Lectures on Machine Learning

Jeremy Teitelbaum

Contents

1 Logistic Regression

Suppose that we are trying to convince customers to buy our product by showing them advertising. Our experience teaches us that there is no deterministic relationship between how often a potential customer sees one of our ads and whether or not they purchase our product, nevertheless it is the case that as they see more ads they become more likely to make a purchase. Logistic regression is a statistical model that can capture the essence of this idea.

To make this problem more abstract, let's imagine that we are trying to model a random event that depends on a parameter. As in our introduction above, the random event might be a user deciding to make a purchase from a website, which, in our very simple model, depends on how many times the user saw an advertisement for the product in question. But we could imagine other situations where the chance of an event happening depends on a paramter. For example, we could imagine that a student's score on a certain test depends on how much studying they do, with the likelihood of passing the test increasing with the amount of studying.

To construct this model, we assume that the probability of a certain event p is related to some parameter x by the following relationship:

$$\log \frac{p}{1-p} = ax + b \tag{1.1}$$

where a and b are constants. The quantity $\frac{p}{1-p}$ is the "odds" of the event occurring. We often use this quantity colloquially; if the chance of our team winning a football game is 1 in 3, then we would say the odds of a win are 1-to-2, which we can interpret as meaning they are twice as likely to lose as to win. The quantity $\log \frac{p}{1-p}$ is, for obvious reasons, called the log-odds of the event.

The assumption in 1.1 can be written

$$\frac{p}{1-p} = e^{ax+b}$$

and we interpret this as telling us that if the parameter x increases by 1, the odds of our event happening go up by a factor of e^a. So, to be even more concrete, if $a = \log 2$, then our logistic model would say that an increase of 1 in our parameter x doubles the odds of our event taking place.

In terms of the probability p, equation 1.1 can be rewritten

$$p = \frac{1}{1 + e^{-ax-b}}$$

This proposed relationship between the probability p and the parameter x is called the *logistic model*. The function

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

is called the *logistic function* and yields an S-shaped curve.

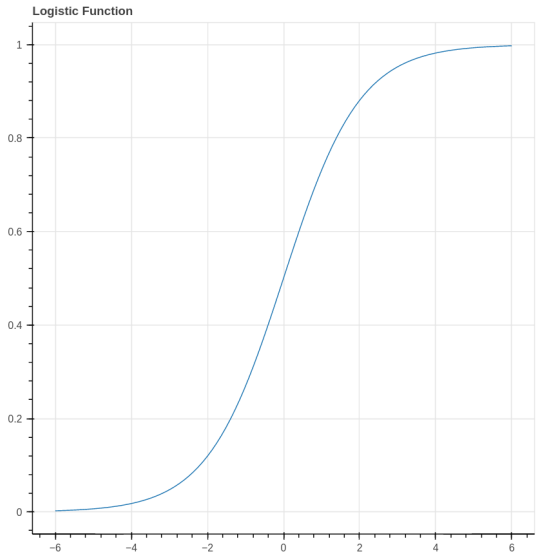

Figure 1.1: Logistic Curve

To fully put the logistic model in perspective, let's choose some explicit parameters and look at what data arising from such a model would look like. Imagine therefore that $a = \log 2$ and $b = 0$, so that the probability of the event we are interested occurring is given by the formula

$$p(x) = \frac{1}{1 + e^{-(\log 2)x}} = \frac{1}{1 + (.5)^x}.$$

Our data consists of counts of how often our event happened for a range of values of x. To generate this data, we can pick x values from the set $\{-3, -2, -1, 0, 1, 2, 3\}$ yielding probabilities $\{.11, .2, .33, .4, .56, .67, .8\}$. Now our data consists of, for each value of x, the result of 100 independent Bernoulli trials with probability $p(x)$. For example, we might find that our event occurred $\{10, 18, 38, 50, 69, 78, 86\}$ times respectively for each of the x values.

1.1 Likelihood and Logistic Regression

In applications, our goal is to choose the parameters of a logistic model to accurately predict the likelihood of the event under study occurring as a function of the measured parameter. Let's imagine that we collected the data that we generated above, without knowing that it's source was a logistic model. So 1.1 shows the number of times the event occurred, for each of the measured values of the x parameter.

Table 1.1: Sample Data

x	-3	-2	-1	0	1	2	3
Occurrences (out of 100)	10	18	38	50	69	78	86

Our objective now is to find a logistic model which best explains this data. Concretely, we need to estimate the coefficients a and b that yield

$$p(x) = \frac{1}{1 + e^{-ax-b}} \tag{1.2}$$

where the resulting probabilities best estimate the data. As we have seen, this notion of "best" can have different interpretations. For example, we could approach this from a Bayesian point of view, adopt a prior distribution on the parameters a and b, and use the data to obtain this prior and obtain a posterior distribution on a and b. For this first look at logistic regression, we will instead adopt a "maximum likelihood" notion of "best" and ask what is the most likely choice of a and b to yield this data.

To apply the maximum likelihood approach, we need to ask "for (fixed, but unknown) values of a and b, what is the likelihood that a logistic model with those parameters would yield the data we have collected? Each column in 1.1 represents 100 Bernoulli trials with a fixed probability $p(x)$. So, for example, the chance q of obtaining 10 positive results with $x = -3$ is given by

$$q(-3) = Cp(-3)^{10}(1 - p(-3))^{90}$$

where C is a constant (it would be a binomial coefficient). Combining this for different values of x, we see that the likelihood of the data is the product

$$L(a, b) = C'p(-3)^{10}(1 - p(-3))^{90}p(-2)^{18}(1 - p(-2))^{82} \cdots p(3)^{86}(1 - p(3))^{14}$$

where C' is another constant. Each $p(x)$ is a function of the parameters a and b, so all together this is a function of those two parameters. Our goal is to maximize it.

One step that simplifies matters is to consider the logarithm of the likelihood:

$$\log L(a, b) = \sum_{i=0}^{6} [x_i \log(p(x_i)) + (100 - x_i) \log(1 - p(x_i))] + C''$$

where C'' is yet another constant. Since our ultimate goal is to maximize this, the value of C'' is irrelevant and we can drop it.

1.1.1 Another point of view on logistic regression

In 1.1 we summarize the results of our experiments in groups by the value of the x parameter. We can think of the data somewhat differently, by instead considering each event separately, corresponding to a parameter value x and an outcome 0 or 1. From this point of view the data summarized in 1.1 would correspond to a vector with 700 rows. The first 100 rows (corresponding to the first column of the table) would have first entry -3, the next 100 would have -2, or so on. So our parameter values form a vector X. Meanwhile, the outcomes form a vector Y with entries 0 or 1.

More generally, imagine we are studying our advertising data and, for each potential customer, we record how many times they saw our ad. We create a vector X whose entries are these numbers. Then we create another vector Y, of the same length, whose entries are either 0 or 1 depending of whether or not the customer purchased our product.

One way to think about logistic regression in this setting is that we are trying to fit a function that, given the value x_i, tries to yield the corresponding value y_i. However, instead of finding a deterministic function, as we did in linear regression, instead we try to fit a logistic function that captures the likelihood that the y-value is a 1 given the x-value. This "curve-fitting perspective" is why this is considered a regression problem.

If, as above, we think of each row of the matrix as an independent trial, then the chance that $y_i = 1$ is $p(x_i)$ and the chance that $y_i = 0$ is $1 - p(x_i)$, where $p(x)$ is given by the logistic function as in equation 1.2. The likelihood of the results we obtained is therefore:

$$L(a, b) = C \prod_{i=0}^{N-1} p(x_i)^{y_i}(1 - p(x_i))^{(1-y_i)}$$

where C is a constant and we are exploiting the trick that, since y_i is either zero or one, $1 - y_i$ is correspondingly one or zero. Thus only $p(x_i)$ or $1 - p(x_i)$ occurs in each term of the product. If we group the terms according to x_i we obtain our earlier formula for $L(a, b)$.

This expresssion yields an apparently similar formula for the log-likelihood (up to an irrelevant constant):

$$\log L(X, a, b) = \sum_{i=0}^{N-1} y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)).$$

Using vector notation, this can be further simplified, where again we drop irrelevant constants:

$$\log L(X, a, b) = Y \cdot \log p(X) + (1 - Y) \cdot \log(1 - p(X)).$$

To be absolutely concrete, in this formula, $p(X)$ is a vector

$$p(X) = [p(x_i)]_{i=0}^{N-1} = \left[\frac{1}{1+e^{-ax_i-b}}\right]_{i=0}^{N-1}$$

so its entries are functions of the unknown parameters a and b.

We might naively try to maximize this by taking the derivatives with respect to a and b and setting them to zero, but this turns out to be impractical. So we need a different approach to finding the parameters a and b which maximize this likelihood function. We will return to this problem later, but before we do so we will look at some generalizations and broader applications of the logistic model.

1.1.2 Logistic regression with multiple features

The next generalization we can consider of the logistic model is the situation where the log-odds of our event of interest depend linearly on multiple parameters. In other words, we have

$$\log \frac{p}{1-p} = m_0x_0 + m_1x_1 + \cdots + m_{k-1}x_{k-1} + b$$

where the a_i and b are constants. Under this model, notice that *the incremental effects of changes to the different parameters x_i have independent effects on the probability.* So, for example, if x_1 were the number of times our potential customer saw an online advertisement and x_2 were the number of times they saw a print advertisement, by adopting this model we are assuming that the impact of seeing more online ads is completely unrelated to the impact of seeing more print ads.

The probability is again given by a sigmoid function

$$p(x_1, \ldots, x_k) = \frac{1}{1+e^{-\sum_{i=0}^{k-1} m_ix_i+b}}$$

This model has an $N \times k$ feature matrix whose rows are the values $x_0, \ldots, x_{k-1}$ for each sample. The outcome of our experimemt is recorded in an $N \times 1$ column vector Y whose entries are 0 or 1. The likelihood function is formally equivalent to what we computed in the case of a single feature, but it will be useful to be a bit careful about vector notation.

Following the same pattern we adopted for linear regression, let X be the $N \times (k+1)$ matrix whose first k columns contain the values x_i for each sample, and whose last column is all 1. Rename the "intercept" variable as a_{k+1} and organize these parameters into a $(k+1) \times 1$ matrix M. Then

$$p(X) = \sigma(XM)$$

and our likelihood becomes

$$\log L(M) = Y \cdot \log \sigma(XM) + (1-Y) \cdot (1 - \log \sigma(XM)). \tag{1.3}$$

1.2 Finding the maximum likelihood solution by gradient descent

Given a set of features X and targets Y for a logistic model, we now want to find the values M so that the log-likelihood of the model for those paramters, given the data, is maximized. While in linear regression we could find a nice closed form solution to this problem, the presence of the non-linear function $\sigma(x)$ in the likelihood makes that impossible for logistic regression. Thus we need to use a numerical approximation. The most straightforward such method is called gradient descent. It is at the foundation of many numerical optimization algorithms, and so while we will develop it here for logistic regression we will have other opportunities to apply it and we will discuss it more thoroughly on its own later.

1.2.1 Gradient descent

Suppose that we have a function $f(x_0, \ldots, x_{k-1})$ and we wish to find its minimum value. To apply gradient descent, we choose an initial starting point $c = (c_0, \ldots, c_{k-1})$ and we iteratively adjust the values of c so that the values $f(c)$ decrease. When we can no longer do that, we've found what is at least a local minimum of f.

How should we make these adjustments? Let us remember the idea of the *directional derivative* from multivariate calculus. The directional derivative $D_v f$ measures the rate of change of f as one moves with velocity vector v from the point x and it is defined as

$$D_v f(x) = \frac{d}{dt} f(x + tv)|_{t=0}$$

From the chain rule, we can compute that

$$D_v f(x) = \sum_{i=0}^{k-1} \frac{\partial f}{\partial x_i} \frac{dx_i}{dt} = (\nabla f) \cdot v$$

where

$$\nabla f = \left[\frac{\partial f}{x_i}\right]_{i=0}^{k-1}$$

is the gradient of f. This argument yields the following result.

Proposition: Let $f(x_0, \ldots, x_{k-1})$ be a smooth function from $\mathbb{R}^k \to \mathbb{R}$. Then for every point $c = (c_0, \ldots, c_{k-1})$, if $\nabla f \neq 0$ then ∇f is a vector pointing in the direction in which f increases most rapidly, and $-\nabla f$ is a vector pointing in the direction in which f decreases most rapidly. If $\nabla f = 0$, then c is a critical point of f.

Proof: The directional derivative $D_v(f) = (\nabla f) \cdot v$ measures the rate of change of f if we travel with velocity v from a point x. To remove the dependence on the magnitude of v (since obviously f will change more quickly if we travel more quickly in a given direction),

we scale v to be a unit vector. Then the dot product giving the rate is maximized when v is parallel to ∇f.

This observation about the gradient yields the algorithm for gradient descent.

Gradient Descent Algorithm: Given a function $f : \mathbb{R}^k \to \mathbb{R}$, choose a point $c^{(0)}$, a small constant ν (called the *learning rate*) and a small constant ϵ (the *tolerance*). Then iteratively compute

$$c^{(n+1)} = c^{(n)} - \nu \nabla f(c^{(n)})$$

until $|c^{(n+1)} - c^{(n)}| < \epsilon$. Then $c^{(n+1)}$ is an (approximate) critical point of f.

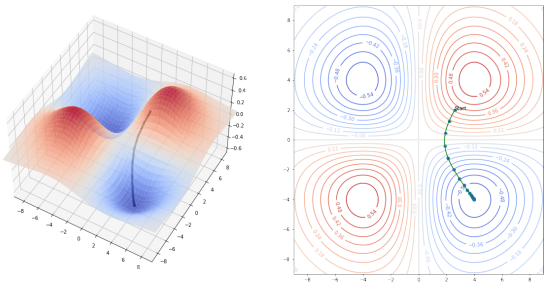

Figure 1.2: Gradient Descent

The behavior of gradient descent is illustrated in 1.2 for the function

$$f(x, y) = \frac{xy}{\sigma\sqrt{2\pi}} e^{(-x^2-y^2)/2\sigma^2}$$

where $\sigma = 4$. This function has two "upward" humps and two "downward" humps. Starting on the inside slope of one of the upward humps, gradient descent finds the bottom of an adjacent "downward" hump.

To get a little more perspective on gradient descent, consider the one-dimensional case, with $f(x) = 4x^3 - 6x^2$. This is a cubic polynomial whose graph has a local maximum and a local minimum, depicted in 1.3.

In this case the gradient is just the derivative $f'(x) = 12x^2 - 12x$ and the iteration is

$$c^{(n+1)} = c^{(n)} - 12\nu((c^{(n)})^2 - c^{(n)}).$$

Even from this simple example we can see the power and also the pitfalls of this method. Suppose we choose $x_0 = 2$, $\nu = .01$, and $\epsilon = .001$. Then the iteration yields:

Table 1.2: Gradient Descent Iterations

Step	0	1	2	3	4	5	6
x	2.0	0.8	0.896	0.952	0.979	0.991	0.997

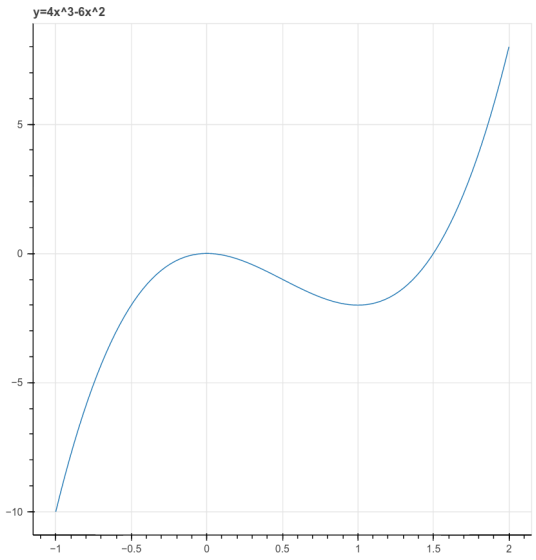

Figure 1.3: A cubic polynomial

As you can see, the points move quickly to the (local) minimum at $x = 1$.

There are two ways (at least) that things can go wrong, however. First suppose we use $x_0 = -1$, instead of $x_0 = 2$, as our first guess. Then we are on the downslope on the left side of the graph, and following the gradient quickly takes us off to $-\infty$.

Table 1.3: Gradient Descent Iterations (first failure mode)

Step	0	1	2	3	4	5
x	-1.00	-2.20	-6.42	-35.04	-792.70	-378296.27

Second, suppose we choose $x_0 = 2$, but choose a somewhat larger learning rate – say, $\nu = .1$. In this case, initially things look good, but the addition of the gradient causes an overshoot which once again takes us over the hump at $x = 0$ and off to $-\infty$ heading to the left.

Table 1.4: Gradient Descent Iterations (second failure mode)

Step	0	1	2	3	4	5	6
x	2.00	-0.11	-0.24	-0.56	-1.49	-5.42	-42.23

Based on these considerations, we see that, for general functions, *if gradient descent converges,* then it will converge to a local minimum of the function. But *it may not converge,* and even if it does, we can't conclude anything about whether we've reached a *global* minimum.

1.3 Gradient Descent and Logistic Regression

We can use gradient descent to find the maximum likelihood set of parameters for our logistic model. As we saw earlier, in 1.3, we have the log likelihood function

$$L(M) = Y \cdot \log \sigma(XM) + (1 - Y) \cdot \log(1 - \sigma(XM))$$

where Y are the target 0/1 values, X is our $N \times (k + 1)$ data matrix whose last column is all ones, and M is the $k + 1 \times 1$ column vector of unknown parameters. Since gradient descent is naturally a *minimizing* algorithm, we will minimize the function $-L(M)$.

Lemma: The logistic function $\sigma(x)$ satisfies the differential equation

$$\frac{d\sigma}{dx} = \sigma(x)(1 - \sigma(x)).$$

Proof: Since

$$\sigma(x) = \frac{1}{1 + e^{-x}},$$
$$1 - \sigma(x) = \frac{e^{-x}}{1 + e^{-x}}.$$

Then we calculate

$$\frac{d\sigma}{dx} = \left(\frac{1}{(1 + e^{-x})}\right)^2 e^{-x} = \left(\frac{1}{1 + e^{-x}}\right)\left(\frac{e^{-x}}{1 + e^{-x}}\right) = \sigma(x)(1 - \sigma(x))$$

which is what we claimed.

We apply this differential equation to compute the gradient of L.

Proposition: The gradient $-\nabla L(M)$ is given by

$$-\nabla L(M) = (\sigma(XM) - Y)^\intercal X.$$

Notice that the right side of this equation yields a $1 \times (k + 1)$ row vector. The entries of this vector are the partial derivatives with respect to the coefficients m_i for $i = 0, \ldots, k$.

Proof: This is yet another exercise in the chain rule and keeping track of indices. Let's first look at the term $Y \cdot \log \sigma(XM)$. Writing it out, we have

$$Y \cdot \log \sigma(XM) = \sum_{i=0}^{N-1} y_i \log \sigma(\sum_{j=0}^{k} x_{ij} m_j).$$

Applying $\partial/\partial m_s$ to this yields

$$\sum_{i=0}^{N-1} y_i (1 - \sigma(\sum_{j=0}^{k} x_{ij} m_j)) x_{is}$$

where we've used the chain rule and the differential equation for σ discussed above. At the same time, we can apply $\partial/\partial m_s$ to the second term $(1 - Y) \cdot \log(1 - \sigma(XM))$ and obtain

$$-\sum_{i=0}^{N-1} (1 - y_i) \sigma(\sum_{j=0}^{k} x_{ij} m_j) x_{is}.$$

The term $\sum_{i=0}^{N-1} y_i \sigma(\sum_{j=0}^{k} x_{ij} x_{is} m_j)$ cancels, yielding

$$\frac{\partial L(M)}{m_s} = -\sum_{i=0}^{N-1} (y_i - \sigma(\sum_{j=0}^{k} x_{ij} m_j)) x_{is}.$$

Looked at properly this is our desired formula:

$$-\nabla L(M) = (\sigma(XM) - Y)^\intercal X.$$