Derivation of Logistic Regression

Author: Sami Abu-El-Haija (samihaija@umich.edu)

We derive, step-by-step, the Logistic Regression Algorithm, using Maximum Likelihood Estimation (MLE). Logistic Regression is used for *binary classification* tasks (i.e. the class [*a.k.a* label] is 0 or 1). Logistic Regression processes a dataset $\mathcal{D} = \{(\mathbf{x}^{(1)}, t^{(1)}), ..., (\mathbf{x}^{(N)}, t^{(N)})\}$, where $t^{(i)} \in \{0, 1\}$ and the feature vector of the i-th example is $\phi(\mathbf{x}^{(i)}) \in \mathbb{R}^M$.

Logistic Regression forms a probabilistic model. It estimates probability distributions of the two classes ($p(t = 1|\mathbf{x}; \mathbf{w})$ and $p(t = 0|\mathbf{x}; \mathbf{w})$). Logistic Regression *fits* its parameters $\mathbf{w} \in \mathbb{R}^M$ to the training data by Maximum Likelihood Estimation (i.e. finds the $\mathbf{w}$ that maximize the probability of the training data).

We introduce the model, give some intuitions to its mechanics in the context of spam classification, then derive how to find the optimum $\mathbf{w}$ given training data.

1 The Logistic Model

Given features for an example, $\phi(\mathbf{x})$, logistic regression models the probability of this example belonging to the class 1 as:

$$p(t = 1|\mathbf{x}; \mathbf{w}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}))$$

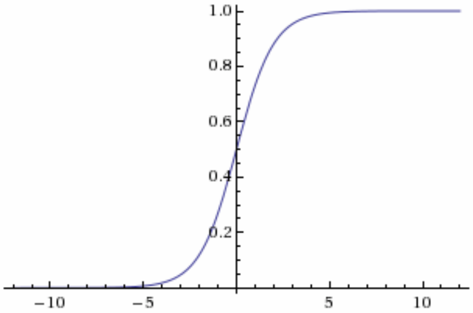
And defines the probability of the example belonging to the class 0 as:

$$\begin{aligned} p(t = 0|\mathbf{x}; \mathbf{w}) &= 1 - p(t = 1|\mathbf{x}; \mathbf{w}) \\ &= 1 - \sigma(\mathbf{w}^T \phi(\mathbf{x})) \end{aligned}$$

Where $\sigma(a)$ is the *sigmoid* function. It is defined as:

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

The sigmoid function looks like:



It can be shown that the derivative of the sigmoid function is (please verify that yourself):

$$\frac{\partial \sigma(a)}{\partial a} = \sigma(a)(1 - \sigma(a))$$

This derivative will be useful later.

2 Intuition

The task of Logistic regression is to find the $\mathbf{w}$ that works well on the training data. But intuitively, what is a *good* $\mathbf{w}$? Let's take the example of spam vs non-spam email classification problem. In this task, we are given N emails, each has a label $\in \{\texttt{NonSpam}, \texttt{Spam}\}$, or, $\{0, 1\}$. For illustrative purposes, let's assume that the terms *viagra* and *condom* appear a lot in spam emails, but on the other hand, the terms *school* and *picnic* appear a lot in non-spam emails. Let's assume that for each email $\mathbf{x}$, our feature-mapping function, $\phi(.)$, produces a vector of length $10,000$. Here we assume that the English vocabulary contains $10,000$ words. We define the feature mapping to produce a binary vector: $\phi(\mathbf{x})$ will have a 1 on the j-th index if the j-th English term appears in the email $\mathbf{x}$. i.e. $\phi_j(\mathbf{x}) = I(\mathbf{x} \text{ contains } j\text{-th word})$.

Visually, for email $\mathbf{x} =$ *"You should buy a viagra"*, its feature vector is, according to our described $\phi(.)$:

$$\phi(\mathbf{x}) = \left[\begin{array}{c} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ \vdots \\ 1 \\ \vdots \\ \vdots \end{array}\right] \begin{array}{l} \texttt{a} \\ \texttt{aardvark} \\ \texttt{aback} \\ \vdots \\ \texttt{buxom} \\ \texttt{buy} \\ \texttt{buzz} \\ \vdots \\ \texttt{shot} \\ \texttt{should} \\ \texttt{shop} \\ \vdots \\ \vdots \\ \texttt{viagra} \\ \vdots \\ \vdots \end{array}$$

Disclaimer: We don't intend to dive deep into computational linguistics and information retrieval, as they deserve a book on their own. Although binary vectors do reasonably well in text classification, they are much outperformed by TF-IDF vectors and more complex feature representations.

Logistic regression learns the parameter $\mathbf{w} \in \mathbb{R}^M$ (where $M = 10,000$ in our example). The optimal $\mathbf{w}$ should have positive values for terms coming from the positive class (*spam words*, like *viagra* and *condom*), and has negative values for 'non-spam words' (like *school* and *picnic*). Visually, logistic regression should learn $\mathbf{w}$ that looks like:

$$\mathbf{w} = \begin{array}{cl} \left[\begin{array}{c} \text{-0.2} \\ \text{-2.8} \\ \vdots \\ 3.1 \\ \vdots \\ \text{-1.2} \\ \vdots \\ \text{-0.9} \\ \vdots \\ 12.7 \\ \vdots \end{array}\right] & \begin{array}{l} \texttt{a} \\ \texttt{aardvark} \\ \vdots \\ \texttt{condom} \\ \vdots \\ \texttt{picnic} \\ \vdots \\ \texttt{school} \\ \vdots \\ \texttt{viagra} \\ \vdots \end{array} \end{array}$$

Having a larger positive value means that the term was strongly more observed in the positive class (spam) in the training set. Similarily, more negative means the word is more observed in the negative class (non-spam).

Given some test email $\mathbf{x}$, Logistic regression classifies the email by computing $p(t = 1|\mathbf{x}; \mathbf{w}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}))$ and $p(t = 0|\mathbf{x}; \mathbf{w}) = 1 - \sigma(\mathbf{w}^T \phi(\mathbf{x}))$ then classifying to the class that has a larger probability measure. Intuitively, if the $\mathbf{x}$ contains a lot of spam words, then the argument of the sigmoid function $(\mathbf{w}^T \phi(\mathbf{x}))$ will be positive. Look at the shape of the sigmoid function above. If $\mathbf{w}^T \phi\,(\mathbf{x}) > 0$, then $\sigma(\mathbf{w}^T \phi(\mathbf{x})) > 0.5$, which means $p(t = 1|\mathbf{x}; \mathbf{w}) > p(t = 0|\mathbf{x}; \mathbf{w})$, causing the email to be classified as spam.

3 Maximum Likelihood Estimation

The likelihood function $L(\mathbf{w})$ is defined as the probability that the current $\mathbf{w}$ assigns to the training set:

$$L(\mathbf{w}) = \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)}; \mathbf{w})$$

However, we have two separate terms for $p(t = 1|\mathbf{x}; \mathbf{w})$ and $p(t = 0|\mathbf{x}; \mathbf{w})$. Nonetheless, it is possible to combine those two terms into one like:

$$p(t^{(i)}|\mathbf{x}^{(i)}; \mathbf{w}) = p(t = 1|\mathbf{x}^{(i)}; \mathbf{w})^{t^{(i)}} p(t = 0|\mathbf{x}^{(i)}; \mathbf{w})^{1-t^{(i)}}$$

The above trick is used a lot in Machine Learning. It is easy to see that only one of the terms will be *active* depending on the value of t. For $t \in \{0, 1\}$, one if the terms will have power 1 and the other will have power 0. Logistic regression tries to finds the $\mathbf{w}$ that maximizes the likelihood $L(\mathbf{w})$, which is the same $\mathbf{w}$ that maximizes the log-likelihood $l(\mathbf{w}) = \log L(\mathbf{w})$.

$$\begin{aligned} \arg\max_{\mathbf{w}} L(\mathbf{w}) &= \arg\max_{\mathbf{w}} \log L(\mathbf{w}) \\ &= \arg\max_{\mathbf{w}} l(\mathbf{w}) \end{aligned}$$

Below, we derive how to find $\mathbf{w}$ that maximizes $l(\mathbf{w})$. To make the math compact, we set $\phi(\mathbf{x}) = \mathbf{x}$:

$$\begin{aligned}
\arg\max_{\mathbf{w}} l(\mathbf{w}) &= \arg\max_{\mathbf{w}} \log \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)};\mathbf{w}) \\
&= \arg\max_{\mathbf{w}} \sum_{i=1}^{N} \log p(t^{(i)}|\mathbf{x}^{(i)};\mathbf{w}) \\
&= \arg\max_{\mathbf{w}} \sum_{i=1}^{N} \log \left[p(t=1|\mathbf{x}^{(i)};\mathbf{w})^{t^{(i)}} p(t=0|\mathbf{x}^{(i)};\mathbf{w})^{1-t^{(i)}} \right] \\
&= \arg\max_{\mathbf{w}} \sum_{i=1}^{N} \log \left[p(t=1|\mathbf{x}^{(i)};\mathbf{w})^{t^{(i)}} \right] + \log \left[p(t=0|\mathbf{x}^{(i)};\mathbf{w})^{1-t^{(i)}} \right] \\
&= \arg\max_{\mathbf{w}} \sum_{i=1}^{N} t^{(i)} \log \left[p(t=1|\mathbf{x}^{(i)};\mathbf{w}) \right] + (1-t^{(i)}) \log \left[p(t=0|\mathbf{x}^{(i)};\mathbf{w}) \right] \\
&= \arg\max_{\mathbf{w}} \sum_{i=1}^{N} t^{(i)} \log \left[\sigma(\mathbf{w}^T\mathbf{x}^{(i)}) \right] + (1-t^{(i)}) \log \left[1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)}) \right]
\end{aligned}$$

In order to find the $\mathbf{w}$ that maximizes the expression, we can take its derivative with respect to $\mathbf{w}$:

$$\begin{aligned}
\nabla_{\mathbf{w}} l(\mathbf{w}) &= \nabla_{\mathbf{w}} \sum_{i=1}^{N} t^{(i)} \log \left[\sigma(\mathbf{w}^T\mathbf{x}^{(i)}) \right] + (1-t^{(i)}) \log \left[1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)}) \right] \\
&\text{by chain rule:} \\
&= \sum_{i=1}^{N} t^{(i)} \left(\frac{1}{\sigma(\mathbf{w}^T\mathbf{x}^{(i)})} \right) \times \left(\sigma(\mathbf{w}^T\mathbf{x}^{(i)})(1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)})) \right) \times \mathbf{x}^{(i)} \\
&\quad + (1-t^{(i)}) \left(\frac{1}{1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)})} \right) \times -1 \times \left(\sigma(\mathbf{w}^T\mathbf{x}^{(i)})(1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)})) \right) \times \mathbf{x}^{(i)} \\
&= \sum_{i=1}^{N} t^{(i)} \left(1-\sigma(\mathbf{w}^T\mathbf{x}^{(i)}) \right) \mathbf{x}^{(i)} + (1-t^{(i)})(-1) \left(\sigma(\mathbf{w}^T\mathbf{x}^{(i)}) \right) \mathbf{x}^{(i)} \\
&= \sum_{i=1}^{N} t^{(i)}\mathbf{x}^{(i)} - t^{(i)}\sigma(\mathbf{w}^T\mathbf{x}^{(i)})\mathbf{x}^{(i)} - \sigma(\mathbf{w}^T\mathbf{x}^{(i)})\mathbf{x}^{(i)} + t^{(i)}\sigma(\mathbf{w}^T\mathbf{x}^{(i)})\mathbf{x}^{(i)} \\
&= \sum_{i=1}^{N} t^{(i)}\mathbf{x}^{(i)} - \sigma(\mathbf{w}^T\mathbf{x}^{(i)})\mathbf{x}^{(i)} \\
\nabla_{\mathbf{w}} l(\mathbf{w}) &= \sum_{i=1}^{N} \left(t^{(i)} - \sigma(\mathbf{w}^T\mathbf{x}^{(i)}) \right) \mathbf{x}^{(i)}
\end{aligned}$$

Finally, it is important to note that it is not possible to set $\nabla_{\mathbf{w}} l(\mathbf{w}) = 0$ and solve for $\mathbf{w}$, as the $\sigma(.)$ is a non-linear function. Therefore, the parameter $\mathbf{w}$ is estimated by Gradient Ascent over the likelihood function or by Newton's method. Note: gradient *ascent* is used to *maximize* a measure (ex: likelihood), contrary to gradient *descent* which is used to *minimize* a measure (ex: an error function).