Noah Arthurs
CS 109

Section #6
August 11, 2019

Section 6: Solution

Formula reference for Logistic Regression:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$
$$LL(\theta) = \sum_{i=0}^{n} y^{(i)} \log \sigma(\theta^T \mathbf{x}^{(i)}) + (1-y^{(i)}) \log\left[1-\sigma(\theta^T \mathbf{x}^{(i)})\right]$$
$$\frac{\partial LL(\theta)}{\partial \theta_j} = \sum_{i=0}^{n} \left[y^{(i)} - \sigma(\theta^T \mathbf{x}^{(i)})\right] x_j^{(i)}$$

1. **Vision Test**

   You decide that the vision tests given by eye doctors would be more precise if we used an approach inspired by logistic regression. In a vision test a user looks at a letter with a particular font size and either correctly guesses the letter or incorrectly guesses the letter.

   You assume that the probability that a particular patient is able to guess a letter correctly is:

   $$p = \sigma(\theta + f)$$

   Where θ is the user's vision score and f is the font size of the letter.

   Explain how you could estimate a user's vision score (θ) based on their 20 responses $(f^{(1)}, y^{(1)}) \ldots (f^{(20)}, y^{(20)})$, where $y^{(i)}$ is an indicator variable for whether the user correctly identified the ith letter and $f^{(i)}$ is the font size of the ith letter. Solve for any and all partial derivatives required by the approach you describe in your answer.

We are going to solve this problem by finding the MLE estimate of θ. To find the MLE estimate, we are going to find the argmax of the log likelihood function. To calculate argmax we are going to use gradient ascent, which requires that we know the partial derivative of the log likelihood function with respect to theta.

First we write the log likelihood:

$$L(\theta) = \prod_{i=1}^{20} p^{y^{(i)}} (1-p)^{[1-y^{(i)}]}$$
$$LL(\theta) = \sum_{i=1}^{20} (y^{(i)} \log(p) + (1-y^{(i)}) \log(1-p))$$

Then we find the derivative of log likelihood with respect to θ for one datapoint:

$$\frac{\partial LL}{\partial \theta} = \frac{\partial LL}{\partial p} \cdot \frac{\partial p}{\partial \theta}$$

We can calculate both the smaller partial derivatives independently:

$$\frac{\partial LL}{\partial p} = \frac{y^{(i)}}{p} - \frac{1 - y^{(i)}}{1 - p} \text{ and } \frac{\partial p}{\partial \theta} = p[1 - p]$$

Putting it all together for one letter:

$$\begin{aligned} \frac{\partial LL}{\partial \theta} &= \frac{\partial LL}{\partial p} \cdot \frac{\partial p}{\partial \theta} = [\frac{y^{(i)}}{p} - \frac{1 - y^{(i)}}{1 - p}]p[1 - p] \\ &= y^{(i)}(1 - p) - p(1 - y^{(i)}) = y^{(i)} - p = y^{(i)} - \sigma(\theta - f) \end{aligned}$$

For all twenty examples:

$$\frac{\partial LL}{\partial \theta} = \sum_{i=1}^{20} y^{(i)} - \sigma(\theta + f^{(i)})$$

2. **Multiclass Bayes**

In this problem we are going to explore how to write Naive Bayes for multiple output classes. We want to predict a single output variable Y which represents how a user feels about a book. Unlike in your homework, the output variable Y can take on one of the *four* values in the set {Like, Love, Haha, Sad}. We will base our predictions off of three binary feature variables $X_1, X_2,$ and X_3 which are indicators of the user's taste. All values $X_i \in \{0, 1\}$.

We have access to a dataset with 10,000 users. Each user in the dataset has a value for X_1, X_2, X_3 and Y. You can use a special query method **count** that returns the number of users in the dataset with the given *equality* constraints (and only equality constraints). Here are some example usages of **count**:

count($X_1 = 1, Y =$ Haha)	returns the number of users where $X_1 = 1$ and $Y =$ Haha.
count($Y =$ Love)	returns the number of users where $Y =$ Love.
count($X_1 = 0, X_3 = 0$)	returns the number of users where $X_1 = 0$, and $X_3 = 0$.

You are given a new user with $X_1 = 1, X_2 = 1, X_3 = 0$. What is the best prediction for how the user will feel about the book (Y)? You may leave your answer in terms of an argmax function. You should explain how you would calculate all probabilities used in your expression. Use **Laplace estimation** when calculating probabilities.

We can make the Naive Bayes assumption of independence and simplify argmax of $P(Y|\mathbf{X})$ to get an expression for $\hat{Y}$, the predicted output value, and evaluate it using the provided **count** function.

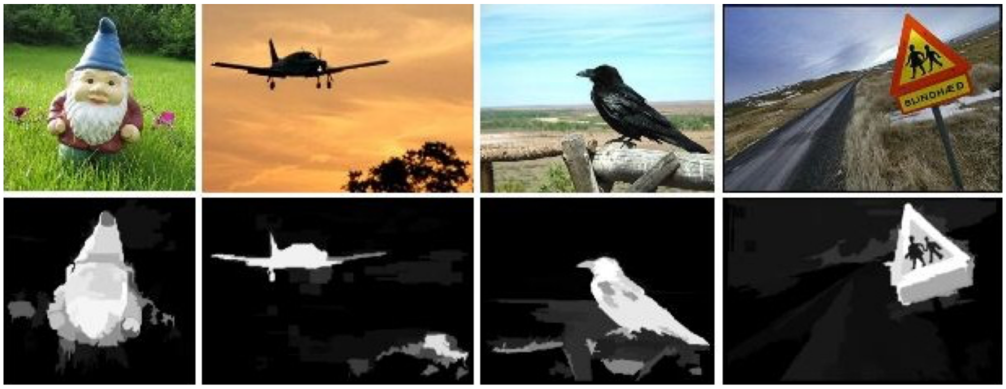
$$\begin{aligned}
\hat{Y} &= \arg\max_y \frac{P(X_1 = 1, X_2 = 1, X_3 = 0|Y = y)P(Y = y)}{P(X_1 = 1, X_2 = 1, X_3 = 0)} \\
&= \arg\max_y \; P(X_1 = 1, X_2 = 1, X_3 = 0|Y = y)P(Y = y) \\
&= \arg\max_y \; P(X_1 = 1|Y = y)P(X_2 = 1|Y = y)P(X_3 = 0|Y = y)P(Y = y), \text{ where:}
\end{aligned}$$

$$\begin{aligned}
P(X_1 = 1|Y = y) &= [\mathbf{count}(X_1 = 1, Y = y) + 1]/\mathbf{count}(Y = y) + 2 \\
P(X_2 = 1|Y = y) &= [\mathbf{count}(X_2 = 1, Y = y) + 1]/\mathbf{count}(Y = y) + 2 \\
P(X_3 = 1|Y = y) &= [\mathbf{count}(X_3 = 1, Y = y) + 1]/\mathbf{count}(Y = y) + 2 \\
P(X_1 = 0|Y = y) &= [\mathbf{count}(X_1 = 0, Y = y) + 1]/\mathbf{count}(Y = y) + 2 \\
P(X_2 = 0|Y = y) &= [\mathbf{count}(X_2 = 0, Y = y) + 1]/\mathbf{count}(Y = y) + 2 \\
P(X_3 = 0|Y = y) &= [\mathbf{count}(X_3 = 0, Y = y) + 1]/\mathbf{count}(Y = y) + 2
\end{aligned}$$

you don't need to use MAP to estimate $P(Y)$: $P(Y = y) = \mathbf{count}(Y = y)/10,000$

3. **The Most Important Features**

Let's explore saliency, a measure of how important a feature is for classification. We define the saliency of the ith input feature for a given example $(\mathbf{x}, y)$ to be the absolute value of the partial derivative of the log likelihood of the sample prediction, with respect to that input feature $\left|\frac{\partial LL}{\partial x_i}\right|$. In the images below, we show both input images and the corresponding saliency of the input features (in this case, input features are pixels):



First consider a trained logistic regression classifier with weights θ. Like the logistic regression classifier that you wrote in your homework it predicts binary class labels. In this question we allow the values of $\mathbf{x}$ to be real numbers, which doesn't change the algorithm (neither training nor testing).

a. What is the Log Likelihood of a single training example $(\mathbf{x}, y)$ for a logistic regression classifier?

$$LL(\theta) = y \cdot \log \sigma\left(\theta^T \cdot \mathbf{x}\right) + \left(1 - y\right)\log\left[1 - \sigma\left(\theta^T \cdot \mathbf{x}\right)\right]$$

b. Calculate is the saliency of a single feature (x_i) in a training example $(\mathbf{x}, y)$.

We can calculate the saliency for a single feature as follows.

$$\begin{aligned}
LL(\theta) &= y \log z + \left(1 - y\right) \log \left(1 - z\right) && \text{where } z = \sigma\left(\theta^T \cdot \mathbf{x}\right) \\
\frac{\partial LL}{\partial x_i} &= \frac{\partial LL}{\partial z} \cdot \frac{\partial z}{\partial x_i} && \text{chain rule} \\
&= \left(\frac{y}{z} - \frac{1 - y}{1 - z}\right) \cdot \left(z(1 - z)\theta_i\right) && \text{partial derivatives} \\
\text{saliency} &= \left| \left(\frac{y}{z} - \frac{1 - y}{1 - z}\right) z(1 - z)\theta_i \right|
\end{aligned}$$

c. Show that the ratio of saliency for features i and j is the ratio of the absolute value of their weights $\frac{|\theta_i|}{|\theta_j|}$.

We can take the ratio as follows using our expression above.

$$\text{saliency for feature } i, S_i = \left| \left(\frac{y}{z} - \frac{1 - y}{1 - z}\right) z(1 - z)\theta_i \right| \text{, and same for } S_j$$

$$\frac{S_i}{S_j} = \frac{\left| \left(\frac{y}{z} - \frac{1-y}{1-z}\right) z(1 - z)\theta_i \right|}{\left| \left(\frac{y}{z} - \frac{1-y}{1-z}\right) z(1 - z)\theta_i \right|} = \frac{S_i}{S_j} = \frac{\left|\theta_i\right|}{\left|\theta_j\right|} \text{ by elimination}$$