CS/ENGRI 172, Fall 2002

9/18/02: Lecture Nine Handout

Topics: A framework for machine learning.
Announcements: Due to a rescheduled engineering faculty event this week, Prof. Lee is trading her Thursday office hour with Amanda Holland-Minkley's Friday office hour. Hence, the locations, but not times, of office hours for the remainder of this week *only* are slightly altered:

W 3-4, Upson 4152 (Lee)
R 1:40-2:40, Upson 328B (Heifets)
R 2:40-4:10, Upson 4116 (Holland-Minkley)
F 11:15-12:15, Upson 328B (Baker)
F 1:30-2:30, Upson 4152 (Lee)

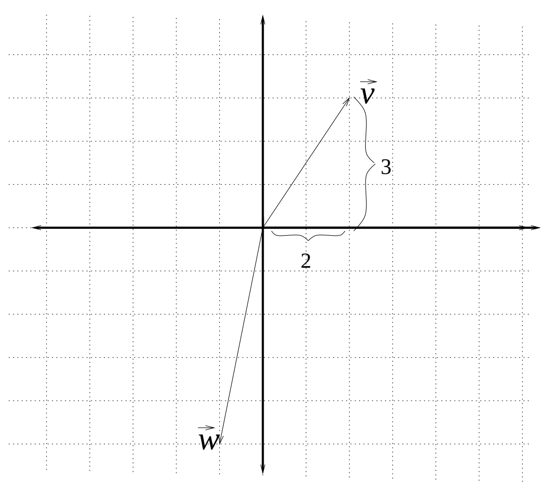
Vector notation for function input

In this course, our view of machine learning will be that of learning a function, which is actually a quite general framework. We will consider the input to the functions to be learned to be (numerically-valued, finite) *feature vectors*, or ordered tuples. An example of our notation is as follows, indicating that $\overrightarrow{x}$ is an n-dimensional vector with its n *components* being x_1 through x_n:

$$\overrightarrow{x} = (x_1, x_2, \ldots, x_n).$$

It is standard to indicate vectors themselves with an arrow or boldfacing, to distinguish them from components, which are single numbers ("scalars").

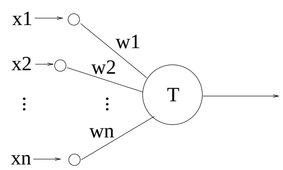
Two-dimensional vectors are probably the most familiar to you, since they can be associated with points in the plane. For example, if $\overrightarrow{v} = (2, 3)$ and $\overrightarrow{w} = (-1, -5)$, then we have the following picture:



A confusing fact of life is that it is common to use the letters "x" and "y" to indicate both vector names, vector components, and, in the case of two-dimensional vectors, coordinates; for example, "the x-coordinate of the vector $\overrightarrow{x} = (x_1, x_2)$ is x_1"). So it goes.

Perceptrons

We will first restrict attention to functions computable by *perceptrons*, which are idealizations of single neuron.



Perceptrons are characterized by a weight vector $\overrightarrow{w}$ and a threshold T. Letting n be the dimensionality of $\overrightarrow{w}$, a perceptron "fires" (outputs a one) on input $\overrightarrow{x} = (x_1, x_2, \ldots, x_n)$ if $w_1x_1 + w_2x_2 + \cdots + w_nx_n \geq T$, and outputs zero otherwise.

The inner product

The *inner product* (sometimes *dot product*) between two vectors of the same dimensionality is an important and useful concept. For two vectors $\overrightarrow{v} = (v_1, v_2, \ldots, v_n)$ and $\overrightarrow{w} = (w_1, w_2, \ldots, w_n)$, the inner product is defined as follows:

$$\overrightarrow{v} \cdot \overrightarrow{w} \stackrel{def}{=} \sum_{i=1}^{n} v_i w_i = v_1w_1 + v_2w_2 + \cdots v_nw_n$$

For example, if $\overrightarrow{v} = (2, 3)$ and $\overrightarrow{w} = (-1, -5)$, then $\overrightarrow{v} \cdot \overrightarrow{w} = -2 - 15 = -17$.

The *length* length$(\overrightarrow{v})$ of a vector $\overrightarrow{v}$ can be computed via the inner product:

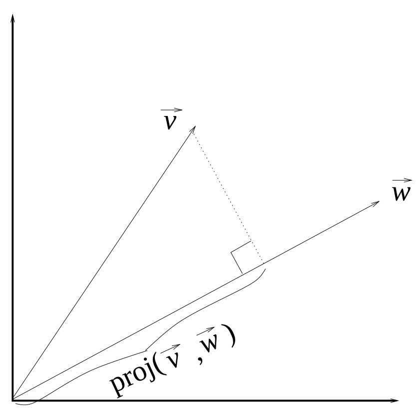
$$\text{length}(\overrightarrow{v}) = \sqrt{\overrightarrow{v} \cdot \overrightarrow{v}}\ .$$

For example, consider the two-dimensional case: $\text{length}((v_1, v_2)) = \sqrt{(v_1, v_2) \cdot (v_1, v_2)} = \sqrt{v_1^2 + v_2^2}$, which is exactly the Pythagorean theorem.

An extremely handy fact is the following identity:

$$\begin{aligned} \overrightarrow{v} \cdot \overrightarrow{w} &= \text{length}(\overrightarrow{v})\text{length}(\overrightarrow{w}) \cos(\angle(\overrightarrow{v}, \overrightarrow{w})) \\ &= \text{length}(\overrightarrow{w})\text{proj}(\overrightarrow{v}, \overrightarrow{w}) \end{aligned}$$

(where for notational convenience we consider the projection of one vector onto another to be a length, i.e. a scalar, rather than a vector).



From this, we can infer that a perceptron classifier corresponds to a *half-plane* concept.