Assignment 3: Classification, Logistic Regression, and Gradient Descent

Machine Learning

Fall 2019

Learning Objectives

- Learn about the framing of the classification problem in machine learning.
- Learn about the logistic regression algorithm.
- Learn about gradient descent for optimization.
- Some C&E topic.

Prior Knowledge Utilized

- Supervised learning problem framing.
- Training / testing splits.

Recall: Supervised Learning Problem Setup

We are given a training set, $(\mathbf{x_1}, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$ where each $\mathbf{x_i}$ represents an element of an input space (e.g., a d-dimensional feature vector) and each y_i represents an element of an output space (e.g., a scalar target value). Our goal is to determine a function $\hat{f}$ that maps from the input space to the output space.

We assume there is a loss function, ℓ, that determines the amount of loss that a particular prediction $\hat{y}_i$ incurs due to a mismatch with the actual output y_i. The best possible model, $\hat{f}^\star$, is the one that minimizes these losses over the training set. This notion can be expressed with the following equation.

$$\hat{f}^\star = \arg\min_{\hat{f}} \sum_{i=1}^{n} \ell\left(\hat{f}(\mathbf{x_i}), y_i\right) \tag{1}$$

1 *The Classification Problem*

So far in this class we've looked at supervised learning problems where the responses y_i are continuous valued and the loss function was quadratic ($\ell(y, \hat{y}) = (y - \hat{y})^2$). This setting is called the regression setting. There are many times, however, where it is unnatural to frame a problem as a regression. A classification problem is one where the y_i's take on a discrete set of values. For instance, you might imagine taking an image of a person and predicting their identity. The identity could be thought of as being from some discrete set. A special case of the classification problem is binary classification when y_i is either 0 or 1 (e.g., a Paul versus Sam detector).

In this assignment will formalize the binary classification problem and see a very useful algorithm for solving it called *logistic regression*. You will also see that the logistic regression algorithm is a very natural extension of linear regression. Our plan for getting there is going to be pretty similar to what we did for linear regression:

- Build some mathematical foundations
- Introduce logistic regression from a top-down perspective
- Learn about logistic regression from a bottom-up perspective

2 *Minimizing the misclassification rate*

To begin our framing of the binary classification problem, let's look at one possible formalization where we are given a training set, $(\mathbf{x_1}, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$, where each $\mathbf{x_i}$ is an element of the input space (e.g., a vector) and each y_i is a binary number (either 1 or 0). Further, we define the loss function $\ell(y, \hat{y}) = \mathbb{I}[y \neq \hat{y}]$ (the funny looking symbol $\mathbb{I}$ is the indicator function that takes on value 1 when the condition inside is true and 0 otherwise. Given these choices, the supervised learning problem becomes.

$$\hat{f}^\star = \arg\min_{\hat{f}} \sum_{i=1}^{n} \mathbb{I}\left[\hat{f}(\mathbf{x_i}) \neq, y_i\right] \tag{2}$$

Exercise 1

Convert Equation 2 to English to make sure you understand it.

☆ Solution

The equation says that $\hat{f}^\star$ is the function $\hat{f}$ that minimizes the number of mistakes that the function makes on the training set.

Intuitively, such a framing makes perfect sense. We should choose the model that makes the fewest mistakes on the training set. It turns out, however, that it is not a particularly easy function to work with mathematically. For one thing it is a bit all or nothing. Either we are completely right or completely wrong. It also turns out to be difficult to minimize for many common classes of functions, $\hat{f}$. It turns out that we can create a much more natural loss function by thinking about the problem in terms of probabilities.

3 *Probability and the log loss*

Imagine that instead of our model, $\hat{f}$, spitting out either 0 or 1, it outputted a probability that the input x_i had an output of 1. In this way the classifier could indicate to us its degree of certainty regarding its prediction. We haven't formally defined probability in this class, and we won't do so here. For this case, we just need to keep a few things in mind about probabilities.

- Probabilities give the chance that some event occurs (probability of 0 means that something will definitely not occur and probability of 1 means it definitely will occur.
- A probability, q, must be between 0 and 1 ($0 \leq q \leq 1$).
- If the probability that event occurs is q, then the probability that it doesn't occur is $1 - q$.

3.1 Log-loss

One of the components of our supervised learning problem framing is the loss function ℓ. Recall that this function takes as input the true output value, y, and a predicted output value, $\hat{y}$, and returns the loss that the model incurs for any potential mismatch between the values. When we were working with linear regression, we sought to minimize the sum of squared errors and consequently used the loss function $\ell(y, \hat{y}) = (y - \hat{y})^2$.

3.2 Logistic function

The logistic function turns out to be very useful for modeling the probability that some event occurs. TODO.

Exercise 2

In this exercise you will be working to better understand some of the properties of the logistic function. Remember, the logistic function, σ, is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad . \tag{3}$$

(a) Do some thought exercises on the logistic function. Limiting cases, etc. TODO.

(b) Show that $\sigma(-x) = 1 - \sigma(x)$.

☆ Solution

$$\begin{align}
\sigma(-x) &= \frac{1}{1 + e^{x}} \tag{4}\\
&= \frac{e^{-x}}{e^{-x} + 1} \quad \text{multiply by top and bottom by } e^{-x} \tag{5}\\
\sigma(-x) - 1 &= \frac{e^{-x}}{e^{-x} + 1} - \frac{1 + e^{-x}}{1 + e^{-x}} \quad \text{subtract } -1 \text{ on both sides} \tag{6}\\
&= \frac{-1}{1 + e^{-x}} \tag{7}\\
&= -\sigma(x) \tag{8}\\
\sigma(-x) &= 1 - \sigma(x) \tag{9}
\end{align}$$

(c) Show that the derivative of the logistic function $\frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$

☆ Solution

Two solutions for the price of 1!

Solution 1:

$$\begin{align}
\frac{d}{dx}\sigma(x) &= -e^{-x}\sigma(x)^2 && \text{apply quotient rule} \tag{10}\\
&= \sigma(x)\left(\frac{-e^{-x}}{1+e^{-x}}\right) && \text{expand out one of the } \sigma(x)\text{'s} \tag{11}\\
&= \sigma(x)\left(\frac{-1}{e^{x}+1}\right) && \text{multiply top and bottom by } e^{x} \tag{12}\\
&= \sigma(x)(-\sigma(-x)) && \text{substitute for } \sigma(-x) \tag{13}\\
&= \sigma(x)(\sigma(x)-1) && \text{apply } \sigma(-x) = 1-\sigma(x) \tag{14}
\end{align}$$

Solution 2:

$$\begin{align}
\frac{d}{dx}\sigma(x) &= \frac{-e^{-x}}{(1+e^{-x})^2} && \text{apply quotient rule} \tag{15}\\
&= \frac{-e^{-x}}{1+2e^{-x}+e^{-2x}} && \text{expand the bottom} \tag{16}\\
&= \frac{-1}{e^{x}+2+e^{-x}} && \text{multiply top and bottom by } e^{x} \tag{17}\\
&= \frac{-1}{(1+e^{x})(1+e^{-x})} && \text{factor} \tag{18}\\
&= -\sigma(x)\sigma(-x) && \text{decompose using definition of } \sigma(x) \tag{19}\\
&= -\sigma(x)(1-\sigma(x)) && \text{apply } \sigma(-x) = 1-\sigma(x) \tag{20}\\
&= \sigma(x)(\sigma(x)-1) && \text{distribute the } -1 \tag{21}
\end{align}$$

(d) The log odds of an event occurring is defined as

$$\ln\left(\frac{p(\text{event occurs})}{p(\text{event does not occur})}\right) = \ln\left(\frac{p(\text{event occurs})}{1-p(\text{event does occur})}\right)\ . \tag{22}$$

If we assume that $p(\text{event occurs}) = \sigma(x)$, show that the log odds of the event occurring is equal to x.

☆ **Solution**

$$
\begin{aligned}
\ln\left(\frac{p(\text{event occurs})}{p(\text{event does not occur})}\right) &= \ln\left(\frac{\sigma(x)}{1-\sigma(x)}\right) && (23)\\
&= \ln\left(\frac{\sigma(x)}{\sigma(-x)}\right) && (24)\\
&= \ln\left(\frac{1+e^{x}}{1+e^{-x}}\right) && (25)\\
&= \ln\left(e^{x}\frac{1+e^{x}}{e^{x}(1+e^{-x})}\right) && (26)\\
&= x+\ln\left(\frac{1+e^{x}}{e^{x}+1}\right) && (27)\\
&= x && (28)
\end{aligned}
$$

4 Top-down View of Logistic Regression

5 Gradient Descent

5.1 Chain Rule for Gradients

5.2 Visualization

6 Algorithm Derivation

Todo: this is easier with the identities of the derivative of a logistic function.

$$
\begin{aligned}
\mathbf{w}^{\star} &= \arg\min_{\mathbf{w}} e(\mathbf{w}) && (29)\\
e(\mathbf{w}) &= \sum_{i=1}^{n} y_i \log\frac{1}{1+e^{-\mathbf{w}^{\top}\mathbf{x_i}}} + (1-y_i)\log\frac{1}{1+e^{\mathbf{w}^{\top}\mathbf{x_i}}} && (30)\\
&= \arg\min_{\mathbf{w}} \sum_{i=1}^{n} -y_i\log\left(1+e^{-\mathbf{w}^{\top}\mathbf{x_i}}\right) - (1-y_i)\log\left(1+e^{\mathbf{w}^{\top}\mathbf{x_i}}\right) && (31)\\
\nabla e(\mathbf{w}) &= \sum_{i=1}^{n} \frac{y_i\mathbf{x_i}}{1+e^{-\mathbf{w}^{\top}\mathbf{x_i}}} - \frac{(1-y_i)\,\mathbf{x_i}}{1+e^{\mathbf{w}^{\top}\mathbf{x_i}}} && (32)\\
&= \sum_{i=1}^{n} \mathbf{x_i}\left(\frac{y_i}{1+e^{-\mathbf{w}^{\top}\mathbf{x_i}}} - \frac{(1-y_i)}{1+e^{\mathbf{w}^{\top}\mathbf{x_i}}}\right) && (33)
\end{aligned}
$$