stratus
innovations group

Machine Learning Math: Understanding Classification Problems

Classification Problems Return One Item From a Discrete Set

In our previous blog posts, we covered *regression problems* where we're trying to predict the value of our target function. In regression problems, our target function is a real-valued function, and we're trying to compute a real number that estimates its value at given data points.

In the next couple of posts, we'll cover *classification problems,* which are also known as *selection problems*. In a classification problem, our target function is returning one item from a discrete set of choices: in other words, it's making a "true or false" decision or picking one option out of a few. Such a function might look at an image and determine whether the image contains a person's face, or it could look at a hand-written letter or number and identify the character.

As before, we have a data set:

$$D = \{ (x_1, y_1), (x_2, y_2), \dots , (x_N, y_N) \}$$

Except that instead of the y_n values being real numbers, we have

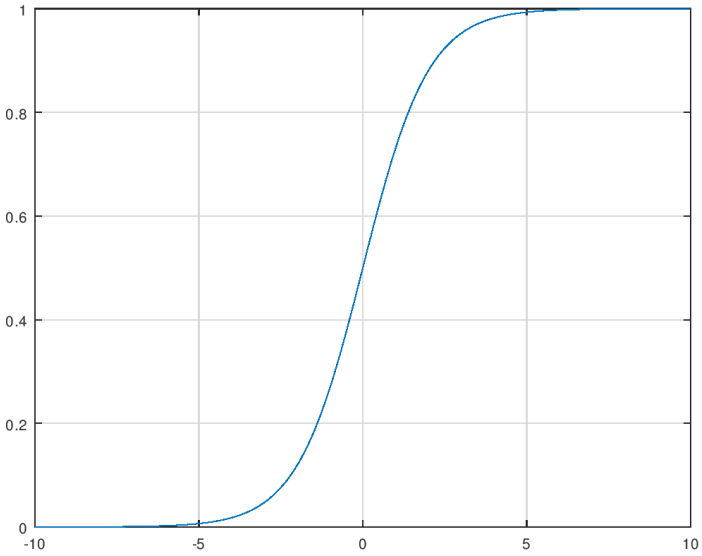
$$y_n \epsilon \{0, 1, 2, \dots, s\}$$

One way to do this is to compute the probability of y_n being each of those values and picking the one with the highest probability. In other words, we want to find the value of s that maximizes the value of $P(y_n = s | x_n)$. The simplest case is when we're trying to make a "true-false" or "yes-no" decision, in which case we have $y_n \epsilon \{0, 1\}$, and we'll start our discussion with this case.

We're computing probabilities, so for our hypothesis sets, we'll want functions that return values in the interval [0,1]. A good choice for such a function is the *logistic function*, defined as:

$$f(x) = \frac{1}{1 + e^{-x}}$$

For large positive values of x, the value of e^{-x} approaches zero, so the value of $f(x)$ approaches 1. For large negative values of x, the value of e^{-x} becomes arbitrarily large, so the value of $f(x)$ approaches zero. The graph of this function is roughly:



The derivative of this function, which we'll use later, is:

$$\begin{aligned}
\frac{d}{dx}f(x) &= \frac{d}{dx}\frac{1}{1+e^{-x}} \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1+e^{-x}-1}{(1+e^{-x})^2} \\
&= \frac{1+e^{-x}}{(1+e^{-x})^2} - \frac{1}{(1+e^{-x})^2} \\
&= \frac{1}{1+e^{-x}} - \frac{1}{(1+e^{-x})^2} \\
&= \frac{1}{1+e^{-x}}\left(1 - \frac{1}{1+e^{-x}}\right) \\
&= f(x)(1-f(x))
\end{aligned}$$

To construct our hypothesis set, we'll select weights w=(w_0,w_1,…,w_d) as we did before, and define:

$$h_{\mathrm{w}}(\mathrm{x}) = \frac{1}{1+e^{-\mathrm{w}\cdot\mathrm{x}}}$$

where we add an x_0 = 1 component to each x_n. Our hypothesis set is the set of all such functions.

Then for (x_N, y_N)ϵ D, we define:

$$P_{\mathrm{w}}(y_n = 1|\mathrm{x}_n) = h_{\mathrm{w}}(\mathrm{x}_n)$$
$$P_{\mathrm{w}}(y_n = 0|\mathrm{x}_n) = 1 - h_{\mathrm{w}}(\mathrm{x}_n)$$

Readers with some experience in probability will recognize this as a *Bernoulli distribution*. Since y_n can only be zero or one, we can write this more succinctly as:

$$P_{\mathrm{w}}(y_n|\mathrm{x}_n) = h_{\mathrm{w}}(\mathrm{x}_n)^{y_n}\big(1 - h_{\mathrm{w}}(\mathrm{x}_n)\big)^{1-y_n}$$

To find our final hypothesis *g(*x*)*, we want to find the w that maximizes the *likelihood*:

$$L(\mathrm{w}) = \prod_{n=1}^{N} P_{\mathrm{w}}(y_n|\mathrm{x}_n)$$
$$= \prod_{n=1}^{N} h_{\mathrm{w}}(\mathrm{x}_n)^{y_n}\big(1 - h_{\mathrm{w}}(\mathrm{x}_n)\big)^{1-y_n}$$

This is equivalent to minimizing the function $-l(\mathrm{w}) = -\ln L(\mathrm{w})$, where ln is the natural logarithm. We can use the gradient descent algorithm to perform this minimization. We start by computing the gradient of $-l(\mathrm{w})$. The i^{th} component of $\nabla(-l(\mathrm{w}))$ is given by:

$$
\begin{aligned}
\frac{\partial}{\partial w_i}(-\ln L(\mathrm{w})) &= -\frac{\partial}{\partial w_i}\sum_{n=1}^{N} y_n \ln h_{\mathrm{w}}(\mathrm{x}_n) + (1-y_n)\ln\big(1-h_{\mathrm{w}}(\mathrm{x}_n)\big) \\
&= -\sum_{n=1}^{N} \frac{\partial}{\partial w_i} y_n \ln h_{\mathrm{w}}(\mathrm{x}_n) + \frac{\partial}{\partial w_i}(1-y_n)\ln\big(1-h_{\mathrm{w}}(\mathrm{x}_n)\big) \\
&= -\sum_{n=1}^{N} \left[y_n \frac{1}{h_{\mathrm{w}}(\mathrm{x}_n)} - (1-y_n)\frac{1}{1-h_{\mathrm{w}}(\mathrm{x}_n)}\right]\frac{\partial}{\partial w_i} h_{\mathrm{w}}(\mathrm{x}_n) \\
&= -\sum_{n=1}^{N} \left[y_n \frac{1}{h_{\mathrm{w}}(\mathrm{x}_n)} - (1-y_n)\frac{1}{1-h_{\mathrm{w}}(\mathrm{x}_n)}\right] h_{\mathrm{w}}(\mathrm{x}_n)\big(1-h_{\mathrm{w}}(\mathrm{x}_n)\big)\frac{\partial}{\partial w_i}\mathrm{w}\cdot\mathrm{x}_n \\
&= -\sum_{n=1}^{N} \big[y_n\big(1-h_{\mathrm{w}}(\mathrm{x}_n)\big) - (1-y_n)h_{\mathrm{w}}(\mathrm{x}_n)\big]\mathrm{x}_{n_i} \\
&= -\sum_{n=1}^{N} \big(y_n - h_{\mathrm{w}}(\mathrm{x}_n)\big)\mathrm{x}_{n_i} \\
&= -\sum_{n=1}^{N} \left(y_n - \frac{1}{1+e^{-\mathrm{w}\cdot\mathrm{x}_n}}\right)\mathrm{x}_{n_i} \\
&= \sum_{n=1}^{N} \left(\frac{1}{1+e^{-\mathrm{w}\cdot\mathrm{x}_n}} - y_n\right)\mathrm{x}_{n_i}
\end{aligned}
$$

We use this to modify the update step of the gradient descent algorithm:

1. Repeat …
    a. Set w′ = w
    b. For i = 1, 2, …, d
        Set $w_i = w_i' - \alpha\sum_{n=1}^{N}\left(\frac{1}{1+e^{-\mathrm{w}'\cdot\mathrm{x}_n}} - y_n\right)\mathrm{x}_{ni}$

    … Until $\|\mathrm{w} - \mathrm{w}'\| < \varepsilon$.

And similarly, for stochastic gradient descent:

3. Repeat …
    a. Set w′ = w
    b. Select a random $(\mathrm{x}_n, y_n) \in D$
    c. For i = 1, 2, …, d
        Set $w_i = w_i' - \alpha\left(\frac{1}{1+e^{-\mathrm{w}'\cdot\mathrm{x}_n}} - y_n\right)\mathrm{x}_{ni}$

    … Until $\|\mathrm{w} - \mathrm{w}'\| < \varepsilon$.

The gradient descent algorithm will allow us to find the weights w that we then use to define our final hypothesis:

$$g(\mathrm{x}) = P_{\mathrm{w}}(y = 1|\mathrm{x}) = \frac{1}{1 + e^{-\mathrm{w}\cdot\mathrm{x}}}$$

In other words, $g(\mathrm{x})$ is the probability that $y=1$. Then, to select our prediction for y, we pick some threshold t in [0,1], and we choose $y=0$ if $g(\mathrm{x}) < t$ and choose $y=1$ otherwise.

In our next few blog posts, we'll discuss how to extend the technique presented here to select y from a larger set of possible values. We'll also discuss the concepts of the *precision* and *recall* of a classification model and see how we might use these concepts to pick a threshold value t that's appropriate for the problem we're trying to solve.

To make sure you don't miss our next post about the mathematics behind machine learning algorithms, check back on our site in the coming weeks or contact us and let us know you want to start receiving our exclusive newsletter.

Stratus Innovations Group: Turning Complex Machine Learning Concepts Into Real Business Solutions

While the math behind machine learning can be abstract and complicated, the problems it solves are concrete, and its results can have a meaningful impact on your business' bottom line. From predictive maintenance to business analytics and beyond, the machine learning experts at Stratus Innovations Group can help you develop a customized cloud-based solution to solve your unique business challenges.

To learn more about how Stratus Innovations can help your company become more profitable and efficient, call us at 844-561-6721 or fill out our quick and easy online contact form. We'll help you bring the power of the cloud to your operations!