CISC 371 Class 5

Directional Derivative, Gradient, and Level Curves

Texts: [1], pp. 625–631

Main Concepts:

- *Partial derivative* $\partial/\partial w_j$
- *Directional derivative* $D_{\vec{u}}$
- *Gradient* $\underline{\nabla} f$
- *Jacobian matrix* $J_{\vec{f}}$

Sample Problem, Data Analytics: For any place in a meteorological map, How much does the temperature increase in a given direction?

In this course, we will use a limited amount of vector calculus. Our primary uses will be to:

- Develop a vector from a real-valued objective function that has a vector argument
- Develop a matrix from a real-valued vector function
- Understand how to select a search direction in a vector space
- Determine when to terminate an iterative search process

In elementary differential calculus, a function has a real-valued scalar argument and a real-valued result, or output. The amount that such a function varies, for a variation in the argument, is the first derivative of the function. This is defined as

$$\frac{df}{dt}(t) \stackrel{\text{def}}{=} \lim_{h \to 0} \frac{f(t+h) - f(t)}{h} \tag{5.1}$$

There are three natural extensions of Equation 5.1, to:

- A scalar function with a vector argument of size m, written as $f(\vec{w})$
- A vector function with a scalar argument, having m entries, written as $\vec{f}(t)$
- A vector function with a vector argument of size n, having m entries, written as $\vec{f}(\vec{w})$

We are primarily interested in the first two extensions because they arise often in numerical optimization.

In multivariable calculus, a function is expressed as having two or more real-valued arguments. A function of two variables, $f(tw1, w_2)$, does not seem to have a derivative as defined in Equation 5.1. Instead, such as function has two *partial* derivatives that are defined as

$$\begin{aligned} \frac{\partial f}{\partial w_1}(w_1, w_2) &\stackrel{\text{def}}{=} \lim_{h \to 0} \frac{f(w_1 + h, w_2) - f(w_1)}{h} \\ \frac{\partial f}{\partial w_2}(w_1, w_2) &\stackrel{\text{def}}{=} \lim_{h \to 0} \frac{f(w_1, w_2 + h) - f(w_1)}{h} \end{aligned} \tag{5.2}$$

In this course, we will use an objective function with a vector argument that we will write as $f(\vec{w})$. To extend Equation 5.2 to vectors, we can recall from linear algebra the idea of an *elementary vector*. The j^{th} entry of $\vec{e}_j$ is 1 and every other entry is 0. Using this idea, we can express the partial derivatives of Equation 5.2 as

$$\frac{\partial f}{\partial w_j}(\vec{w}) \stackrel{\text{def}}{=} \lim_{h \to 0} \frac{f(\vec{w} + h\,\vec{e}_j) - f(\vec{w})}{h} \tag{5.3}$$

Equation 5.3 is the same as the definition from multivariable calculus, using vectors instead of multiple scalar arguments. The usual rules for finding a partial derivative are the same, such as holding other variables as constant when performing the differentiation.

We can think of Equation 5.3 in two ways that will be useful later in this course:

- The partial derivative for coordinate j
- The derivative of $f(\vec{w})$ in the direction $\vec{e}_j$

For example, suppose that we want to numerically examine a map of North America than comes from a computation on effects of climate change. One question that we could ask is: for any location on the map, how much will the temperature rise or fall in a certain direction? We could answer this question by sampling in some direction by a fixed distance, and subtracting the temperatures. This would not be the "real" answer: what we should do is try to find the limit of the finite temperature difference as the fixed distance goes to zero. An example of a temperature-prediction map is shown in Figure 5.1.

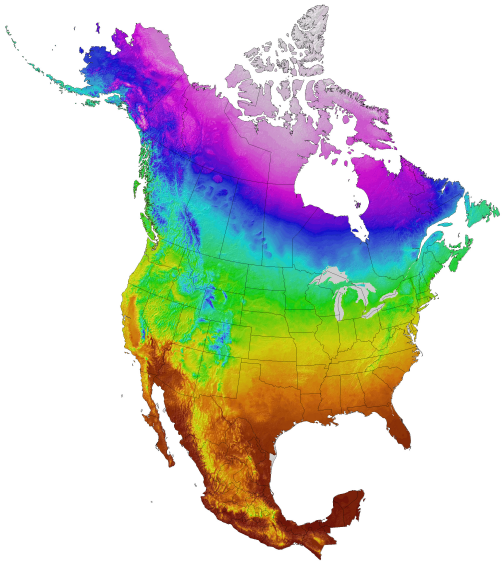

Figure 5.1: A "heat map" of estimated future temperatures in North America, where the coldest regions are colored as magenta and the hottest as deep red. A temperature is a scalar value at a point and the temperature may increase or decrease in any direction from an interior point in the continent. From Wang *et al.* 2016 [2].

5.1 The Directional Derivative

What if we use a vector other than an elementary vector? Suppose that we used a vector $\vec{v}$ in place of $\vec{e}_j$ in Equation 5.3. This would give us the derivative in the direction of the vector $\vec{v}$. The *directional derivative* has many notations; we will write it as

$$D_{\vec{v}} f(\vec{w}) \stackrel{\text{def}}{=} \lim_{h \to 0} \frac{f(\vec{w} + h\,\vec{v}) - f(\vec{w})}{h} \tag{5.4}$$

Using the Chain Rule, the directional derivative can be shown as equivalent to the sum of the products of each partial derivative with the corresponding entry of $\vec{v}$, so

$$\begin{aligned} D_{\vec{v}} f(\vec{w}) &= \frac{\partial f}{\partial w_1} v_1 + \frac{\partial f}{\partial w_2} v_2 + \cdots + \frac{\partial f}{\partial w_n} v_n \\ &= \sum_{j=1}^{n} \frac{\partial f}{\partial w_j} v_j \end{aligned} \tag{5.5}$$

We would like to use a more concise notation for the sum. Let us explore a few options that are available.

Because $\vec{v}$ is a vector, it is specifically a column vector with m rows. The structure of $\vec{v}$ constrains Equation 5.5 to be one of three forms:

- The dot product of a vector of partial derivatives with $\vec{v}$:

$$\begin{bmatrix} \dfrac{\partial f}{\partial w_1} \\ \dfrac{\partial f}{\partial w_2} \\ \vdots \\ \dfrac{\partial f}{\partial w_n} \end{bmatrix} \cdot \vec{v}$$

- The dot product of $\vec{v}$ with a vector of partial derivatives

$$\vec{v} \cdot \begin{bmatrix} \dfrac{\partial f}{\partial w_1} \\ \dfrac{\partial f}{\partial w_2} \\ \vdots \\ \dfrac{\partial f}{\partial w_n} \end{bmatrix}$$

- The product of a "row matrix" of partial derivatives with $\vec{v}$

$$\begin{bmatrix} \dfrac{\partial f}{\partial w_1} & \dfrac{\partial f}{\partial w_2} & \cdots & \dfrac{\partial f}{\partial w_n} \end{bmatrix} \vec{v}$$

We will choose the third notation, recognizing that it is not universally used in the literature on optimization. For extra clarity, we will always use an underscore to indicate a row matrix; for example, we will write the row vector with symbol "a" and m columns as

$$\underline{a} \stackrel{\text{def}}{=} \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \tag{5.6}$$

For this course, a mathematical object that looks like a row matrix will be called *one-form* or a *1-form*. We will not use the term "row matrix" so that we can avoid confusion in writing, but a student who thinks of a 1-form as a row-type object should be able to understand the following course material.

The idea of a 1-form leads us to a fundamental definition for numerical optimization.

Definition: gradient operator of $f(\vec{w})$

For $\vec{w} \in \mathbb{R}^n$ and any continuously differentiable $f : \mathbb{R}^n \to \mathbb{R}$, the *gradient operator* of a function $f(\vec{w})$ is defined as the 1-form

$$\underline{\nabla} f(\vec{w}) \stackrel{\text{def}}{=} \left[\frac{\partial f}{\partial w_1}(\vec{w}) \quad \frac{\partial f}{\partial w_2}(\vec{w}) \quad \cdots \quad \frac{\partial f}{\partial w_n}(\vec{w}) \right] \tag{5.7}$$

We will use the term *gradient* as a shorthand for "gradient operator applied to the function $f(\vec{w})$". We can now combine Equation 5.4 and Equation 5.5 into a single result:

$$D_{\vec{v}} f(\vec{w}) = [\underline{\nabla} f(\vec{w})]\vec{v} \tag{5.8}$$

Many texts in numerical optimization will treat the result of using the gradient operator as a *vector*. Treating the gradient as a vector is easy to represent and easy to manipulate, but is incorrect from the viewpoint of differential geometry. The vector representation and can cause difficulties in understanding concepts such as the contour density in a 2D plot of a scalar function that has a size-2 vector argument.

We will make an explicit assumption about the search space $\mathbb{V}$ in which we seek a minimizer: $\mathbb{V}$ *is a Euclidean space*. Not all search spaces are Euclidean; a familiar example is the representation of a point in a 2D plane using polar coordinates ϕ and ρ, where ϕ is the angle of a point from the X axis and ρ is the radius from the origin to the point. For a non-Euclidean search space, converting between a 1-form and a vector requires a Riemannian metric or another advanced geometrical object that is beyond the scope of this course.

For the important and usual condition, where the search space $\mathbb{V}$ is Euclidean, conversion between a 1-form and a vector can be done with the ordinary transpose operator. This operator produces the two equations[1] for conversion

$$\begin{array}{rcl} [\vec{a}]^T & = & \underline{a} \\ [\underline{a}]^T & = & \vec{a} \end{array} \tag{5.9}$$

When using either form of Equation 5.9, it is important to keep in mind the requirement that the search space must have a Euclidean structure.

[1] In a non-Euclidean geometry, Equation 5.9 would require using metric tensor g for the search space $\mathbb{V}$ and its inverse tensor.

5.2 The Jacobian Matrix

An immediate use of the gradient is to express the first derivative of a vector function that has a vector argument. We can write such a function as $\vec{f}(\vec{w})$, which expands as

$$\vec{f}(\vec{w}) \stackrel{\text{def}}{=} \begin{bmatrix} f_1(\vec{w}) \\ f_2(\vec{w}) \\ \vdots \\ f_m(\vec{w}) \end{bmatrix} \tag{5.10}$$

The derivative of $f_1(\vec{w})$ in Equation 5.10 is $\underline{\nabla} f_1(\vec{w})$. Written as a 1-form, the gradient is

$$\underline{\nabla} f_1(\vec{w}) = \begin{bmatrix} \dfrac{\partial f_1}{\partial w_1} & \dfrac{\partial f_1}{\partial w_2} & \cdots & \dfrac{\partial f_1}{\partial w_n} \end{bmatrix} \tag{5.11}$$

The derivative of $f_2(\vec{w})$ in Equation 5.10 is $\underline{\nabla} f_2(\vec{w})$, which is

$$\underline{\nabla} f_2(\vec{w}) = \begin{bmatrix} \dfrac{\partial f_2}{\partial w_1} & \dfrac{\partial f_2}{\partial w_2} & \cdots & \dfrac{\partial f_2}{\partial w_n} \end{bmatrix} \tag{5.12}$$

Likewise, the derivatives of $f_3(\vec{w})$ through $f_m(\vec{w})$ have the same structure, which is gradient 1-forms. Gathering these "rows", and omitting the argument $\vec{w}$ as an abbreviation, produces the *Jacobian* matrix

$$J_{\vec{f}}(\vec{w}) \stackrel{\text{def}}{=} \begin{bmatrix} \underline{\nabla} f_1 \\ \underline{\nabla} f_2 \\ \underline{\nabla} f_3 \\ \vdots \\ \underline{\nabla} f_m \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial w_1} & \dfrac{\partial f_1}{\partial w_2} & \dfrac{\partial f_1}{\partial w_3} & \cdots & \dfrac{\partial f_1}{\partial w_n} \\ \dfrac{\partial f_2}{\partial w_1} & \dfrac{\partial f_2}{\partial w_2} & \dfrac{\partial f_2}{\partial w_3} & \cdots & \dfrac{\partial f_2}{\partial w_n} \\ \dfrac{\partial f_3}{\partial w_1} & \dfrac{\partial f_3}{\partial w_2} & \dfrac{\partial f_3}{\partial w_3} & \cdots & \dfrac{\partial f_3}{\partial w_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial w_1} & \dfrac{\partial f_m}{\partial w_2} & \dfrac{\partial f_m}{\partial w_3} & \cdots & \dfrac{\partial f_m}{\partial w_n} \end{bmatrix} \quad \text{with} \quad J_{ij} \stackrel{\text{def}}{=} \frac{\partial f_i}{\partial w_j} \tag{5.13}$$

The Jacobian matrix of Equation 5.13, evaluated at an argument $\vec{w}_0$, is a matrix of $m \times n$ real numbers. It can be analyzed the way that any other matrix can be analyzed; for example, we can ask whether the Jacobian matrix at a point $\vec{w}_0$ has full rank.

The Jacobian also tersely represents the partial derivatives of the individual functions f_i with respect to variable w_j. By expansion, we can confirm that the product of a Jacobian matrix and any search-space vector $\vec{v}$ is

$$J_{\vec{f}}\vec{v} = \begin{bmatrix} D_{\vec{v}}f_1 \\ D_{\vec{v}}f_2 \\ D_{\vec{v}}f_3 \\ \vdots \\ D_{\vec{v}}f_m \end{bmatrix} \tag{5.14}$$

It is important to know that some authors define the Jacobian in the "transpose" sense. When reading work other sources, we must take care to determine the convention that is used.

5.3 Level Curves

A function that has a vector argument will evaluate to a single scalar value for each vector in the domain of the function. For most of the functions that we will use in this course, there are generally many – usually an infinitude – of domain vectors that map to the same scalar value.

If we refer to a scalar value as a *level*, then all of the domain vectors that map to a given level can be gathered together. These vectors are the *level curve* of the function at the given level.

Definition: level curve of $f(\vec{w})$ at l is $\mathbb{S}_C(f, l)$

For any $\vec{u} \in \mathbb{R}^n$, any $l \in \mathbb{R}$, and any $f : \mathbb{R}^n \to \mathbb{R}$, the *level curve* of f at l is defined as

$$\mathbb{S}_C(f, l) \stackrel{\text{def}}{=} \{\vec{u} : f(\vec{u}) = l\} \tag{5.15}$$

Another word for a level curve is a *contour*. This word is strongly associated with 2D representations of the local 3D topography of our planet's surface. We can use standard plotting software, such as `MATLAB`, to draw the level curves of a function.

$$f_1(\vec{w}) = \vec{m}_1^T\vec{w} + c_1 \quad \text{where} \quad \vec{m}_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \text{and} \quad c_1 = -2 \tag{5.16}$$

$$f_2(\vec{w}) = [\vec{w} - \vec{g}_2]^T[\vec{w} - \vec{g}_2] \quad \text{where} \quad \vec{g}_2 = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \tag{5.17}$$

For example, Figure 5.2 shows the surfaces and contours of a linear function and a quadratic function. We will often find it useful to have both representations when we try to understand the geometry of optimization.

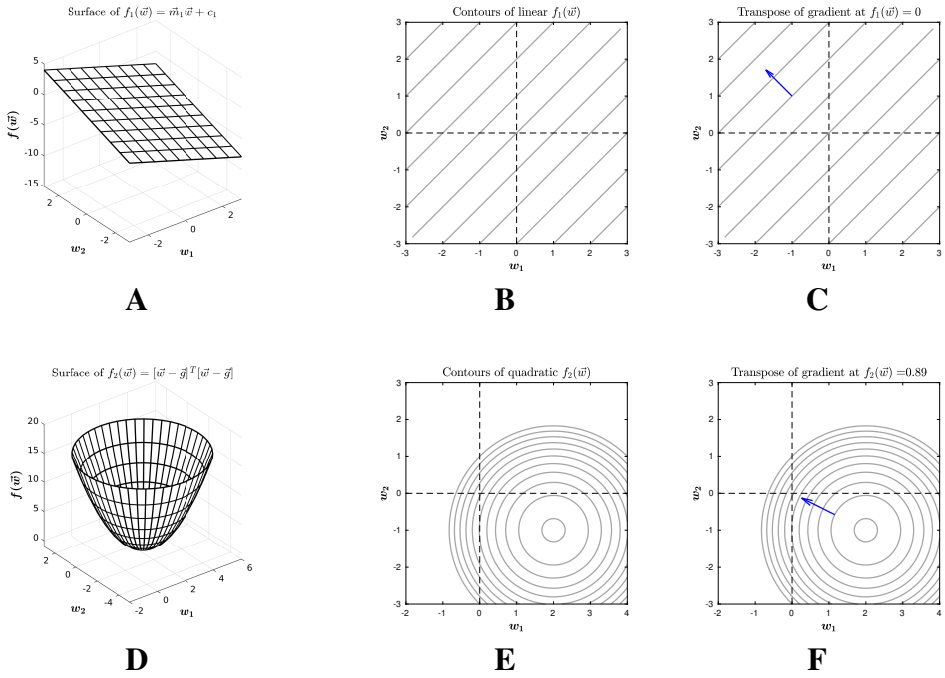

Figure 5.2: Surface plots and level curves of convex functions that have a 2D vector argument. (A) A surface plot of function $f_1(\vec{w})$ defined in Equation 5.16. (B) A contour plot of $f_1(\vec{w})$. (C) The contours of $f_1(\vec{w})$ with the transpose of a gradient, shown in blue. (D) A surface plot of function $f_3(\vec{w})$ defined in Equation 5.17. (E) A contour plot of $f_3(\vec{w})$. (F) The contours of $f_2(\vec{w})$ with the transpose of a gradient, shown in red.

Figure 5.2 also show the transpose of a gradient that is superimposed on the contour plot, where the arrow represents a unit vector. One way to interpret the gradient 1-form is that it describes the instantaneous number of contours that are "crossed" in a unit step in the direction of its transpose. Function $f_1(\vec{w})$ has a constant gradient at every point $\vec{w}$ of $[-1\ 1]$ so, in direction $[-1\ 1]^T$, the gradient will "cross" $\sqrt{2}$ unit-spaced levels. Function $f_2(\vec{w})$, evaluated at the point shown in the figure, will "cross" approximately 1.89 unit-spaced levels.

The physical interpretation of the gradient may be clearer if we refer to the motivating example of Figure 5.1. A point in the map is measured, from some origin, in meters (m). At some point, in some direction, the gradient measures the change of temperature in °K for a unit step. The units of the gradient, which are °K/m, are not the units of temperature and are not the units of distance – the gradient is certainly not a vector in the underlying space, which in this example is the map.

A level curve, as specified in Definition 5.15, can be found for a nonconvex function. Another term for a level curve of a function is a *contour*. A set of contours is a contour plot, which is a

useful way of understanding a complicated function. An example of a MATLAB builtin function is show in Figure 5.3, where we graph a surface mesh and a set of level curves. This function is complicated enough that it is not easy to determine where the maxima, minima, and saddle points are from the surface mesh. Careful study of the contour plot can reveal the locations of some of the stationary points for this function.

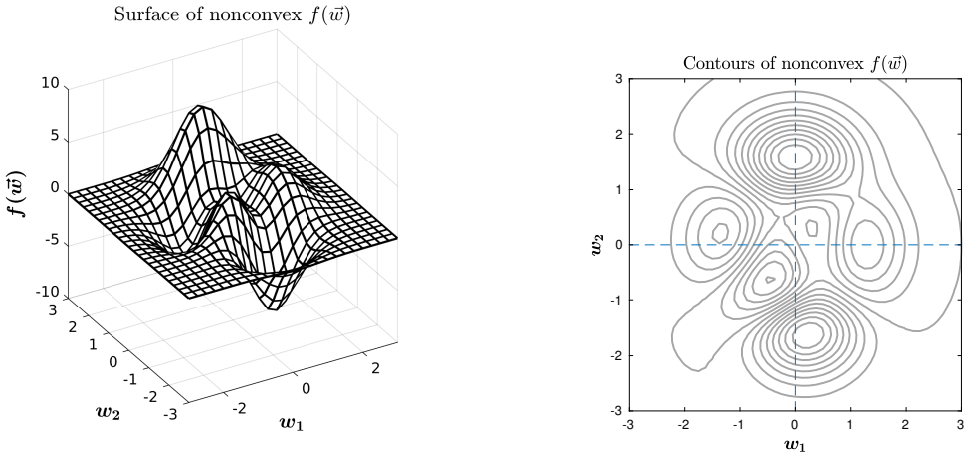

Figure 5.3: Surface plot and level curves of a nonconvex function. (A) A surface plot of the function suggests there are three local maxima and two local minima. (B) The level curves suggest that there are at least three saddle points in the plotted region, located near concavities of contours.

Extra Notes

5.4 Extra Notes on Gradients of Linear and Quadratic Functions

There are two kinds of function that we will use often in this course: a linear function of a vector, and a quadratic form of a vector. Before we derive the gradients of such functions, it is useful to recall a basic result in linear algebra.

Theorem: Symmetry of transpose product

For any $\vec{u} \in \mathbb{R}^n$ *and for any* $\vec{v} \in \mathbb{R}^n$,

$$\vec{u}^T \vec{v} = \vec{v}^T \vec{u} \tag{5.18}$$

Proof:

$$\vec{u}^T\vec{v} = \vec{u}\cdot\vec{v} = \sum_{i=1}^{n} u_i v_i = \sum_{i=1}^{n} v_i u_i = \vec{v}\cdot\vec{u} = \vec{v}^T\vec{u}$$

We can use Theorem 5.18 to prove a related result for any square matrix that is symmetric.

Theorem: Symmetry of transpose product

For any $\vec{u} \in \mathbb{R}^n$, for any $\vec{v} \in \mathbb{R}^n$, and for any symmetric matrix $B_{n\times n}$,

$$\vec{u}^T B\vec{v} = \vec{v}^T B\vec{u} \tag{5.19}$$

Proof:

$$\vec{u}^T B\vec{v} = [\vec{u}^T B^T]\vec{v} = [B\vec{u}]^T\vec{v} = \vec{v}^T B\vec{u}$$

We can use the definition of the directional derivative in Equation 5.4, and its equivalent form in Equation 5.8, to find the gradients of two kinds of functions that we will use often.

Theorem: Gradient of a linear function

For any $\vec{w} \in \mathbb{R}^n$ and for any vector $\vec{a} \in \mathbb{R}^n$, the gradient of the function

$$f(\vec{w}) = \vec{a}^T\vec{w}$$

is

$$\underline{\nabla} f(\vec{w}) = \vec{a}^T \tag{5.20}$$

Proof:

$$\begin{aligned}
D_{\vec{v}} f(\vec{w}) &= \lim_{h\to 0} \frac{\vec{a}^T[\vec{w}+h\vec{v}] - \vec{a}^T\vec{w}}{h} \\
&= \lim_{h\to 0} \frac{\vec{a}^T\vec{w} + \vec{a}^T[h\vec{v}] - \vec{a}^T\vec{w}}{h} \\
&= \lim_{h\to 0} \frac{\vec{a}^T[h\vec{v}]}{h} \\
&= \lim_{h\to 0} \frac{h\left(\vec{a}^T\vec{v}\right)}{h} \\
&= \vec{a}^T\vec{v} \\
\Rightarrow \quad \underline{\nabla} f(\vec{w}) &= \vec{a}^T
\end{aligned}$$

Theorem: Gradient of a quadratic function

For any $\vec{w} \in \mathbb{R}^n$ and for any symmetric matrix $B_{n\times n}$, the gradient of the function

$$f(\vec{w}) = \vec{w}^T B \vec{w}$$

is

$$\underline{\nabla} f(\vec{w}) = 2\vec{w}^T B \tag{5.21}$$

Proof:

$$\begin{aligned}
D_{\vec{v}} f(\vec{w}) &= \lim_{h\to 0} \frac{[\vec{w}^T + h\vec{v}]B[\vec{w} + h\vec{v}] - \vec{w}^T B\vec{w}}{h} \\
&= \lim_{h\to 0} \frac{\vec{w}^T B\vec{w} + \vec{w}^T B[h\vec{v}] + [h\vec{v}^T]B\vec{w} + [h\vec{v}^T]B[h\vec{v}] - \vec{w}^T B\vec{w}}{h} \\
&= \lim_{h\to 0} \frac{\vec{w}^T B[h\vec{v}] + [h\vec{v}^T]B\vec{w} + [h\vec{v}^T]B[h\vec{v}]}{h} \\
&= \lim_{h\to 0} \frac{h\left(\vec{w}^T B\vec{v} + \vec{v}^T B\vec{w}\right) + h^2(\vec{v}^T B\vec{v})}{h} \\
&= \vec{w}^T B\vec{v} + \vec{v}^T B\vec{w} \\
&= \vec{w}^T B\vec{v} + \vec{w}^T B\vec{v} \\
&= [2\vec{w}^T B]\vec{v} \\
\Rightarrow \quad \underline{\nabla} f(\vec{w}) &= 2\vec{w}^T B
\end{aligned}$$

References

[1] Nocedal J, Wright S: Numerical Optimization. Springer Science & Business Media, 2006

[2] Wang T, Hamann A, Spittlehouse D, Carroll C: Locally downscaled and spatially customizable climate data for historical and future periods for North America. *PloS one* 11(6):e0156720, 2016