1 Basic Vector Operations

The basic structure of multivariable calculus is the vector. By convention we assume in this class that vectors are column vectors:

\[ \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_n \end{bmatrix} \]

The transpose of this vector, \( \mathbf{v}^T \), is a row vector:

\[ \mathbf{v}^T = [v_1 \ v_2 \ v_3 \ \cdots \ v_n] \]

The inner product, also known as the dot product, reduces two vectors of equal length to a scalar:

\[ \mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y} = \sum_i x_i y_i \]

In contrast, the outer product takes two vectors and produces an \( n \times n \) matrix:

\[ \mathbf{x} \times \mathbf{y}^T = \mathbf{xy}^T = \begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_n \\ x_2y_1 & x_2y_2 & \cdots & x_2y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_ny_1 & x_ny_2 & \cdots & x_ny_n \end{bmatrix} \]

2 Multivariable Functions and Vector Calculus

A multivariable function takes a number of variables (or a vector) as parameters and returns a single value. In domain terms, a multivariable function \( f(\mathbf{x}) \) maps from an \( n \)-dimensional vector space down to a scalar domain. The gradient is the basic vector derivative operation. Given a scalar function \( f(\mathbf{x}) \),

\[ \nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \]
yields a vector representing the direction and the rate of change of the function \( f \) within \( \mathbb{R}^n \)-space.

**Example 1:** Let \( f(x) = 1^T x = \sum x_i \). Then the gradient of \( f \) is

\[
\nabla f(x) = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = 1
\]

**Example 2:** Let \( f(x) = x^T x = \sum x_i^2 \). Then the gradient of \( f \) is

\[
\nabla f(x) = \begin{bmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_n \end{bmatrix} = 2x
\]

The product and chain rules hold when dealing with gradients and vector-valued functions, but looks slightly different:

- **Product Rule:** \( \nabla (f(x)g(x)) = f(x)\nabla g(x) + \nabla f(x)g(x) \), where \( f \) and \( g \) are both vector-to-scalar functions (\( f : \mathbb{R}^n \to \mathbb{R} \) and \( g : \mathbb{R}^n \to \mathbb{R} \)).

- **Chain Rule:** \( \frac{\partial}{\partial t} f(g(t)) = \nabla f(g(t))^T \frac{\partial g}{\partial t} \), where \( f(x) \) is a vector-to-scalar function and \( g(t) \) is a standard one-parameter vector-valued function (\( f : \mathbb{R}^n \to \mathbb{R} \) and \( g : \mathbb{R} \to \mathbb{R}^n \)). Note that in because \( g \) is a vector-valued function, the partial derivative \( \frac{\partial g}{\partial t} \) is itself a vector: \( \frac{\partial g}{\partial t} = \left[ \frac{\partial g_1}{\partial t} \quad \frac{\partial g_2}{\partial t} \quad \cdots \quad \frac{\partial g_n}{\partial t} \right]^T \).

It is important to note the vector operations that make these rules work. In the case of the Product Rule, the input functions \( f(x) \) and \( g(x) \) are both scalar, but their gradients \( \nabla f \) and \( \nabla g \) are vectors. Thus, \( \nabla (fg) \) is the sum of two scalar-multiplied vectors, and is therefore a vector. Similarly, \( \nabla f(g(t)) \) and \( \frac{\partial g(t)}{\partial t} \) are both vectors, and so their dot product is the scalar one would expect from a partial derivative of a scalar function.

Gradients are very useful when plotted on a map of the variable field, such as a contour map. The gradient points in the direction of the steepest rate of change of \( f(x) \) as one moves up and down the variable axes. On a contour map of a hill, for instance, this represents the direction of fastest ascent. As with scalar derivatives, \( \nabla f(x) = 0 \) when the function \( f \) is at a local extremum (maximum or minimum).

### 3 Matrices, Eigenvalues and Eigenvectors

Let \( A \) be an \( n \times n \) matrix, \( b \) be an \( n \)-element vector, and \( c \) be a scalar. The function

\[
f(x) = \frac{1}{2} x^T Ax + b^T x + c
\]

is a scalar function. The term \( x^T Ax \) can be thought of as the *curvature* in direction \( x \). If \( A \) is symmetric (i.e., \( a_{i,j} = a_{j,i} \)), then \( \nabla f = Ax + b \). If \( A \) is not symmetric, then \( \nabla f = A'x + b \), where \( A' = \frac{1}{2}A + A^T \).

As a side note, If \( Ax = \lambda x \) where \( \lambda \) is some scalar value, then the vector \( x \) is an *eigenvector* of \( A \) and \( \lambda \) is an *eigenvalue* of \( A \). This will come up again later.
4 Jacobians and Hessians

So far, we have assumed that our function \( f \) has a scalar value. But what if we have a vector-valued function \( f : \mathbb{R}^m \rightarrow \mathbb{R}^n \)? This function takes an \( m \)-element input vector and returns an \( n \)-element output vector. The gradients of each vector element \( f(x)_i \) form the rows of a special \( m \times n \) matrix called the Jacobian:

\[
J = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_m}
\end{bmatrix}
\]

Armed with the Jacobian, we can now express the chain rule for a vector-valued multivariable function:

\[
\frac{\partial}{\partial t} f(g(t)) = J \frac{\partial g}{\partial t}
\]

Another useful matrix is the Hessian \( \nabla^2 f \), which is a matrix of a scalar-valued function \( f(x) \)'s second-order derivatives:

\[
\nabla^2 f = \begin{bmatrix}
\frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2}
\end{bmatrix}
= \begin{bmatrix}
\frac{\partial^2 f}{\partial x_i \partial x_j}
\end{bmatrix}_{ij}
\]

**Definition:** A matrix \( A \) is **positive semidefinite** if \( \forall x \ x^T A x \geq 0 \). This also means that all the eigenvalues of \( A \) are positive. This definition is important because if \( f \) is at a maximum, then \( -\nabla^2 f \) is positive semidefinite. Similarly, \( \nabla^2 f \) is positive semidefinite when \( f \) is at a minimum.

**Example:** Let \( f(x) = \frac{1}{2} x^T A x = x_1^2 + x_2^2 \), and let \( A = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \). To find the extrema, we calculate the gradient \( \nabla f = x \). Setting \( \nabla f = x \) to 0, we find the local extremum at the origin. To determine the orientation of this extremum, we compute the Hessian:

\[
\nabla^2 f = \begin{bmatrix}
\frac{\partial^2 (x_1^2 + x_2^2)}{\partial x_1 \partial x_j}
\end{bmatrix}_{ij} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\]

Since all elements in \( \nabla^2 f \) are positive, we know that the extremum is a minimum.

**Example:** Let \( f(x) = \frac{1}{2} x^T A x = -x_1^2 - x_2^2 \), and let \( A = -I = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \). To find the extrema, we calculate the gradient \( \nabla f = x \). Setting \( \nabla f = x \) to 0, we find the local extremum at the origin. To determine the orientation of this extremum, we compute the Hessian:

\[
\nabla^2 f = \begin{bmatrix}
\frac{\partial^2 (-x_1^2 - x_2^2)}{\partial x_1 \partial x_j}
\end{bmatrix}_{ij} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}
\]

Since all elements in \( \nabla^2 f \) are negative, we know that the extremum is a maximum.

5 Newton’s Method

We now have all the tools we need for Newton’s method of approximating a vector-valued function. This is basically the second-order expansion of a Taylor series about some point \( x_0 \).
\[ \hat{f} = f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2} (x - x_0)^T \nabla^2 f(x_0) (x - x_0) \]

The gradient of this approximation is:

\[ \nabla \hat{f} = \nabla^2 f(x_0) (x - x_0) + \nabla f(x_0) \]

Remember that this is an approximation about a point \( x_0 \), and becomes less accurate as one travels from this point. We can use this gradient to find the maximum of \( \hat{f} \) by setting \( \nabla \hat{f} \) to 0 and solving for \( x_{\text{max}} \):

\[ x_{\text{max}} = x_0 - (\nabla^2 f(x_0))^{-1} \nabla f(x_0) \]

Recall that the Hessian \( \nabla^2 f \) is a matrix, so to remove it from one side of the equation you must multiply both sides by the inverse matrix \( (\nabla^2 f)^{-1} \).