column vectors into other column vectors. We can think of the n-dimensional vector $\boldsymbol{x}$ being shoved into the matrix from the top (n, the number of columns, is the number of inputs). $\boldsymbol{x}$ churns through the matrix $\boldsymbol{A}$ and is ultimately thrown out as a new m-dimensional vector from the left of the matrix:

$$\begin{array}{rc} & \text{Inputs} \quad \leftarrow \\ \text{Outputs} \leftarrow & \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \end{array} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Conveniently, we can think of the Jacobian matrix $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$ as having the same interpretation. The inputs line the top of the matrix, and the outputs line the left side of the matrix. The ijth entry of the matrix itself tells us how tweaking the jth input affects the i output of the original function; again we have inputs corresponding to the j index (columns) and outputs corresponding to the i index (rows).

This input-output convention fits very nicely with the way the chain rule works. Suppose now there's another function $\boldsymbol{h} : \mathbb{R}^k \to \mathbb{R}^n$ that takes k-dimensional vectors $\boldsymbol{u} \in \mathbb{R}^k$ to vectors $\boldsymbol{x} \in \mathbb{R}^n$ in the domain (input space)

of $\boldsymbol{g}$. For each g_i the chain rule says

$$\frac{\partial}{\partial u_k} g_i(h(\boldsymbol{u})) = \sum_{j=1}^{n} \frac{\partial g_i}{\partial x_j} \frac{\partial h_j}{\partial u_k}, \tag{4}$$

so if you squint hard enough you can convince yourself that the individual Jacobians of $\boldsymbol{g}$ and $\boldsymbol{h}$ relate to the overall Jacobian of the composed function $\boldsymbol{f} = \boldsymbol{g} \circ \boldsymbol{h} : \mathbb{R}^k \to \mathbb{R}^m$ in the following way[2]

$$\frac{\partial}{\partial \boldsymbol{u}} (\boldsymbol{g} \circ \boldsymbol{h}) = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{u}}. \tag{5}$$

Note that the matrices line up nicely without having to take any transposes, and we could have continued the chain further to the right to account for as many function compositions as necessary since the rule works recursively.

Now, that's convenient. The matrix $\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{u}}$ tells us how h_j changes when we tweak u_k, and then in turn $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$ tells us how the final output g_i changes when the input x_j, which h_j feeds into, changes. Since the outputs of $\boldsymbol{h}$ become the inputs of $\boldsymbol{g}$, the derivatives align such that the outputs of $\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{u}}$ become the inputs of $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$. This relation is furthered when take time derivatives (which, really, we can also view as just another application of the chain rule since $\boldsymbol{u}$ is now a vector valued function of time):

$$\frac{d}{dt} (\boldsymbol{g} \circ \boldsymbol{h})(\boldsymbol{u}) = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{u}} \dot{\boldsymbol{u}}. \tag{6}$$

We can again think through this formula intuitively in terms of the inputs and outputs of the constituent matrices. $\boldsymbol{u}(t)$ is the first function we encounter in the composition, so it makes sense that the rightmost factor in the time derivative tells us how the underlying inputs to the entire system are changing with time. Then, since $\boldsymbol{u}$ feeds into $\boldsymbol{n}$ and in turn the inputs to $\boldsymbol{h}$ churn through the function to become the outputs of $\boldsymbol{h}$, the middle term tells us how changes to the inputs of $\boldsymbol{h}$ become changes to the outputs of $\boldsymbol{h}$. Combined, $\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{u}} \dot{\boldsymbol{u}}$ tells us how the changes in $\boldsymbol{u}$ with respect to time, which are changes to the inputs of $\boldsymbol{h}$, become changes to the outputs of $\boldsymbol{h}$. And finally, since outputs of $\boldsymbol{h}$ feed into $\boldsymbol{g}$ as inputs, the final multiplication by $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$ tells us how those changes to the outputs of $\boldsymbol{h}$ ultimately become changes to the outputs of $\boldsymbol{g}$. The outputs of $\boldsymbol{g}$ are the outputs of the entire system, so we're done.

In other words, since matrices take inputs along the top to outputs along the left side, the way in which inputs lead to outputs in $\boldsymbol{g} \circ \boldsymbol{h}(\boldsymbol{u})$ dictates the final structure of how the matrices of partial derivatives should be laid out. The convention we chose above allows us to leverage our intuition of how a matrix should behave to understand the flow of information through the chain rule.

[2]See Appendix A for a high level discussion of why the chain rule becomes matrix multiplication.

2 The inverse function theorem from 30,000 feet

The inverse function theorem says (loosely) that if the Jacobian $\frac{\partial \phi}{\partial \boldsymbol{x}}$ of a function $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is invertible at some $\boldsymbol{x}$, then an inverse function exists in some neighborhood around $\phi(\boldsymbol{x})$. Moreover, explicitly, the Jacobian of the inverse function at $\boldsymbol{y} = \phi(\boldsymbol{x})$ is given by the inverse of the Jacobian at $\boldsymbol{x}$. i.e.

$$\frac{\partial}{\partial \boldsymbol{y}} \phi^{-1}(\underbrace{\phi(\boldsymbol{x})}_{=\boldsymbol{y}}) = \left(\frac{\partial \phi}{\partial \boldsymbol{x}}(\boldsymbol{x}) \right)^{-1}. \tag{7}$$

This section explores the intuition around that theorem (although not its proof—there are a number of good resources for that on the web). We'll see here that it can be understood intuitively as just an application of linear algebra.

Consider the first-order Taylor approximation to ϕ at the point $\boldsymbol{x}_0$:

$$\boldsymbol{y} = \phi(\boldsymbol{x}_0) + \frac{\partial \phi}{\partial \boldsymbol{x}} (\boldsymbol{x} - \boldsymbol{x}_0). \tag{8}$$

If this approximation is "good enough" in a small region around $\boldsymbol{x}_0$, we can assume that the properties of the linearization are essentially the same as the properties of the actual function in that region. (If we're careful, which we won't be here, we can even argue that they limit to be the same as we approach $\boldsymbol{x}_0$.)

This linear approximation is invertible if and only if $\frac{\partial \phi}{\partial \boldsymbol{x}}$ is full rank, and it's inverse in that case is

$$\boldsymbol{x} = \boldsymbol{x}_0 + \left(\frac{\partial \phi}{\partial \boldsymbol{x}} \right)^{-1} (\boldsymbol{y} - \underbrace{\phi(\boldsymbol{x}_0)}_{\boldsymbol{y}_0}), \tag{9}$$

which is the linearization of a function from the range to the domain with Jacobian $\left(\frac{\partial \phi}{\partial \boldsymbol{x}} \right)^{-1}$. Thus, it makes sense that the function is invertible if and only if its Jacobian is and that the inverse function's Jacobian is the inverse of the forward function's Jacobian.

A Why the chain rule really becomes matrix multiplication

Most Advanced Calculus textbooks present this result almost as though it were a magical, coincidental, fortuitous result that just happened to work out because the stars aligned and fate smiled down at the first soul brave enough to tackle writing it out. But, as with most apparent coincidences in math, that's not the case. It's a manifestation of something more fundamental.

Advanced analysis text, especially those leading up to calculus on manifolds, present a lot of the rules of calculus using a basis-free approach. That means rather than representing n-dimensional vectors using collections of n numbers,

the expositions leave them simply as abstract elements of a set that satisfies enough axioms to be considered a linear vector space. If we so choose, we can always represent a vector using the n coefficients of that vector's description in some n-dimensional basis for the space, but since the choice of basis is largely arbitrary, these texts study the underlying properties of linear algebra and calculus abstractly as much as possible without fixing a particular basis in advance.

In this setting, the abstract chain rule says that the total derivative of a composed map $\boldsymbol{g} \circ \boldsymbol{h}$ is a linear transform consisting of a product of the linear transforms that come individually from $\boldsymbol{g}$ and $\boldsymbol{h}$. Fundamentally, the chain rule just relates overall derivatives to products of constituent derivatives. Once we ultimately calculate these results for particular choices of bases, products of linear transforms then manifest as matrix multiplication rules that tell us how the *coefficients* of a vector in the domain turn into *coefficients* of the transformed vector in the range. Thus, our matrix multiplication of Jacobian matrices is actually just a bookkeeping trick to track how coefficients of particular vector representations transform under the more fundamental basis-free linear transformations representing the underlying derivatives.

Matrix multiplication doesn't just happen to work out because it was a clever way to compactly represent all the sums that crop up when applying the chain rule to composed vector-valued multivariate functions, the matrix multiplication arises because it represents how the coefficients under our particular (Cartesian) choice of bases transform under the abstract chain rule governing the process.