CS 188 Summer 2020 Introduction to Artificial Intelligence

Final Review

Calculus, MLE and BP Solutions

1 Maximum Likelihood

A Geometric distribution is a probability distribution of the number X of Bernoulli trials needed to get one success. It depends on a parameter p, which is the probability of success for each individual Bernoulli trial. Think of it as the number of times you must flip a coin before flipping heads. The probability is given as follows:

$$P(X = k) = p(1-p)^{k-1} \tag{1}$$

p is the parameter we wish to estimate.

We observe the following samples from a Geometric distribution: $x_1 = 5$, $x_2 = 8$, $x_3 = 3$, $x_4 = 5$, $x_5 = 7$. What is the maximum likelihood estimate for p?

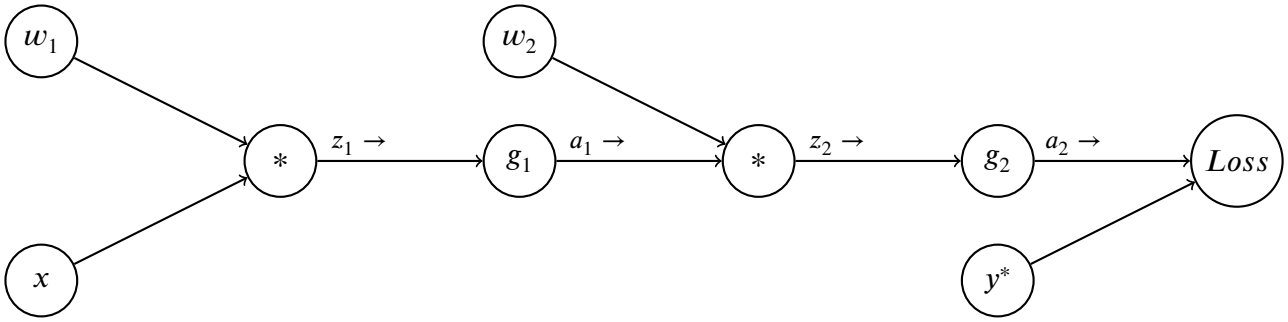
$$\begin{aligned} L(p) &= P(X = x_1)P(X = x_2)P(X = x_3)P(X = x_4)P(X = x_5) && (2)\\ &= P(X = 5)P(X = 8)P(X = 3)P(X = 5)P(X = 7) && (3)\\ &= p^5(1-p)^{23} && (4)\\ \log(L(p)) &= 5\log(p) + 23\log(1-p) && (5)\\ & && (6) \end{aligned}$$

We must maximize the log-likelihood of p, so we will take the derivative, and set it to 0.

$$\begin{aligned} 0 &= \frac{5}{p} - \frac{23}{1-p} && (7)\\ p &= 5/28 && (8) \end{aligned}$$

2 Neural Nets

Consider the following computation graph for a simple neural network for binary classification. Here x is a single real-valued input feature with an associated class y^* (0 or 1). There are two weight parameters w_1 and w_2, and non-linearity functions g_1 and g_2 (to be defined later, below). The network will output a value a_2 between 0 and 1, representing the probability of being in class 1. We will be using a loss function $Loss$ (to be defined later, below), to compare the prediction a_2 with the true class y^*.



1. Perform the forward pass on this network, writing the output values for each node z_1, a_1, z_2 and a_2 in terms of the node's input values:

$$z_1 = x * w_1$$
$$a_1 = g_1(z_1)$$
$$z_2 = a_1 * w_2$$
$$a_2 = g_2(z_2)$$

2. Compute the loss $Loss(a_2, y^*)$ in terms of the input x, weights w_i, and activation functions g_i:
   Recursively substituting the values computed above, we have:

$$Loss(a_2, y^*) = Loss(g_2(w_2 * g_1(w_1 * x)), y^*)$$

3. Now we will work through parts of the backward pass, incrementally. Use the chain rule to derive $\frac{\partial Loss}{\partial w_2}$. Write your expression as a product of partial derivatives at each node: i.e. the partial derivative of the node's output with respect to its inputs. (Hint: the series of expressions you wrote in part 1 will be helpful; you may use any of those variables.)

$$\frac{\partial Loss}{\partial w_2} = \frac{\partial Loss}{\partial a_2} \frac{\partial a_2}{\partial z_2} \frac{\partial z_2}{\partial w_2}$$

4. Suppose the loss function is quadratic, $Loss(a_2, y^*) = \frac{1}{2}(a_2 - y^*)^2$, and g_1 and g_2 are both sigmoid functions $g(z) = \frac{1}{1+e^{-z}}$ (note: it's typically better to use a different type of loss, *cross-entropy*, for classification problems, but we'll use this to make the math easier).

   Using the chain rule from Part 3, and the fact that $\frac{\partial g(z)}{\partial z} = g(z)(1 - g(z))$ for the sigmoid function, write $\frac{\partial Loss}{\partial w_2}$ in terms of the values from the forward pass, y^*, a_1, and a_2:

   First we'll compute the partial derivatives at each node:

$$\frac{\partial Loss}{\partial a_2} = (a_2 - y^*)$$
$$\frac{\partial a_2}{\partial z_2} = \frac{\partial g_2(z_2)}{\partial z_2} = g_2(z_2)(1 - g_2(z_2)) = a_2(1 - a_2)$$
$$\frac{\partial z_2}{\partial w_2} = a_1$$

   Now we can plug into the chain rule from part 3:

$$\frac{\partial Loss}{\partial w_2} = \frac{\partial Loss}{\partial a_2}\frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial w_2}$$
$$= (a_2 - y^*) * a_2(1 - a_2) * a_1$$

5. Now use the chain rule to derive $\frac{\partial Loss}{\partial w_1}$ as a product of partial derivatives at each node used in the chain rule:

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial a_2}\frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_1}$$

6. Finally, write $\frac{\partial Loss}{\partial w_1}$ in terms of x, y^*, w_i, a_i, z_i:

   The partial derivatives at each node (in addition to the ones we computed in Part 4) are:

$$\frac{\partial z_2}{\partial a_1} = w_2$$
$$\frac{\partial a_1}{\partial z_1} = \frac{\partial g_1(z_1)}{\partial z_1} = g_1(z_1)(1 - g_1(z_1)) = a_1(1 - a_1)$$
$$\frac{\partial z_1}{\partial a_1} = x$$

   Plugging into the chain rule from Part 5 gives:

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial a_2}\frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_1}$$
$$= (a_2 - y^*) * a_2(1 - a_2) * w_2 * a_1(1 - a_1) * x$$

7. What is the gradient descent update for w_1 with step-size α in terms of the values computed above?

$$w_1 \leftarrow w_1 - \alpha \cdot (a_2 - y^*) * a_2(1 - a_2) * w_2 * a_1(1 - a_1) * x$$

8. Now suppose that the neural network is modified to re-use a single weight w instead of two separate weights w_1 and w_2. What is the new update rule for this single weight w?

   Reusing results from parts (4) and (6) and substituting in w for w_1, w_2, so

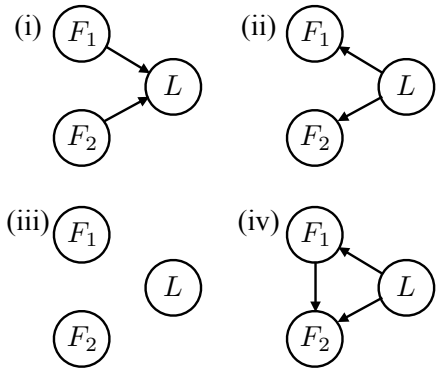
$$\frac{\partial Loss}{\partial w} = \frac{\partial Loss}{\partial w_1} + \frac{\partial Loss}{\partial w_2}$$

$$= (a_2 - y^*) * a_2(1 - a_2) * w * a_1(1 - a_1) * x + (a_2 - y^*) * a_2(1 - a_2) * a_1$$

   Thus, the new update rule for w is

$$w \leftarrow w - \alpha \cdot (a_2 - y^*) * a_2(1 - a_2) * w * a_1(1 - a_1) * x + (a_2 - y^*) * a_2(1 - a_2) * a_1$$

Q3. ML: Maximum Likelihood



Training Data

$(L = 1, \quad F_1 = 1, \quad F_2 = 1)$
$(L = 1, \quad F_1 = 1, \quad F_2 = 1)$
$(L = 0, \quad F_1 = 1, \quad F_2 = 1)$
$(L = 1, \quad F_1 = 0, \quad F_2 = 0)$
$(L = 0, \quad F_1 = 0, \quad F_2 = 0)$
$(L = 0, \quad F_1 = 0, \quad F_2 = 0)$
$(L = 0, \quad F_1 = 0, \quad F_2 = 1)$

You've decided to use a model-based approach to classification of text documents. Your goal is to build a classifier that can determine whether or not a document is about cats. You're taking a minimalist approach and you're only characterizing the input documents in terms of two binary features: F_1 and F_2. Both of these features have domain $\{0, 1\}$. The thing you're trying to predict is the label, L, which is also binary valued. When $L = 1$, the document is about cats. When $L = 0$, the document is not.

The particular meaning of the two features F_1 and F_2 is not important for your current purposes. You are only trying to decide on a particular Bayes' net structure for your classifier. You've got your hands on some training data (shown above) and you're trying to figure out which of several potential Bayes' nets (also shown above) might yield a decent classifier when trained on that training data.

(a) Which of the Bayes' nets, once learned from the training data with maximum likelihood estimation, would assign non-zero probability to the following query: $P(L = 1|F_1 = 0, F_2 = 0)$? Fill in all that apply.

● (i) ● (ii) ● (iii) ● (iv)

(b) Which of the Bayes' nets, once learned from the training data with maximum likelihood estimation, would assign non-zero probability to the following query: $P(L = 1|F_1 = 0, F_2 = 1)$? Fill in all that apply.

○ (i) ● (ii) ● (iii) ○ (iv)

(c) Which of the Bayes' nets, once learned from the training data with Laplace smoothing using $k = 1$, would assign non-zero probability to the following query: $P(L = 1|F_1 = 0, F_2 = 1)$? Fill in all that apply.

● (i) ● (ii) ● (iii) ● (iv)

(d) What probability does Bayes' net (i), once learned from the training data with Laplace smoothing using $k = 1$, assign to the query $P(L = 1|F_1 = 0, F_2 = 1)$?

$\frac{1}{3}$

(e) As $k \to \infty$ (the constant used for Laplace smoothing), what does the probability that Bayes' net (i) assigns to the query $P(L = 1|F_1 = 0, F_2 = 1)$ converge to?

$\frac{1}{2}$