

Logistic Regression and Decision Trees

CS 221

Section 4

October 16, 2009

Today we will derive the gradient descent update rule for logistic regression using maximum likelihood and also go over an example of creating decision trees.

1 Maximum likelihood

Maximum likelihood is a general parameter estimation method. The intuition behind maximum likelihood is that we want to choose a hypothesis which makes the data as probable as possible. This will require us to make assumptions about the way that our data is generated. In general we will define probabilistic models which will describe our data generation.

1.1 Example

Suppose we are given the task of predicting the probability that a future tossed thumbtack will land with the pointy side up. To aid us in this task we are given a dataset which contains the results of a set of tosses of the thumbtack in question. How should we proceed?

Let's model the thumbtack flip in the following way: as a Bernoulli random variable, where the probability that the thumbtack lands point up is θ and the probability that it lands point down is $1 - \theta$. Let's also assume that each toss was independent, with result drawn from the same Bernoulli distribution.

Let's say that our data D contains 8 examples where the thumbtack landed point up, and 2 where it landed point down. We can now talk about the probability of this data, assuming the model parameter θ . This probability is:

$$p(D; \theta) = \theta^8 (1 - \theta)^2$$

We call this probability the **likelihood**. Our task is to choose the parameter θ that we feel best describes the probability that the thumbtack lands point up, and we have a tool to tell us the likelihood of any θ that we pick. Which θ should we pick?

The principle of **maximum likelihood** says that we should choose θ so as to make the probability of the data as high as possible. I.e. we should choose the value for θ that *maximizes* the likelihood. So, what would this be for our example? We need to solve

$$\theta = \arg \max_{\theta} \theta^8 (1 - \theta)^2$$

In general it is awkward to take derivatives of products like this, instead we can maximize the **log likelihood** $\log p(D; \theta)$. This will give us the same answer as maximizing the likelihood because the logarithm is a monotonically increasing function. We can find the maximum likelihood in our example by setting $\frac{\partial \log p(D; \theta)}{\partial \theta}$ to zero:

$$\begin{aligned} \frac{\partial}{\partial \theta} \log p(D; \theta) &= \frac{\partial}{\partial \theta} (\log (\theta^8 (1 - \theta)^2)) \\ &= \frac{\partial}{\partial \theta} (\log \theta^8 + \log (1 - \theta)^2) \\ &= \frac{\partial}{\partial \theta} (8 \log \theta + 2 \log (1 - \theta)) \\ &= \frac{8}{\theta} - \frac{2}{1 - \theta} \\ \frac{2}{1 - \theta} &= \frac{8}{\theta} \\ 2\theta &= 8 - 8\theta \\ 10\theta &= 8 \\ \theta &= 0.8 \end{aligned}$$

Thus, 0.8 is our maximum likelihood estimate for the parameter θ . In other words, the data we saw is most likely if $\theta = 0.8$. Note that this matches the intuition of the situation. If someone had asked you what the probability of a thumbtack landing point up was, and also told you that it landed point up 8 out of 10 times previously, you might have answered 80% as your best guess. Now you know in what sense that can be considered the “right” answer.

This principle is used all over in machine learning. Specifically, we will use now to derive the gradient ascent update rule for logistic regression.

2 Logistic regression

Recall from class that in logistic regression, we chose to represent a hypothesis as $g(\theta^T x)$, where

$$g(z) = \frac{1}{1 + e^{-z}}$$

This was so that our prediction would always be between 0 and 1, fitting our classification task better. First, we digress slightly to show a useful property of the derivative of the sigmoid function $g(z)$, which we will write $g'(z)$.

$$\begin{aligned}
 g'(z) &= \frac{d}{dz} \frac{1}{1 + e^{-z}} \\
 &= \frac{-1}{(1 + e^{-z})^2} \cdot \frac{d}{dz} (1 + e^{-z}) \\
 &= \frac{-1}{(1 + e^{-z})^2} (-e^{-z}) \\
 &= \frac{1}{(1 + e^{-z})^2} (e^{-z}) \\
 &= \frac{1}{1 + e^{-z}} \cdot \left(\frac{e^{-z}}{1 + e^{-z}} \right) \\
 &= \frac{1}{(1 + e^{-z})} \cdot \left(\frac{1 + e^{-z} - 1}{1 + e^{-z}} \right) \\
 &= \frac{1}{(1 + e^{-z})} \cdot \left(\frac{1 + e^{-z}}{1 + e^{-z}} - \frac{1}{1 + e^{-z}} \right) \\
 &= \frac{1}{(1 + e^{-z})} \cdot \left(1 - \frac{1}{(1 + e^{-z})} \right) \\
 g'(z) &= g(z)(1 - g(z))
 \end{aligned}$$

So, how can we use the principle of maximum likelihood to find the right θ to use in our hypothesis? We first endow our classification model with a set of probabilistic models, and then proceed as in the earlier example.

We first assume that our label y is a Bernoulli random variable. Lets assume that the value that our hypothesis returns is $P(y = 1|x; \theta)$. So

$$\begin{aligned}
 P(y = 1|x; \theta) &= h_{\theta}(x) \\
 P(y = 0|x; \theta) &= 1 - h_{\theta}(x)
 \end{aligned}$$

Note that this can be written more compactly as

$$p(y|x; \theta) = (h_{\theta}(x))^y (1 - h_{\theta}(x))^{1-y}$$

We are now in the exact same situation as in the earlier thumbtack example, except for instead of a simple real-valued θ as the parameter of the Bernoulli distribution, we now have our hypothesis $h_{\theta}(x)$.

We can write down the likelihood of the parameters as

$$\begin{aligned}
L(\theta) &= p(\vec{y}|X; \theta) \\
&= \prod_{i=1}^m p(y^{(i)}|x^{(i)}; \theta) \\
&= \prod_{i=1}^m (h_{\theta}(x^{(i)}))^{y^{(i)}} (1 - h_{\theta}(x^{(i)}))^{1-y^{(i)}}
\end{aligned}$$

As before, it will be easier to maximize the log likelihood:

$$\begin{aligned}
l(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^m (h_{\theta}(x^{(i)}))^{y^{(i)}} (1 - h_{\theta}(x^{(i)}))^{1-y^{(i)}} \\
&= \sum_{i=1}^m \log \left[(h_{\theta}(x^{(i)}))^{y^{(i)}} (1 - h_{\theta}(x^{(i)}))^{1-y^{(i)}} \right] \\
&= \sum_{i=1}^m \log \left[(h_{\theta}(x^{(i)}))^{y^{(i)}} \right] + \log \left[(1 - h_{\theta}(x^{(i)}))^{1-y^{(i)}} \right] \\
l(\theta) &= \sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))
\end{aligned}$$

Before, we were able to solve for the maximum log likelihood by setting the derivative equal to zero and solving for θ . Now, with the more complex h , we will use gradient ascent to maximize $l(\theta)$. Our updates will be $\theta := \theta + \alpha \nabla_{\theta} l(\theta)$. We will work through the derivation for a single training example (x, y) :

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} l(\theta) &= \left(y \frac{1}{g(\theta^T x)} - (1 - y) \frac{1}{1 - g(\theta^T x)} \right) \frac{\partial}{\partial \theta_j} g(\theta^T x) \\
&= \left(y \frac{1}{g(\theta^T x)} - (1 - y) \frac{1}{1 - g(\theta^T x)} \right) g(\theta^T x) (1 - g(\theta^T x)) \frac{\partial}{\partial \theta_j} \theta^T x \\
&= (y(1 - g(\theta^T x)) - (1 - y)g(\theta^T x)) x_j \\
&= (y - h_{\theta}(x)) x_j
\end{aligned}$$

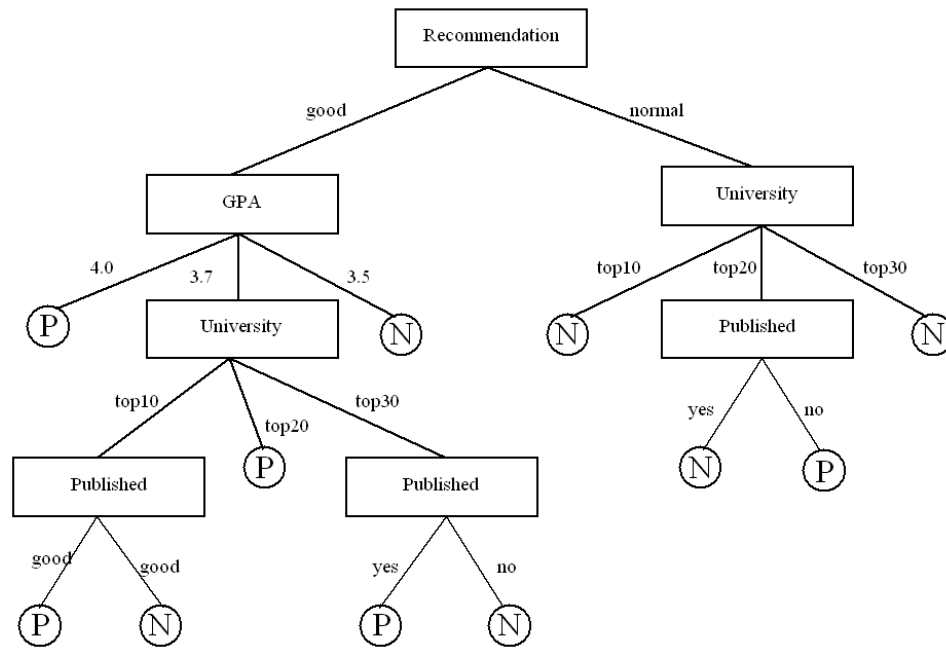
We used the fact that $g'(z) = g(z)(1 - g(z))$. We now have the gradient ascent rule of

$$\theta_j := \theta_j + \alpha \sum_{i=1}^m (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$$

So, we have shown that logistic regression is simply a special case of maximum likelihood where the target variables are Bernoulli(θ), and θ is a sigmoidal function of x .

3 Decision Trees from Data

Consider the criteria for accepting candidates to the Ph.D. program at the mythical University of St. Nordaf. Each candidate is evaluated according to four attributes: The grade point average (GPA), the quality of the undergraduate university attended, the publication record, and the strength of the recommendation letters. To simplify our example, let us discretize and limit the possible value of each attribute: Possible GPA scores are 4.0, 3.7, and 3.5; universities are categorized as top10, top20, and top30 (by top20 we mean places 11-20, and by top 30 we mean 21-30); publication record is a binary attribute - either the applicant has published previously or not; and recommendation letters are similarly binary, they are either good or normal. Finally, the candidates are classified into two classes: accepted, or P (for ‘positive’), and rejected, or N (for ‘negative’). Following is an example of one possible decision tree determining acceptance.



Applicant Pat doesn’t know this decision tree, but she does have the following data regarding twelve of last year’s applicants:

No.	Attributes				Class
	GPA	University	Published	Recommendation	
1	4.0	top10	yes	good	P
2	4.0	top10	no	good	P
3	4.0	top20	no	normal	P
4	3.7	top10	yes	good	P
5	3.7	top20	no	good	P
6	3.7	top30	yes	good	P
7	3.7	top30	no	good	N
8	3.7	top10	no	good	N
9	3.5	top20	yes	normal	N
10	3.5	top10	no	normal	N
11	3.5	top30	yes	normal	N
12	3.5	top30	no	good	N

1. Verify that the tree given above correctly categorizes these examples.

Answer: Yes the tree provided does classify the examples correctly. We can verify this by tracing a path from the root to a leaf for every example and confirming that the leaf value matches the example value.

2. Pat uses the decision tree algorithm shown in class (with the information gain computations for splitting variables) to induce the decision tree employed by St. Nordaf's officials. What tree will the algorithm come up with? Show **all** the computations involved, in addition to the tree itself.

Answer: Recall that we choose to split on the feature which gives the greatest increase in the log likelihood. This corresponded to choosing the feature that gave the greatest reduction in entropy, also known as the highest information gain.

Given a leaf l , its contribution to the log likelihood is $-m_l \mathcal{H}(p_l)$, where m_l is the number of samples associated with l , and p_l is the number associated with the leaf (we found that the optimal value for p_l is $\frac{m_{l+}}{m_{l+} + m_{l-}}$). The entropy function is defined as

$$\mathcal{H}(p) = -p \log_2 p - (1 - p) \log_2 (1 - p)$$

Suppose we split a leaf l in a tree T into $l_1 \dots l_k$ in tree T' . Then l 's contribution to T 's log likelihood, $-m_l \mathcal{H}(p_l)$, is now replaced by l_1, \dots, l_k 's contribution to T' 's log likelihood, $-\sum_{i=1}^k m_{l_i} \mathcal{H}(p_{l_i})$. Hence the increase in log likelihood / entropy reduction / information gain is

$$-\sum_{i=1}^k m_{l_i} \mathcal{H}(p_{l_i}) - (-m_l \mathcal{H}(p_l))$$

or

$$m_l \mathcal{H}(p_l) - \sum_{i=1}^k m_{l_i} \mathcal{H}(p_{l_i})$$

We will use this equation throughout our decision tree computation.

Initially, we have a single root leaf that contains all of the 12 samples, of which 6 are positive and 6 are negative:

$$\begin{aligned} m_i \mathcal{H}(p_i) &= 12\mathcal{H}\left(\frac{1}{2}\right) \\ &= 12 \end{aligned}$$

The information gains for splitting on each of the features is therefore as follows:

$$\begin{aligned} \text{Gain(GPA)} &= 12 - \left[3\mathcal{H}\left(\frac{3}{3}\right) + 5\mathcal{H}\left(\frac{3}{5}\right) + 4\mathcal{H}\left(\frac{0}{4}\right) \right] \\ &= 12 - [3(0) + 5(0.97095) + 4(0)] \\ &= 7.145 \end{aligned}$$

$$\begin{aligned} \text{Gain(University)} &= 12 - \left[5\mathcal{H}\left(\frac{3}{5}\right) + 3\mathcal{H}\left(\frac{2}{3}\right) + 4\mathcal{H}\left(\frac{1}{4}\right) \right] \\ &= 12 - [5(0.97095) + 3(0.91830) + 4(0.81128)] \\ &= 1.145 \end{aligned}$$

$$\begin{aligned} \text{Gain(Published)} &= 12 - \left[5\mathcal{H}\left(\frac{3}{5}\right) + 7\mathcal{H}\left(\frac{3}{7}\right) \right] \\ &= 12 - [5(0.97095) + 7(0.98523)] \\ &= 0.249 \end{aligned}$$

$$\begin{aligned} \text{Gain(Recommendation)} &= 12 - \left[4\mathcal{H}\left(\frac{1}{4}\right) + 8\mathcal{H}\left(\frac{5}{8}\right) \right] \\ &= 12 - [4(0.81128) + 8(0.95443)] \\ &= 1.119 \end{aligned}$$

Since *GPA* has the highest information gain, we split first on *GPA*. This divides our evidence *E* into three classes: *GPA* = 4.0, *GPA* = 3.7, and *GPA* = 3.5. For $E_{4.0}$, all instances are positive, so that branch is done. For $E_{3.5}$, all instances are negative, so again we're done. For $E_{3.7}$ there are some positive and some negative instances, so we must repeat this procedure again.

We again first compute the current contribution of this leaf to the log likelihood. The leaf contains 5 samples, of which 3 are positive and 2 are

negative:

$$\begin{aligned}m_i \mathcal{H}(p_i) &= 5\mathcal{H}\left(\frac{3}{5}\right) \\ &= 4.854\end{aligned}$$

Now we can compute the information gain for splitting on each of the three remaining attributes:

$$\begin{aligned}\text{Gain(University)} &= 4.854 - \left[2\mathcal{H}\left(\frac{1}{2}\right) + 1\mathcal{H}\left(\frac{1}{1}\right) + 2\mathcal{H}\left(\frac{1}{2}\right)\right] \\ &= 4.854 - [2(1) + 1(0) + 2(1)] \\ &= 0.854\end{aligned}$$

$$\begin{aligned}\text{Gain(Published)} &= 4.854 - \left[2\mathcal{H}\left(\frac{2}{2}\right) + 3\mathcal{H}\left(\frac{1}{3}\right)\right] \\ &= 4.854 - [2(0) + 3(0.91830)] \\ &= 2.099\end{aligned}$$

$$\begin{aligned}\text{Gain(Recommendation)} &= 4.854 - \left[5\mathcal{H}\left(\frac{3}{5}\right)\right] \\ &= 4.854 - [5(0.97095)] \\ &= 0\end{aligned}$$

Of these, *Published* has the highest gain, so we choose to split on it next. This divides these 5 cases into two categories of evidence, E_{yes} (yes, the student did publish) and E_{no} . For E_{yes} , all cases are P , so we are done.

We still need to split on E_{no} , a set containing 3 samples (1 positive and 2 negative). First we compute the amount of information in this subtree:

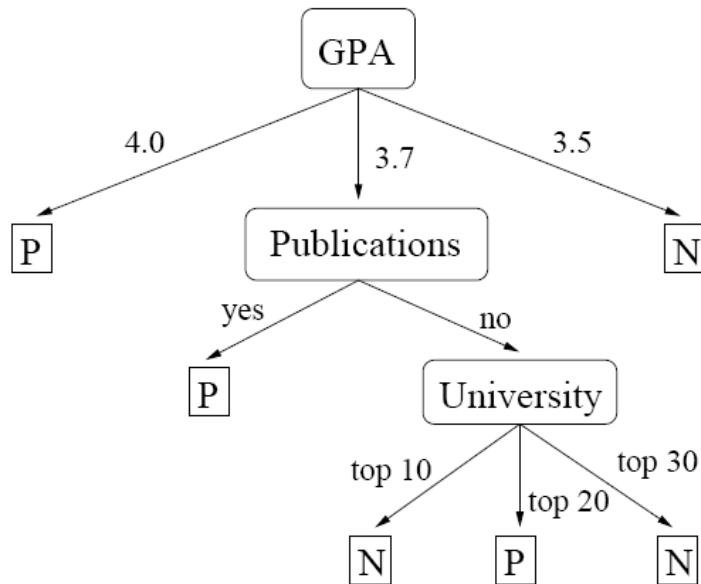
$$\begin{aligned}m_i \mathcal{H}(p_i) &= 3\mathcal{H}\left(\frac{1}{3}\right) \\ &= 2.755\end{aligned}$$

And then the information gain of the remaining attributes:

$$\begin{aligned} \text{Gain(University)} &= 2.755 - \left[1\mathcal{H}\left(\frac{0}{1}\right) + 1\mathcal{H}\left(\frac{1}{1}\right) + 1\mathcal{H}\left(\frac{0}{1}\right) \right] \\ &= 2.755 - [1(0) + 1(0) + 1(0)] \\ &= 2.755 \end{aligned}$$

$$\begin{aligned} \text{Gain(Recommendation)} &= 2.755 - \left[3\mathcal{H}\left(\frac{1}{3}\right) \right] \\ &= 2.755 - [3(0.91830)] \\ &= 0 \end{aligned}$$

So we split on *University*. It breaks the evidence down into E_{top10} , E_{top20} , and E_{top30} , each of which is completely classified, so we're done:



3. Is the tree you got in question 2 equivalent to the tree given above (i.e., do the two trees classify every application in the same way)? If the answer is yes, explain whether or not this is a coincidence. If the answer is no, give an example of a data case that will be classified differently by the two trees.

Answer: No, this tree is not equivalent to the one used by St. Nordaf's officials. for example, the case { GPA = 4.0, University = top10, Published

= yes, Recommendation = normal } is classified N by St. Nordaf's, and classified P by this tree.