Hyperplane Geometry in Machine Learning

Keith M. Chugg, B. Keith Jenkins
vers. 0.10

February 22, 2023

Contents

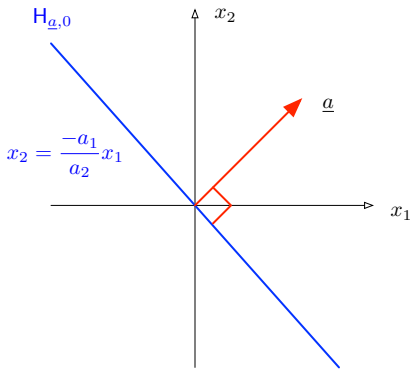


Figure 1: The hyperplane orthogonal to $\underline{a}$, passing through the origin, for $M = 2$.

Hyperplanes arise in the interpretation of many machine learning (ML) algorithms and problems, especially those with linear models. This document develops several concepts that are useful throughout the class and, more generally, in the study of ML.

1 Hyperplane Passing Through the Origin

A hyperplane passing through the origin is defined as

$$\boxed{\mathsf{H}_{\underline{a},0} = \{\underline{x} : \underline{a}^{\mathrm{t}}\underline{x} = 0\} \quad \underline{a} \neq \underline{0}} \tag{1}$$

where the variable vector $\underline{x}$ and the coefficient vector $\underline{a}$ are both in $\mathbb{R}^M$. When there is little room for confusion, we will simplify this notation to H_0. The hyperplane represents a linear constraint on the variables $\{x_i\}_{i=1}^M$ and therefore $\mathsf{H}_{\underline{a},0}$ is a $(M-1)$ dimensional linear subspace of $\mathbb{R}^M$. For $M = 1$, $M = 2$, and $M = 3$, this is a point, line, and plane, respectively. For $M > 3$, this is a hyper-dimensional extension of a plane, hence the name hyperplane.

For $M = 2$ the equation for the line is

$$\underline{x} \in \mathsf{H}_{\underline{a},0} \subset \mathbb{R}^2 \quad \Longleftrightarrow \quad x_2 = \frac{-a_1}{a_2}x_1 \tag{2}$$

where it has been assumed that $a_2 \neq 0$. This is the equation of a line in the (x_1, x_2) plane with slope $-a_1/a_2$. This example is shown in Fig. 1.

Note that the line in Fig. 1 is orthogonal to the vector $\underline{a}$. This is true in general. In fact, $\underline{a}$ forms a basis for the orthogonal complement of $\mathsf{H}_{\underline{a},0}$, which is a one-dimensional linear subspace of $\mathbb{R}^M$. In other words, any vector that is orthogonal to all vectors in $\mathsf{H}_{\underline{a},0}$ is a scalar multiple of $\underline{a}$. The example from Fig. 1 is extended to $M = 3$ in Fig. 2, where we've tried out best to illustrate the plane $\mathsf{H}_{\underline{a},0}$ passing through the origin with $\underline{a}$ orthogonal to the plane. Any vector $\underline{x} \in \mathsf{H}_{\underline{a},0}$ is orthogonal to $\underline{a}$ by definition so $\underline{a}$ is orthogonal to the hyperplane. Another way to see this is that $\mathsf{H}_{\underline{a},0}$ is a level curve for the function $f(\underline{x}) = \underline{a}^{\mathrm{t}}\underline{x}$, $\nabla_{\underline{x}} f(\underline{x}) = \underline{a}$, and the gradient of a function is orthogonal to all if its level curves. For this reason, we can use the following terminology

$$\boxed{\mathsf{H}_{\underline{a},0} = \text{the hyperplane normal to } \underline{a} \text{ passing through the origin}}$$

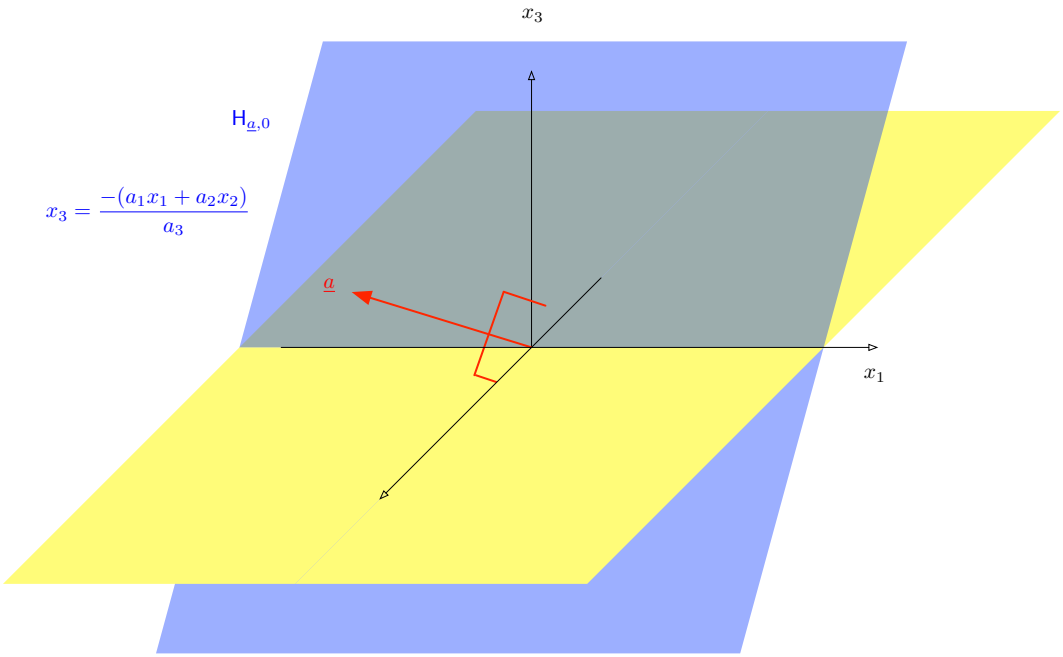


Figure 2: The hyperplane orthogonal to $\underline{a}$, passing through the origin, for $M = 3$.

The fact that $\mathsf{H}_{\underline{a},0}$ is a $M-1$ dimensional subset of $\mathbb{R}^M$ and span$(\underline{a})$ is the one-dimensional orthogonal complement motivates an orthogonal decomposition of an arbitrary vector $\underline{x} \in \mathbb{R}^M$ of the form

$$\underline{x} = \underline{x}_{H_0} + \underline{x}_{H_0^\perp} \tag{3a}$$
$$= \underline{x}_{H_0} + \left(\underline{x}^{\mathrm{t}}\underline{u}_a\right)\underline{u}_a \tag{3b}$$
$$= \underline{x}_{H_0} + R_{H^\perp_{\underline{a},0}}(\underline{x})\underline{u}_a \tag{3c}$$

where $\underline{u}_a$ is unit vector in the direction of $\underline{a}$ and $R_{H^\perp_{\underline{a},0}}(\underline{x})$ is the *projection coefficient of* $\underline{x}$ *in the direction of* $\underline{a}$:

$$\underline{u}_a = \frac{\underline{a}}{\|\underline{a}\|} \tag{4}$$

$$R_{H^\perp_{\underline{a},0}}(\underline{x}) \stackrel{\Delta}{=} \underline{x}^{\mathrm{t}}\underline{u}_a = \frac{\underline{a}^{\mathrm{t}}\underline{x}}{\|\underline{a}\|} \tag{5}$$

The term $\underline{x}_{H_0}$ in (3) is the projection of $\underline{x}$ onto the subspace $\mathsf{H}_{\underline{a},0}$ and $\underline{x}_{H_0^\perp}$ is the projection onto the one-dimensional space with basis $\underline{u}_a$. Thus, $\underline{x}_{H_0}$ is the closest point in $\mathsf{H}_{\underline{a},0}$ to $\underline{x}$ – *i.e.,* it is the minimizing vector for the problem: $\min_{\underline{v}\in\mathsf{H}_{\underline{a},0}} \|\underline{x} - \underline{v}\|$. Similarly, $\underline{x}_{H_0^\perp}$ is the closest point in $\mathsf{H}^\perp_{\underline{a},0}$ to $\underline{x}$ – *i.e.,* the minimizing value of $\underline{v}$ of $\min_{\underline{v}\in\mathsf{H}^\perp_{\underline{a},0}} \|\underline{x} - \underline{v}\|$.

The $M = 2$ example from Fig. 1 is updated to show this decomposition in Fig. 3. Note that $R_{H^\perp_{\underline{a},0}}(\underline{x}) = \|\underline{x}\| \cos\theta_{x,a}$ where $\theta_{x,a}$ is the angle between $\underline{x}$ and $\underline{a}$ (or, equivalently $\underline{u}_a$). This means that $R_{H^\perp_{\underline{a},0}}(\underline{x}) > 0$ for $\theta_{x,a} \in (-\pi/2, +\pi/2)$ and $R_{H^\perp_{\underline{a},0}}(\underline{x}) < 0$ for $\theta_{x,a} \in (\pi/2, 3\pi/2)$. Stated

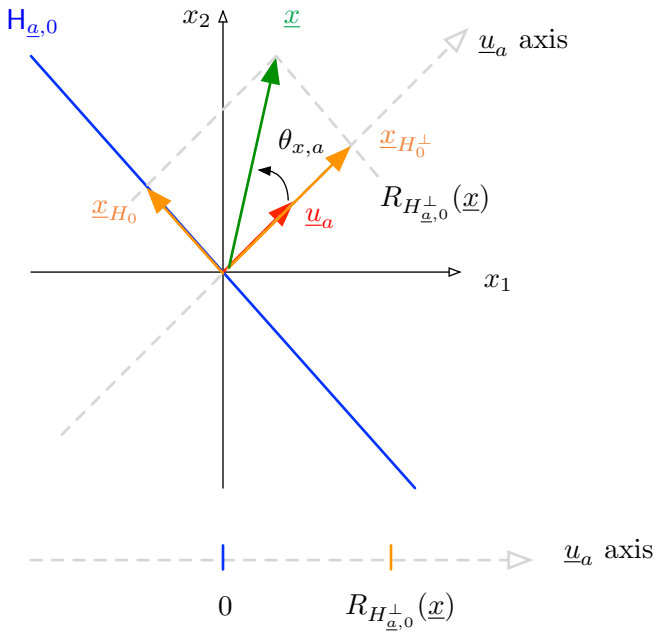


Figure 3: The hyperplane $\mathsf{H}_{\underline{a},0}$ and a vector $\underline{x}$ in $M = 2$. The orthogonal decomposition of $\underline{x}$ into $\underline{x}_{H_0} + \underline{x}_{H_0^\perp}$ is shown. Also shown below is the view on the 1-dimensional space with basis $\underline{u}_a$.

differently, $R_{H^\perp_{\underline{a},0}}(\underline{x})$ is positive when $\underline{x}$ points in the same general direction as $\underline{a}$ and negative when it points in generally the opposite direction as $\underline{a}$.

Finally, note that $\mathsf{H}_{\underline{a},0} = \mathsf{H}_{-\underline{a},0}$ since if $\underline{x}$ is orthogonal to $\underline{a}$ it is also orthogonal to $-\underline{a}$. However,

$$R_{H^\perp_{\underline{a},0}}(\underline{x}) = -R_{H^\perp_{-\underline{a},0}}(\underline{x}) \tag{6}$$

since, referring to Fig. 3, the direction of the $\underline{u}_{-a}$ axis is antipodal to that of the $\underline{u}_a$ axis. This concept is illustrated in Fig. 4 in which the example of Fig. 3 is continued. The quantity $R_{H^\perp_{\underline{a},0}}(\underline{x})$ is sometimes referred to as an algebraic distance or a signed distance. This is because $R_{H^\perp_{\underline{a},0}}(\underline{x})$ is the algebraic difference between the projection coefficient of $\underline{x}$ onto $\underline{a}$ and the projection of the zero vector onto $\underline{a}$. This difference is positive when $\theta_{x,a} \in (-\pi/2, +\pi/2)$ and negative for $\theta_{x,a} \in (\pi/2, 3\pi/2)$. This is a somewhat trivial observation amounting to $R_{H^\perp_{\underline{a},0}}(\underline{x}) - 0 = R_{H^\perp_{\underline{a},0}}(\underline{x})$, but it will become more apparent in the next section why we point out this interpretation at this point in the development. In any event, $R_{H^\perp_{\underline{a},0}}(\underline{x})$ is more accurately referred to as a difference than a distance since a mathematical measure of distance is strictly non-negative.

2 Hyperplane Offset From the Origin

A Hyperplane offset, or shifted, from the origin is defined by

$$\boxed{\mathsf{H}_{\underline{a},c} = \{\underline{x} : \underline{a}^{\mathrm{t}}\underline{x} = c\} = \{\underline{x} : \underline{a}^{\mathrm{t}}\underline{x} - c = 0\} \quad \underline{a} \neq \underline{0}} \tag{7}$$

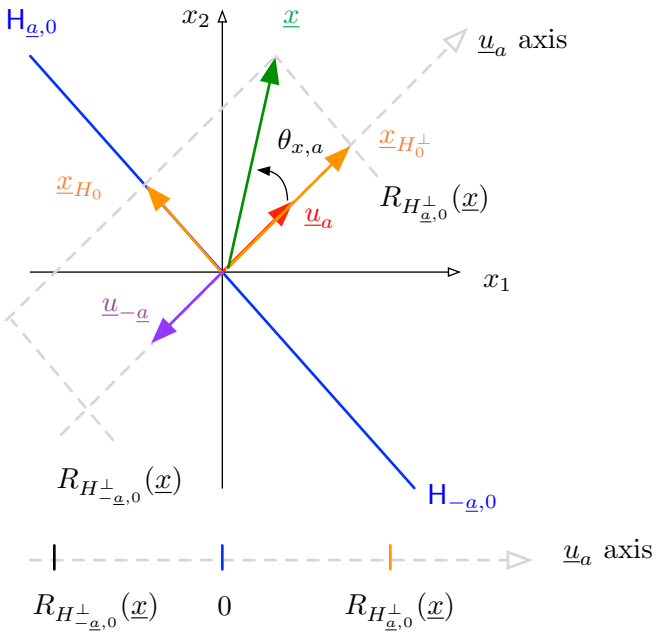


Figure 4: The same diagram as shown in Fig. 3, but with the unit vector in the direction of $-\underline{a}$ shown along with the projection coefficient of $\underline{x}$ along that direction, $R_{H^\perp_{-\underline{a},0}}(\underline{x})$.

where, again, $\underline{x}$ and $\underline{a}$ are the variable and coefficient vectors, respectively, and, now, c is a real constant. If $c = 0$, this is a hyperplane through the origin, but for $c \neq 0$, this is a hyperplane that does not pass through the origin – i.e., $\underline{a}^{\mathrm{t}}\underline{0} + c \neq 0$. Note that $\mathsf{H}_{\underline{a},c}$ is a subspace of $\mathbb{R}^M$ for all choices of c, but it is a linear subspace only for $c = 0$ since $\mathsf{H}_{\underline{a},c}$ does not contain the origin for $c \neq 0$. For $c \neq 0$, $\mathsf{H}_{\underline{a},c}$ *is an affine transformation of the linear space* $\mathsf{H}_{\underline{a},0}$. For $M = 2$, this is a line, governed by

$$\underline{x} \in \mathsf{H}_{\underline{a},c} \subset \mathbb{R}^2 \quad \Longleftrightarrow \quad x_2 = \frac{-a_1}{a_2}x_1 + \frac{c}{a_2} \tag{8}$$

This is illustrated in Fig. 5, which extends the example of Fig. 3, where two values of c are considered. It is worth noting that in this example, since $\underline{a}$ is in the first quadrant, a_1, a_2 are both positive. Since the intercept on the x_2-axis is positive, it can be deduced that $c_2 > c_1 > 0$ in the example of Fig. 5. Thus, a positive value of c offsets $\mathsf{H}_{\underline{a},0}$ in the the direction of $\underline{a}$ and a negative value of c offsets $\mathsf{H}_{\underline{a},0}$ in the antipodal direction of $\underline{a}$. Two other examples of planes offset from the origin are shown in Fig. 6 for $M = 3$.

In fact, every vector in $\mathsf{H}_{\underline{a},c}$ is vector in $\mathsf{H}_{\underline{a},0}$ plus a scalar multiple of $\underline{a}$ that depends on c. More precisely,

$$\underline{x} \in \mathsf{H}_{\underline{a},c} \quad \Longleftrightarrow \quad \underline{x} = \underline{x}_{H_0} + \frac{c}{\|\underline{a}\|}\underline{u}_a \tag{9}$$

Notice that $\underline{x}_{H_0} \in \mathsf{H}_{\underline{a},0}$ and it is the projection of $\underline{x}$ onto $\mathsf{H}_{\underline{a},0}$. The vector $\frac{c}{\|\underline{a}\|}\underline{u}_a$ is orthogonal to $\mathsf{H}_{\underline{a},0}$, so it has zero projection coefficient on $\mathsf{H}_{\underline{a},0}$. This concept is illustrated in Fig. 7. Also, we can

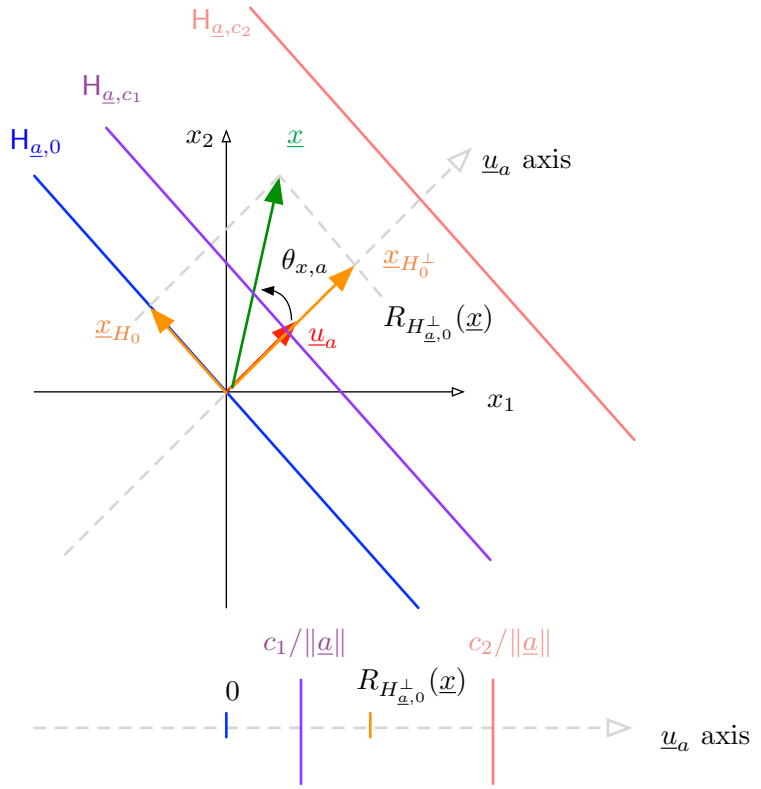
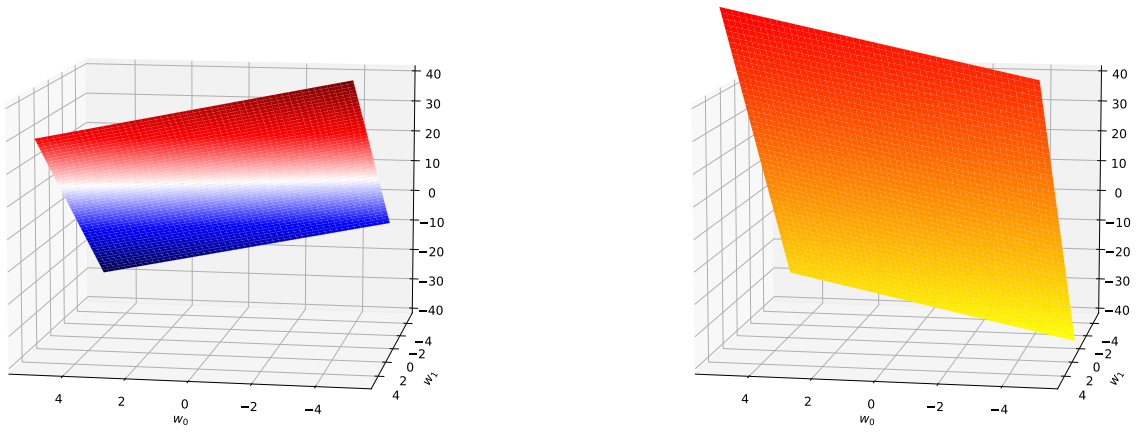


Figure 5: Three hyperplanes, each orthogonal to $\underline{a}$, are shown: $\mathsf{H}_{\underline{a},0}$ which passes through the origin and $\mathsf{H}_{\underline{a},c_1}$ and $\mathsf{H}_{\underline{a},c_2}$ which are offset from the origin along the direction of $\underline{a}$ by $c_1/\|\underline{a}\|$ and $c_2/\|\underline{a}\|$, respectively. In this example c_1 and c_2 are positive. Note that $\underline{x}$ lies on the non-origin side of $\mathsf{H}_{\underline{a},c_1}$ and the origin side of $\mathsf{H}_{\underline{a},c_2}$. Also shown is the projection onto the one-dimensional subspace with basis $\underline{u}_a$.



(a) $\underline{a} = [-2\ 6]^{\mathrm{t}}$ and $c = 10$ (b) $\underline{a} = [2\ 10]^{\mathrm{t}}$ and $c = -5$

Figure 6: Examples of offset hyperplanes with $M = 3$.

confirm algebraically that $\underline{x}$ in (9) is in $\mathsf{H}_{\underline{a},c}$ via

$$\underline{a}^{\mathrm{t}}\left(\underline{x}_{H_0} + \frac{c}{\|\underline{a}\|}\underline{u}_a\right) = \|\underline{a}\|\underline{u}_a^{\mathrm{t}}\left(\underline{x}_{H_0} + \frac{c}{\|\underline{a}\|}\underline{u}_a\right) \tag{10a}$$

$$= 0 + c\|\underline{u}_a\|^2 \tag{10b}$$

$$= c \tag{10c}$$

This motivates the definition of an *offset vector* $\underline{o}_{a,c}$ that offsets the origin to $\mathsf{H}_{\underline{a},c}$

$$\underline{o}_{a,c} = \frac{c}{\|\underline{a}\|}\underline{u}_a \tag{11}$$

and the notation that

$$\mathsf{H}_{\underline{a},c} = \mathsf{H}_{\underline{a},0} + \frac{c}{\|\underline{a}\|}\underline{u}_a \tag{12}$$

which is shorthand for (9). For example, in Fig. 7, since $\underline{x} \in \mathsf{H}_{\underline{a},c}$, $\underline{o}_{a,c} = \underline{x}_{H_0^{\perp}}$. More generally, $\underline{o}_{a,c}$ is the vector in $\mathsf{H}_{\underline{a},c}$ that is closest to the origin. Since $\underline{a}$ is orthogonal to $\mathsf{H}_{\underline{a},c}$, $\underline{o}_{a,c}$ is a scalar multiple of $\underline{a}$. The offset vectors $\underline{o}_{a,c_1}$ and $\underline{o}_{a,c_2}$ are not shown in Fig. 5 to avoid clutter, but they are vectors from the origin along the $\underline{u}_a$ direction to the purple and salmon planes, respectively. The projection coefficients of $\underline{o}_{a,c_1}$ and $\underline{o}_{a,c_2}$ are shown on the one-dimensional diagram, along the $\underline{u}_a$ direction, shown below the two-dimensional diagram. Given this observation, it is reasonable to refer to $\mathsf{H}_{\underline{a},c}$ as

$\mathsf{H}_{\underline{a},c}$ = the hyperplane normal to $\underline{a}$, offset from the origin by $c/\|\underline{a}\|$ in the direction of $\underline{a}$

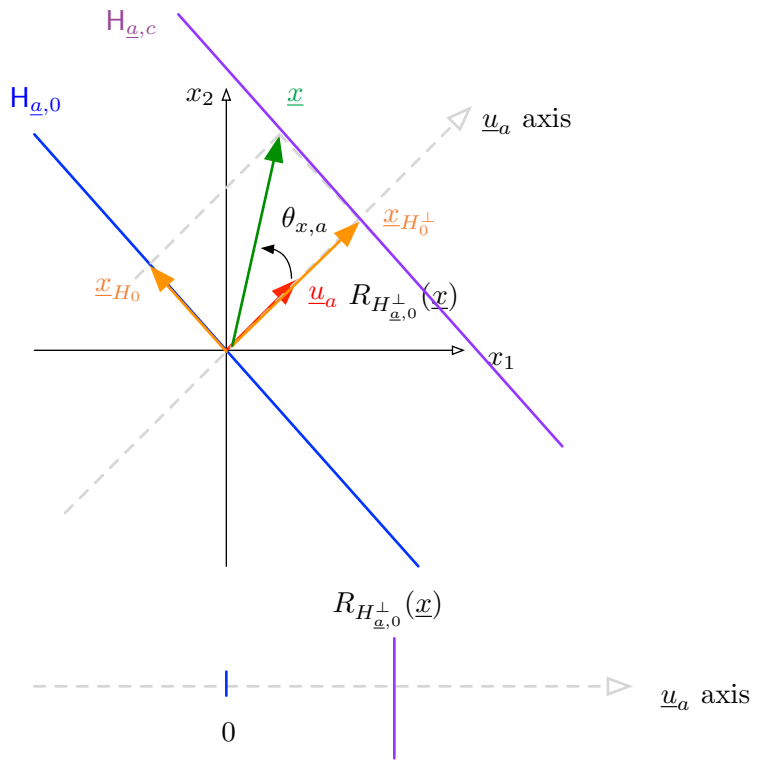


Figure 7: The geometry for the example of $\underline{x}$ in the hyperplane orthogonal to $\underline{a}$ offset in the direction of $\underline{a}$ by $c/\|\underline{a}\| = R_{H^\perp_{\underline{a},0}}(\underline{x})$.

2.1 Distance Measure for Vectors and Hyperplanes

The offset vector $\underline{o}_{a,c}$ is the vector in $\mathsf{H}_{\underline{a},c}$ that is closest to the origin in the Euclidean sense. It follows that the norm of this vector represents the distance between $\mathsf{H}_{\underline{a},c}$ and $\underline{0}$

$$\text{Distance between } \mathsf{H}_{\underline{a},c} \text{ and } \underline{0} = \|\underline{o}_{a,c}\| = \frac{|c|}{\|\underline{a}\|} \tag{13}$$

This is illustrated in Fig. 5 for two positive values of c. The case of $c < 0$ is illustrated in Fig. 8. In both Figs. 5 and 8, the vector $\underline{x}$ lies between the planes $\mathsf{H}_{\underline{a},c_1}$ and $\mathsf{H}_{\underline{a},c_2}$. In both cases, the distance from the origin to $\mathsf{H}_{\underline{a},c_1}$ and $\mathsf{H}_{\underline{a},c_2}$ are $|c_1|/\|\underline{a}\|$ and $|c_2|/\|\underline{a}\|$, respectively. Also, in both cases the distance between the vector $\underline{x}$ (or, equivalently, $\underline{x}_{\mathsf{H}_0^\perp}$) and $\mathsf{H}_{\underline{a},c_1}$ and $\mathsf{H}_{\underline{a},c_2}$ are $|R_{H_{\underline{a},0}^\perp}(\underline{x}) - c_1/\|\underline{a}\||$ and $|R_{H_{\underline{a},0}^\perp}(\underline{x}) - c_2/\|\underline{a}\||$, respectively. More generally,

$$\begin{aligned}
\delta(\underline{x}, \mathsf{H}_{\underline{a},c}) &= \delta(\mathsf{H}_{\underline{a},c}, \underline{x}) && \text{(14a)}\\
&= \text{Distance between } \mathsf{H}_{\underline{a},c} \text{ and } \underline{x} && \text{(14b)}\\
&\stackrel{\Delta}{=} \min_{\underline{v} \in \mathsf{H}_{\underline{a},c}} \|\underline{x} - \underline{v}\| && \text{(14c)}\\
&= \left| R_{H_{\underline{a},0}^\perp}(\underline{x}) - \frac{c}{\|\underline{a}\|} \right| && \text{(14d)}\\
&= \frac{|\underline{a}^{\mathrm{t}}\underline{x} - c|}{\|\underline{a}\|} && \text{(14e)}
\end{aligned}$$

This distance generalizes to a distance between two hyperplanes via the following definition

$$\begin{aligned}
\delta(\mathsf{H}_{\underline{a},c_1}, \mathsf{H}_{\underline{a},c_2}) &= \delta(\mathsf{H}_{\underline{a},c_2}, \mathsf{H}_{\underline{a},c_1}) && \text{(15a)}\\
&= \text{Distance between } \mathsf{H}_{\underline{a},c_1} \text{ and } \mathsf{H}_{\underline{a},c_2} && \text{(15b)}\\
&\stackrel{\Delta}{=} \min_{\underline{v}_1 \in \mathsf{H}_{\underline{a},c_1}, \underline{v}_2 \in \mathsf{H}_{\underline{a},c_2}} \|\underline{v}_1 - \underline{v}_2\| && \text{(15c)}\\
&= \delta(\underline{o}_{a,c_1}, \mathsf{H}_{\underline{a},c_2}) = \delta(\underline{o}_{a,c_2}, \mathsf{H}_{\underline{a},c_1}) && \text{(15d)}\\
&= \|\underline{o}_{a,c_1} - \underline{o}_{a,c_2}\| && \text{(15e)}\\
&= \frac{|c_1 - c_2|}{\|\underline{a}\|} && \text{(15f)}
\end{aligned}$$

2.2 Difference Measure for Vectors and Hyperplanes

The function $\delta(\cdot)$ defined above is a distance in the mathematical sense. Specifically, it is symmetric in its arguments, it is always non-negative, and it obeys the triangle inequality. However, referring to Figs. 5 and 8, the distance measure is not informative regarding the question of which side of the hyperplane $\underline{x}$ lies. To address this, we introduce the *(signed) difference function*

$$\begin{aligned}
d_{\underline{a}}(\mathsf{H}_{\underline{a},c}, \underline{x}) &= \text{Projection coefficient of } \underline{x} \text{ on } \underline{a} \;-\; \text{Projection coefficient of } \underline{o}_{a,c} \text{ on } \underline{a} && \text{(16a)}\\
&= R_{H_{\underline{a},0}^\perp}(\underline{x}) - \frac{c}{\|\underline{a}\|} && \text{(16b)}\\
&= \frac{\underline{a}^{\mathrm{t}}\underline{x} - c}{\|\underline{a}\|} && \text{(16c)}
\end{aligned}$$

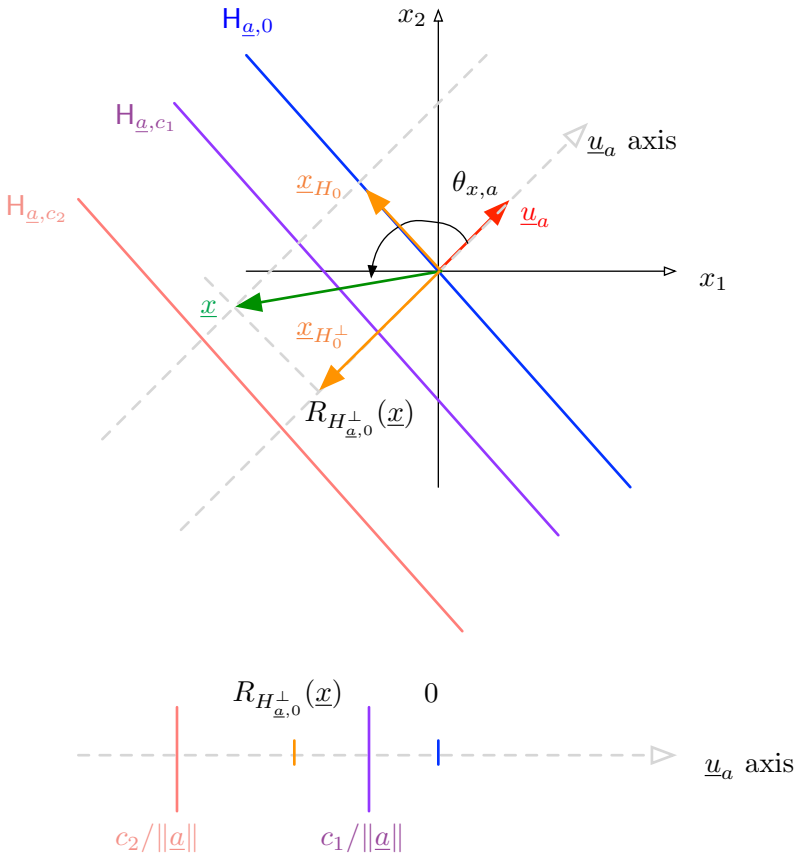


Figure 8: Three hyperplanes, each orthogonal to $\underline{a}$, are shown: $\mathsf{H}_{\underline{a},0}$ which passes through the origin and $\mathsf{H}_{\underline{a},c_1}$ and $\mathsf{H}_{\underline{a},c_2}$ which are offset from the origin along the direction of $\underline{a}$ by $c_1/\|\underline{a}\|$ and $c_2/\|\underline{a}\|$, respectively. In this example c_1 and c_2 are negative. Note that $\underline{x}$ lies on the non-origin side of $\mathsf{H}_{\underline{a},c_1}$ and the origin side of $\mathsf{H}_{\underline{a},c_2}$. Also shown is the projection onto the one-dimensional subspace with basis $\underline{u}_a$.

In the example of Fig. 5, $d_{\underline{a}}(\underline{x}, \mathsf{H}_{\underline{a},c_1}) > 0$ and $d_{\underline{a}}(\underline{x}, \mathsf{H}_{\underline{a},c_2}) < 0$. In the example of Fig. 8, the same holds, which illustrates:[1]

$$d_{\underline{a}}(\mathsf{H}_{\underline{a},c}, \underline{x}) > 0 \iff \underline{x} \text{ is on the non-origin-side of } \mathsf{H}_{\underline{a},c} \tag{17a}$$
$$d_{\underline{a}}(\mathsf{H}_{\underline{a},c}, \underline{x}) < 0 \iff \underline{x} \text{ is on the origin-side of } \mathsf{H}_{\underline{a},c} \tag{17b}$$

and if this difference is zero, $\underline{x}$ is in $\mathsf{H}_{\underline{a},c}$.

This signed difference can have subtle properties such as:

$$d_{\underline{a}}(\underline{x}, \mathsf{H}_{\underline{a},c}) \stackrel{\Delta}{=} \text{Projection coefficient of } \underline{o}_{a,c} \text{ on } \underline{a} - \text{Projection coefficient of } \underline{x} \text{ on } \underline{a} \tag{18a}$$
$$= -d_{\underline{a}}(\mathsf{H}_{\underline{a},c}, \underline{x}) \tag{18b}$$

This emphasizes the directional nature of $d_{\underline{a}}(\cdot, \cdot)$. Similarly,

$$d_{-\underline{a}}(\mathsf{H}_{\underline{a},c}, \underline{x}) = \text{Projection coefficient of } \underline{x} \text{ on } (-\underline{a}) \; - \; \text{Projection coefficient of } \underline{o}_{a,c} \text{ on } (-\underline{a}) \tag{19a}$$
$$= -d_{\underline{a}}(\mathsf{H}_{\underline{a},c}, \underline{x}) \tag{19b}$$

In many cases, where this interpretation is used, the vector $\underline{a}$ is fixed. This suggests that the following shorthand notation

$$\boxed{d(\mathsf{H}_{\underline{a},c} \to \underline{x}) \stackrel{\Delta}{=} d_{\underline{a}}(\mathsf{H}_{\underline{a},c}, \underline{x}) = \frac{\underline{a}^{\mathrm{t}}\underline{x} - c}{\|\underline{a}\|}} \tag{20}$$

The arrow reminds us that we measure differences in the direction of $\underline{a}$. A natural way to read this is "*the difference of $\underline{x}$ and the offset (or offset hyperplane) in the direction of $\underline{a}$*." It follows that

$$d_{-\underline{a}}(\mathsf{H}_{\underline{a},c}, \underline{x}) = d(\mathsf{H}_{\underline{a},c} \leftarrow \underline{x}) = -d_{\underline{a}}(\mathsf{H}_{\underline{a},c}, \underline{x}) = -d(\mathsf{H}_{\underline{a},c} \to \underline{x}) \tag{21}$$

which can be read as "the difference of the offset (or offset hyperplane) and $\underline{x}$ in the direction of $-\underline{a}$."

3 Variable-Coefficient Duality

The hyperplane in the variable $\underline{x}$, orthogonal to the coefficient vector $\underline{a}$, and offset by $c/\|\underline{a}\|$ in the direction of $\underline{a}$ is defined in (7). All of the interpretation of this was done in $\underline{x}$-space – *i.e.,* we considered $\underline{x}$ as the variable and $\underline{a}$ as a vector constant. Inspecting this definition it is clear that $\underline{x}^{\mathrm{t}}\underline{a} = \underline{a}^{\mathrm{t}}\underline{x}$ so that the constraining equation of the hyperplane is symmetric in $\underline{a}$ and $\underline{x}$. Thus, we can consider the hyperplane in the variable $\underline{a}$, orthogonal to the coefficient vector $\underline{x}$, and offset by $c/\|\underline{x}\|$ in the direction of $\underline{x}$

$$\mathsf{H}_{\underline{x},c} = \{\underline{a} : \underline{x}^{\mathrm{t}}\underline{a} = c\} = \{\underline{a} : \underline{x}^{\mathrm{t}}\underline{a} - c = 0\} \quad \underline{a} \neq \underline{0} \tag{22}$$

In some instances, we may find it useful to interpret both $\mathsf{H}_{\underline{a},c}$ and $\mathsf{H}_{\underline{x},c}$.

[1] Here we not that the hyperplane partitions the space $\mathbb{M}$ into two regions and we refer to the one of these that contains the origin as the origin side.

4 Examples from Machine Learning

During the study of machine learning, we frequently interpret linear models and the associated learning processes using hyperplanes. This all fits within the formulation described above so carefully covering the general case, as done above, allows straightforward application of these results when the opportunity arises.

4.1 Linear Classifiers and the Decision Boundary

A linear classifier for the two class problem is given by

$$g(\underline{x}) = \underline{w}^{\mathrm{t}}\underline{x} + w_0 = \left[\underline{w}^{(+)}\right]^{\mathrm{t}} \underline{x}^{(+)} \underset{\Gamma_2}{\overset{\Gamma_1}{\gtrless}} 0 \tag{23}$$

where $\underline{w}^{(+)}$ is the augmented weight vector that includes w_0 and $\underline{x}^{(+)}$ is the augmented data vector with first component 1. The decision boundary is when the expression on the left is equal to zero, which is clearly a hyperplane. This hyperplane can be viewed in terms of the augmented feature vector variable or the non-augmented feature variable. We consider these next.

4.1.1 In Augmented Feature Space

Considering the decision rule (23) in terms of the variables given by the augmented feature vector $\underline{x}^{(+)}$ and the fixed coefficient vector given by the augmented weight vector, $\underline{w}^{(+)}$, we have a hyperplane passing through the origin

$$\mathsf{H}_{\underline{w}^{(+)},0} = \{\underline{x}^{(+)} : \left[\underline{w}^{(+)}\right]^{\mathrm{t}} \underline{x}^{(+)} = 0\} \quad \underline{w}^{(+)} \neq \underline{0} \tag{24}$$

This is a hyperplane passing through the origin in $\mathbb{R}^{D+1}$. An example is shown in Fig. 9. Note that here

$$d(\mathsf{H}_{\underline{w}^{(+)},0} \to \underline{x}^{(+)}) = R_{H^{\perp}_{\underline{w}^{(+)},0}}(\underline{x}^{(+)}) = \|\underline{x}^{(+)}\| \cos\theta_{x,w^{(+)}} \tag{25}$$

so that the decision rule is

$$\left[\underline{w}^{(+)}\right]^{\mathrm{t}} \underline{x}^{(+)} \underset{\Gamma_2}{\overset{\Gamma_1}{\gtrless}} 0 \quad \Longleftrightarrow \quad \cos\theta_{x,w^{(+)}} \underset{\Gamma_2}{\overset{\Gamma_1}{\gtrless}} 0 \tag{26}$$

Thus, roughly speaking, if $\underline{x}^{(+)}$ points more in the direction of $\underline{w}^{(+)}$ than in the direction of $-\underline{w}^{(+)}$, it will be classified as coming from class 1.

4.1.2 In Non-Augmented Feature Space

Considering the decision rule (23) in terms of the non-augmented feature vector $\underline{x} \in \mathbb{R}^D$, with coefficients given by the non-augmented weight vector, $\underline{w}$, we have a hyperplane in $\mathbb{R}^D$ offset from the origin by $-w_0/\|\underline{w}\|$ in the $\underline{w}$ direction

$$\mathsf{H}_{\underline{w},-w_0} = \{\underline{x} : \underline{w}^{\mathrm{t}}\underline{x} = -w_0\} = \{\underline{x} : \underline{w}^{\mathrm{t}}\underline{x} + w_0 = 0\} \quad \underline{w} \neq \underline{0} \tag{27}$$

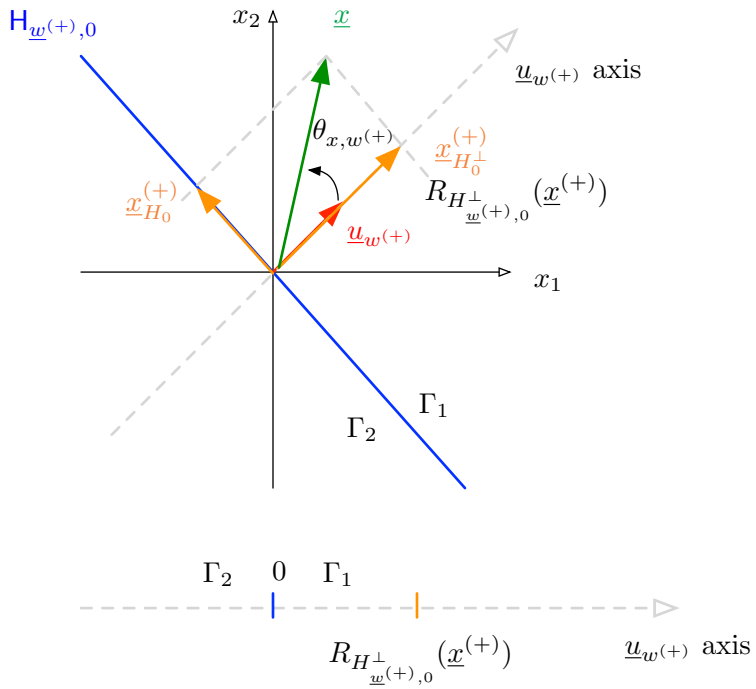


Figure 9: The decision boundary for a linear classifier in augmented feature space is a hyperplane orthogonal to $\underline{w}^{(+)}$ passing through the origin. In this example, $\cos\theta_{x,w^{(+)}} > 0$, so $\underline{x}^{(+)}$ lies in Γ_1.

This is illustrated in Fig. 10 for $M = 2$. Note in this case

$$d(\mathsf{H}_{\underline{w},-w_0} \to \underline{x}) = d_{\underline{w}}(\mathsf{H}_{\underline{w},-w_0}, \underline{x}) = R_{H^\perp_{\underline{w},0}}(\underline{x}) - \frac{-w_0}{\|\underline{w}\|} = \frac{\underline{w}^{\mathrm{t}}\underline{x} + w_0}{\|\underline{w}\|} = \frac{g(\underline{x})}{\|\underline{w}\|} \tag{28}$$

The example in Fig. 10 has $w_0 < 0$ and $d(\mathsf{H}_{\underline{w},-w_0} \to \underline{x}) > 0$. This is a value of $\underline{x}$ that falls in Γ_1. This is consistent with (17) – *i.e.*, $d(\mathsf{H}_{\underline{w},-w_0} \to \underline{x}) = g(\underline{x})/\|\underline{w}\| > 0$ implies that $\underline{x}$ is on the non-origin side of the hyperplane. If we consider the example in Fig. 10, but with $w_0 > 0$, then the decision boundary will move into quadrants 2 and 3. The condition $d(\mathsf{H}_{\underline{w},-w_0} \to \underline{x}) = g(\underline{x})/\|\underline{w}\| > 0$ will still correspond to being on the origin-side of the boundary and also in Γ_1, according to (17) and (23), respectively.

4.2 Gradient Descent Learning

A function $J(\underline{w})$ that maps a vector[2] argument $\underline{w} \in \mathbb{R}^{D+1}$ to a scalar value has gradient defined by

$$\nabla_{\underline{w}} J(\underline{w}) = \left[\begin{array}{cccc} \frac{\partial J(\underline{w})}{\partial w_0} & \frac{\partial J(\underline{w})}{\partial w_1} & \cdots & \frac{\partial J(\underline{w})}{\partial w_D} \end{array} \right]^{\mathrm{t}} \tag{29}$$

To interpret the gradient, consider the dot product with a unit vector $\underline{u}$

$$\underline{u}^t \nabla_{\underline{w}} J(\underline{w}) = \sum_{m=0}^{D} u_m \frac{\partial J(\underline{w})}{\partial w_m} \tag{30}$$

[2]In the remainder of this document, we use augmented vectors for the weights and features with implicit notation – *i.e.*, without the $(\cdot)^{(+)}$ explicit notation.

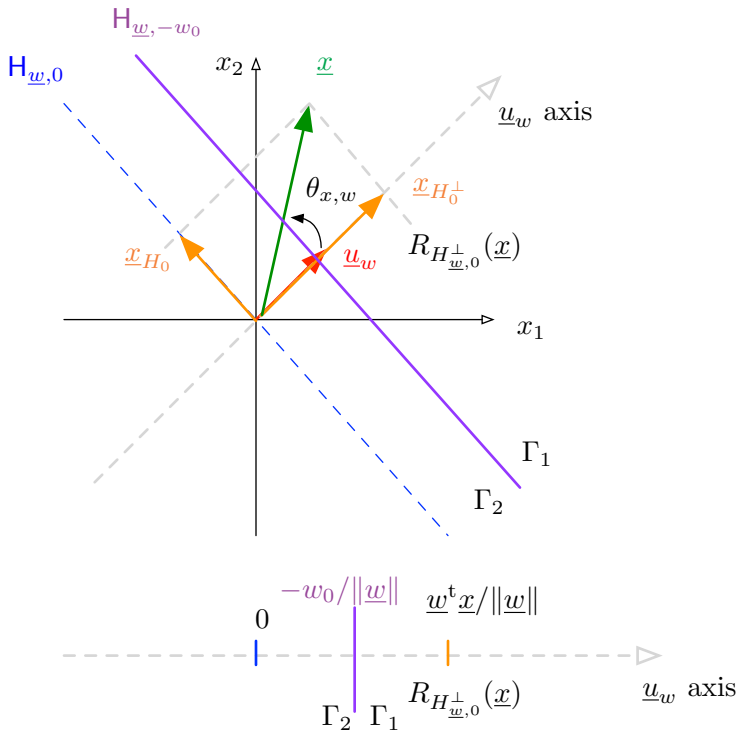


Figure 10: The decision boundary $g(\underline{x}) = 0$ in non-augmented feature space is a hyperplane in $\underline{x}$, orthogonal to $\underline{w}$, offset by $-w_0/\|\underline{w}\|$ in the direction of $\underline{w}$. Note that the difference of $\underline{x}$ in the direction of $\underline{w}$ and the decision boundary hyperplane is $d(\mathsf{H}_{\underline{w},-w_0} \to \underline{x}) = g(\underline{x})/\|\underline{w}\|$. This example is drawn with $w_0 < 0$ and $d(\mathsf{H}_{\underline{w},-w_0} \to \underline{x}) > 0$, so $\underline{x}$ lies in Γ_1.

which is the *directional derivative* in the direction of $\underline{u}$. The directional derivative is a measure of the rate of change in $J(\underline{w})$ when moving in the direction of $\underline{u}$. Specifically, for small ϵ we have the approximation

$$J(\underline{w}+\epsilon\underline{u}) \approx J(\underline{w}) + \epsilon \underline{u}^t \nabla_{\underline{w}} J(\underline{w}) \tag{31}$$

The Cauchy-Schwartz theorem implies that the direction of the maximum rate of change (steep ascent) is in the direction of the gradient itself and the direction of steepest descent is in the antipodal direction of the gradient. Specifically, letting θ be the angle between the vectors $\underline{u}$ and $\nabla_{\underline{w}} J(\underline{w})$

$$\underline{u}^t \nabla_{\underline{w}} J(\underline{w}) = \|u\| \|\nabla_{\underline{w}} J(\underline{w})\| \cos\theta = \|\nabla_{\underline{w}} J(\underline{w})\| \cos\theta \tag{32}$$

This is maximized when $\theta = 0$ and minimized when $\theta = \pi$. It follows that the direction of steepest ascent, $\underline{u}_{\max}$, and descent, $\underline{u}_{\min}$, are

$$\underline{u}_{\max} = \frac{\nabla_{\underline{w}} J(\underline{w})}{\|\nabla_{\underline{w}} J(\underline{w})\|} \qquad \underline{u}_{\min} = \frac{-\nabla_{\underline{w}} J(\underline{w})}{\|\nabla_{\underline{w}} J(\underline{w})\|} \tag{33}$$

We can use the approximation in (31) to express a first order approximation of $J(\underline{w})$ around a given value of $\underline{w}$. Specifically, consider a learning approach where the current value for the parameter estimate vector is $\underline{w}(i)$ and it is desired to update this to obtain, hopefully, a value for $\underline{w}(i+1)$ that has a lower criterion function – *i.e.,* we seek $\underline{w}(i+1)$ with $J(\underline{w}(i+1)) < J(\underline{w}(i))$. If the step $\underline{w}(i+1) - \underline{w}(i)$ is small, we can use an approximation of $J(\underline{w})$ to seek the vector $\underline{w}(i+1)$. In fact, we can use a first-order approximation of $J(\underline{w})$ for this purpose

$$J(\underline{w}) \approx \hat{J}_{1,\underline{w}(i)}(\underline{w}) = J(\underline{w}(i)) + \left[\nabla_{\underline{w}} J(\underline{w}(i))\right]^{\mathrm{t}} (\underline{w} - \underline{w}(i)) = \underline{a}^t \underline{w} - c \tag{34}$$

where $\underline{a} = \nabla_{\underline{w}} J(\ \underline{w}(i))$ and $c = J(\underline{w}(i)) - \left[\nabla_{\underline{w}} J(\ \underline{w}(i))\right]^{\mathrm{t}} \underline{w}(i)$.

Inspecting (34), we see it is closely related to the expression[3] for a hyperplane offset from the origin in (7). In fact, if we consider the surface $\hat{J}_{1,\underline{w}(i)}(\underline{w})$ it is a hyperplane in $\mathbb{R}^{D+2}$ and is the linear approximation to $J(\underline{w})$ around the point $\underline{w}(i)$. For example, if $D = 1$ so that $\underline{w} = w$ is a scalar, then $\hat{J}_{1,w(i)}(w)$ is the familiar tangent line to the curve $J(w)$

$$\hat{J}_{1,w(i)}(w) = J(w(i)) + \dot{J}(w(i))[w - w(i)] \tag{35}$$

where $\dot{J}(w)$ is the derivative of $J(w)$. This concept is shown in Fig. 11 for one dimensional $\underline{w}$.

As a proxy for minimizing $J(\underline{w})$, we can take a small step in the direction of $-\nabla_{\underline{w}} J(\underline{w}(i))$ from the current estimate $\underline{w}(i)$. This is the *method of steepest descent or gradient descent (GD)*

$$\boxed{\underline{w}(i+1) = \underline{w}(i) - \eta(i) \nabla_{\underline{w}} J(\underline{w}(i))} \tag{36}$$

where $\eta(i) > 0$ is the *step size or learning rate at iteration* i. Note that this is equivalent to

$$[\underline{w}(i+1)]_j = [\underline{w}(i)]_j - \eta(i) \left.\frac{\partial J(\underline{w})}{\partial w_j}\right|_{w_j = w_j(i)} \qquad j = 0, 1, \ldots D \tag{37}$$

or, in word, we update the j^{th} component by the associated partial derivative.

[3] This is a hyperplane with variable vector $[\hat{J}_{1,w(i)}(\underline{w})\ \underline{w}^{\mathrm{t}}]^{\mathrm{t}}$, coefficient vector $[\ 1\ \underline{a}^{\mathrm{t}}\ ]^{\mathrm{t}}$, and constant c. We plot $\hat{J}_{1,w(i)}(\underline{w})$ in (34) analogously to the approach used in Fig. 2.

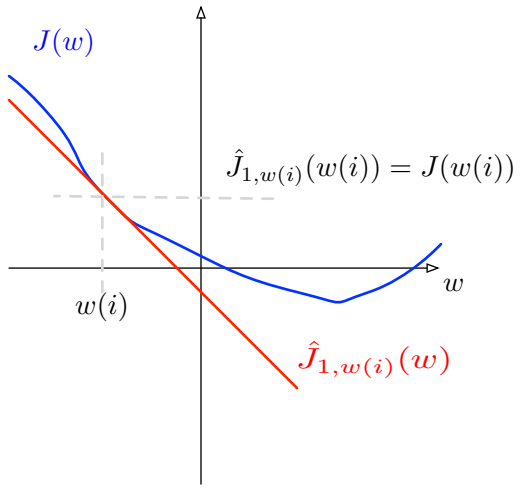


Figure 11: The first order approximation when $\underline{w}$ is one dimensional.

Let us consider this gradient descent method for a quadratic criterion function of the form

$$J(\underline{w}) = \frac{1}{2}\underline{w}^{\mathrm{t}}\underline{\underline{C}}\,\underline{w} + \underline{b}^{\mathrm{t}}\underline{w} + d \tag{38}$$

where $\underline{\underline{C}}$ is a symmetric matrix. This has gradient

$$\nabla_{\underline{w}} J(\underline{w}) = \underline{\underline{C}}\,\underline{w} + \underline{b} \tag{39}$$

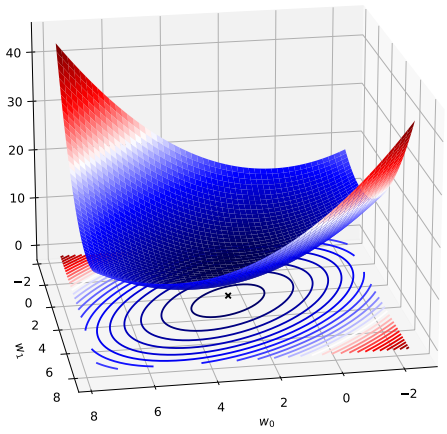
Consider the specific example of

$$\underline{\underline{C}} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} -1 \\ -2 \end{bmatrix}, \quad d = 10 \tag{40}$$

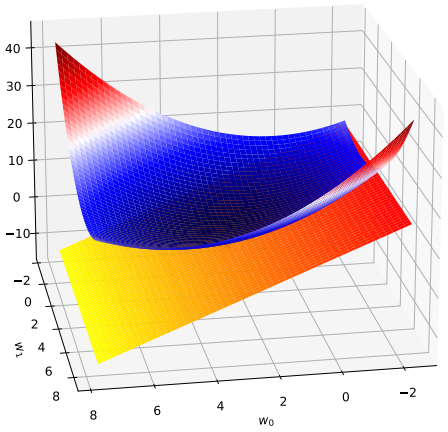
This is a convex function so that the critical point, $\underline{w}^*$, solving $\nabla_{\underline{w}} J(\underline{w}^*) = 0$ minimizes $J(\underline{w})$:

$$\underline{w}^* = \frac{1}{3}\begin{bmatrix} 8 \\ 9 \end{bmatrix} \tag{41}$$

A plot of this surface is shown in Fig. 12a which also shows the contours in the (w_0, w_1) plane. The first order approximation at $w(i) = \underline{0}$ is shown in Fig. 12b. From Fig. 12b it is clear that $\hat{J}_{1,w(i)}(w)$ is only a *local* approximation of the function $J(\underline{w})$ around the point of approximation (*i.e.,* around the origin in this example). The gradient descent algorithm in (36) is certainly reasonable assuming that $\|\eta(i)\nabla_{\underline{w}} J(\underline{w}(i))\|$ is small, but using a step value that is too large can result in an increase in the criterion function. For example, taking a large step down the tangent plane in Fig. 12b will result in passing the point $\underline{w}^*$ and the result will be $J(\underline{w}(i+1)) > J(\underline{w}(i))$. With this first order approximation, care must therefore be taken to choose a small enough step size. Another, more complex, approach is to use a second order approximation of $J(\underline{w})$ – *i.e.,* so-called second-order optimization methods. The contours from Fig. 12a are shown again in Fig. 13 where we have also indicated the direction of steepest descent at a few locations. Notice that the direction of steepest descent (the negative gradient) is orthogonal to the level-curves. Also, the norm of the gradient is larger where $J(\underline{w})$ is changing more rapidly – *i.e.,* where the contours are closer together.



(a) The example criterion function $J(\underline{w})$ with contours and the minimizing value of $\underline{w}$ shown.



(b) The first order approximation at $\underline{w}(i) = \underline{0}$

Figure 12: An example of a quadratic $J(\underline{w})$ with the first order approximation.

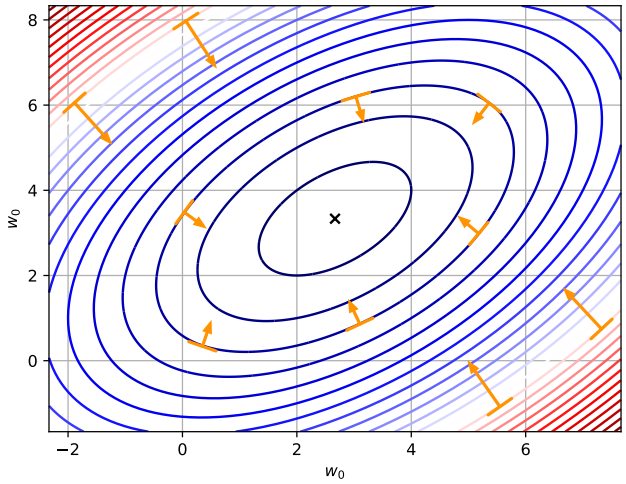


Figure 13: The direction of steepest descent added to the contour plots for the quadratic criterion function from Fig. 12. Note that the direction of steepest descent is always orthogonal to the contour lines of $J(\underline{w})$ and the gradient has larger norm in areas where the contours are closer together.

4.2.1 Gradient Descent Update in the Perceptron Learning Algorithm

The two-class perceptron classifier criterion function, assuming augmented feature and weight vectors, is

$$J(\underline{w}) = \sum_{n=1}^{N} J_n(\underline{w}) \tag{42a}$$

$$J_n(\underline{w}) = -[\![z_n \underline{w}^{\mathrm{t}} \underline{x}_n \leq 0]\!] z_n \underline{w}^{\mathrm{t}} \underline{x}_n = [\![z_n \underline{w}^{\mathrm{t}} \underline{x}_n \leq 0]\!] |\underline{w}^{\mathrm{t}} \underline{x}_n| \tag{42b}$$

where $z_n = +1$ and $z_n = -1$ for class 1 and class 2 values of $\underline{x}_n$, respectively, and $[\![\cdot]\!]$ is the indicator function. This loss only penalizes misclassifications and the penalty for a misclassification is $-z_n \underline{w}^{\mathrm{t}} \underline{x}_n = |\underline{w}^{\mathrm{t}} \underline{x}_n|$, since $-z_n \underline{w}^{\mathrm{t}} \underline{x}_n \leq 0$ for an error. Since the criterion function is $g(\underline{w}^{\mathrm{t}} \underline{x}_n)$, the indicator function indicates on which side of the decision boundary hyperplane, discussed in Section 4.1.1, the point $\underline{x}_n$ lies. The gradient for perceptron loss is

$$\nabla_{\underline{w}} J_n(\underline{w}) = -[\![z_n \underline{w}^{\mathrm{t}} \underline{x}_n \leq 0]\!] z_n \underline{x}_n = \begin{cases} -\underline{x}_n & \text{error occurs and } \underline{x}_n \text{ is from class 1} \\ +\underline{x}_n & \text{error occurs and } \underline{x}_n \text{ is from class 2} \\ \underline{0} & \text{no classification error for } \underline{x}_n \end{cases} \tag{43}$$

Consider a gradient descent update with a single point (mini-batch size 1), then we have

$$\underline{w}(i+1) = \underline{w}(i) + \eta(i) [\![z_n \underline{w}^{\mathrm{t}}(i) \underline{x}_n \leq 0]\!] z_n \underline{x}_n \tag{44}$$

If we consider the case when the $\underline{x}_n$ is on the decision boundary defined by the current weight vector $\underline{w}(i)$, then

$$\mathsf{H}_{z_n \underline{x}_n, 0} = z_n \underline{w}^{\mathrm{t}} \underline{x}_n = 0 \tag{45}$$

is a hyperplane in the variable $\underline{w}$, passing through the origin, with coefficient vector $\underline{x}_n$. From (44), this implies that the gradient descent update is always orthogonal to the data vector $\underline{x}_n$. The learning processes can thus be viewed in $\underline{w}$-space as shown in Fig. 14.

4.2.2 Gradient Descent Update in the LMS Algorithm

For a mean-squared error or L2 loss, the criterion function is again additive across data points with, once more, $\underline{x}_n$ and $\underline{w}$ implicitly in augmented form,

$$J_n(\underline{w}) = (y_n - \underline{w}^{\mathrm{t}} \underline{x}_n)^2 \tag{46}$$

The gradient for each data point is

$$\nabla_{\underline{w}} J_n(\underline{w}) = 2(\underline{w}^{\mathrm{t}} \underline{x}_n - y_n) \underline{x}_n \tag{47}$$

so that a single-point GD update is

$$\underline{w}(i+1) = \underline{w}(i) - \eta(i)(\underline{w}^{\mathrm{t}}(i) \underline{x}_n - y_n) \underline{x}_n \tag{48}$$

where we have absorbed a factor of 2 in the gradient into the learning rate. Note that, as was the case in perceptron learning, the weight update is a scalar multiple of the current data point $\underline{x}_n$. Letting $\epsilon_n = (w^{\mathrm{t}} \underline{x}_n - y_n)$, we can consider the condition

$$\epsilon_n = \underline{x}_n^{\mathrm{t}} \underline{w} - y_n = 0 \tag{49}$$

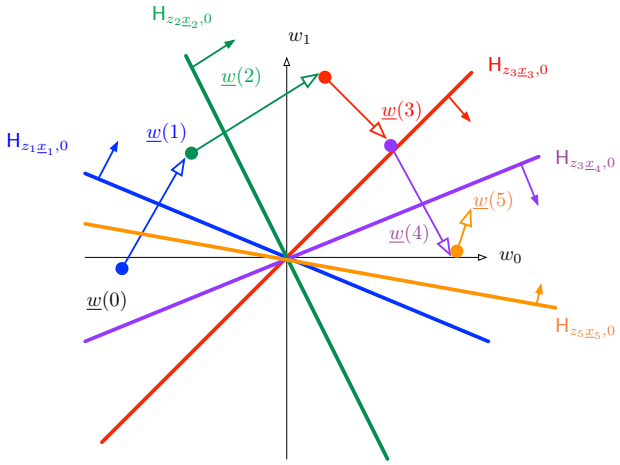


Figure 14: The learning process for perceptron learning gradient descent with mini-batch size 1. At each update, the weight vector is updated in the direction of $z_n\underline{x}_n$. The vector $z_n\underline{x}_n$ is a normal vector of $\mathsf{H}_{z_n\underline{x}_n,0}$ which is the current boundary for correct classification. This normal vector is illustrated nominally as the small vector on each of the hyperplanes so that correct classification is on the arrow side of the hyperplane. Note that we have assumed that each of the data vectors $\underline{x}_n$ for $n = 1, 2, 3, 4$ are misclassified by the current weight-vector – *i.e.,* in practice, there will not be a weight update for every data vector as some will be correctly classified by the current weight vector.

which defines a hyperplane in the augmented weight vector space, with coefficient vector $\underline{x}_n$, which is offset from the origin by $y_n/\|\underline{w}\|$ in the direction of $\underline{x}_n$ – *i.e.,* $\mathsf{H}_{\underline{x}_n,y_n}$. For this hyperplane, it follows that

$$d(\mathsf{H}_{\underline{x}_n,-y_n} \to \underline{w}) = \frac{\underline{x}_n^{\mathrm{t}}\underline{w} - y_n}{\|\underline{x}_n\|} = \frac{\epsilon_n}{\|\underline{x}_n\|} \tag{50}$$

The LMS learning process can be visualized in $\underline{w}$-space and it is very similar to the process shown in Fig. 14. At the i^{th} step in the LMS algorithm, $w(i)$ is updated in the direction[4] of $\epsilon_n(i)\underline{x}_n$, which is also the orthogonal direction towards the offset hyperplane $\mathsf{H}_{\underline{x}_n,y_n}$. In other words, the update is in the direction that is shortest to reach $\epsilon_n = 0$. One step of the LMS algorithm is illustrated in Fig. 15.

4.3 General, Single Point GD for a Linear Model

The expressions in (43) and (47) are similar in that the gradient is a scalar multiple of the data vector $\underline{x}_n$. Thus, with a mini-batch size of 1, the GD update is a step in the direction of $\pm\underline{x}_n$. Note that this will always be the case for a linear model since $J_n(\underline{w}) = f(\underline{w}^{\mathrm{t}}\underline{x}_n)$ for some scalar function $f(\cdots)$ and, via the chain rule, we have

$$\nabla_{\underline{w}} J_n(\underline{w}) = \nabla_{\underline{w}} f(\underline{w}^{\mathrm{t}}\underline{x}_n) = \dot{f}(\underline{w}^{\mathrm{t}}\underline{x}_n)\underline{x}_n \tag{51}$$

[4]Here $\epsilon_n(i)$ is the value of ϵ_n computed with $\underline{w}(i)$.

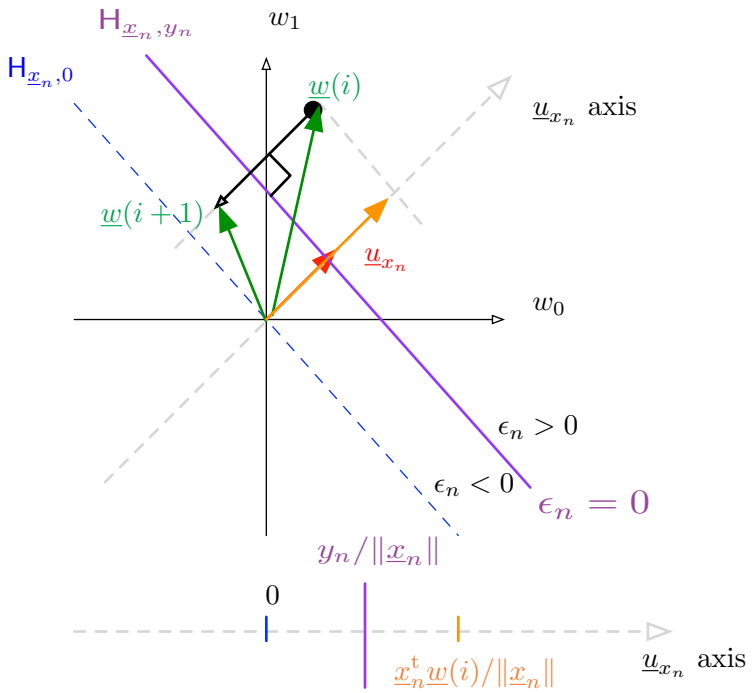


Figure 15: One update of the LMS algorithm. Note that, since $\underline{w}(i)$ is shown on the non-origin side of $\mathsf{H}_{\underline{x}_n,y_n}$, $\epsilon_n(i) > 0$ and the update moves in the direction of $-\underline{x}_n$ which is the direction towards the hyperplane where $\epsilon_n = 0$. Depending on the value of $\epsilon_n(i)$ and $\eta(i)$, the next weight vector $w(i+1)$ may be on either side of $\mathsf{H}_{\underline{x}_n,y_n}$. In this example, the LMS step places $\underline{w}(i+1)$ on the origin side of this hyperplane.

where $\dot{f}(v) = df(v)/dv$. The learning of any GD algorithm for a linear model follows a similar geometry to that shown in Figures 14 and 15. Specifically, if a single-point is used to compute the gradient approximation, the step is in the direction of $\pm\underline{x}_n$. For larger batch sizes (including the full training set), the gradient is a weighted average of $\{\underline{x}_n\}_{n\in\mathcal{B}}$ where $\mathcal{B}$ is the set of points in the mini-batch used for the iteration. The weighting of these points is a function of the loss function; in particular the resulting gradient form. For example, in the case of perceptron learning, the update is simply proportional to $\sum_{\mathcal{B}} z_n\underline{x}_n$ – *i.e.,* the sum over all the reflected data vectors in the mini-batch that are misclassified with the current value of $w(i)$. In the MSE case, the weighted sum is $\sum_{\mathcal{B}} \epsilon_n(i)\underline{x}_n$ so that data points that have larger error contribute more to the sum. More generally, referring to (51), the gradient used for the update is

$$\frac{1}{|\mathcal{B}|}\sum_{\mathcal{B}} \dot{f}(\underline{w}^{\mathrm{t}}\underline{x}_n)\underline{x}_n \tag{52}$$