CS 337, Fall 2023
Logistic Regression

Scribes: Arhaan Ahmad*, Prerak Contractor*, Venkatesh*
Tanmay Arun Patil, Harshit K Morj, Ayan Minham Khan
Edited by: Barah Fazili

September 3, 2023

Disclaimer. Please note this document has not received the usual scrutiny that formal publications enjoy. This may be distributed outside this class only with the permission of the instructor.

Remark: In the subsequent text, whenever the term $\mathbf{w}^T\mathbf{x}$ occurs, it is to be assumed that $\mathbf{x}$ is padded with the constant 1 as the first feature.

1 Classification using Linear Regression

Recall: In Linear Regression, the output $\hat{y} \in \mathbb{R}$ for an input $\mathbf{x} \in \mathbb{R}^d$ is estimated by a linear function $\hat{y} = \mathbf{w}^T\mathbf{x}$ using a least square objective.

Classification Task

The classification task is to categorize a given input $\mathbf{x}$ into a class label $\hat{y}$ from the set of class labels $\mathcal{Y}$. The task is referred to as **binary classification** if the set of class labels is $\mathcal{Y} = \{0, 1\}$ (or $\{-1, 1\}$)

A natural way to re-purpose linear regression for binary classification is to predict the class label as follows:

$$\hat{y} = \begin{cases} 1 & \text{if} \quad \mathbf{w}^T\mathbf{x} \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

Where τ acts as a threshold value and is a hyper-parameter for the model.

However, this approach has certain limitations:

- It is not easy to pick a good value for τ.
- It is difficult to calibrate the prediction quality, that is estimating the confidence the model has for a certain prediction.

To overcome these issues, we modify the range of score that the model assigns from $(-\infty, \infty)$ to $(0, 1)$ which can then be interpreted as the "probability" of the input $\mathbf{x}$ being in class $\hat{y} = 1$.

*Most of the content was derived from their submission.

2 Logistic Regression

2.1 Sigmoid Function

The first task is to collapse the score from (∞, ∞) to $(0, 1)$. To do so, we use the **Sigmoid Function**, which is as follows:

$$\boxed{\sigma(x) = \frac{1}{1 + \exp(-x)}}$$

Observe that sigmoid is a monotonically increasing function with the range being $(0, 1)$. Another property which makes it useful for our task is the form it's derivative takes:

$$\begin{aligned}
\sigma'(x) &= \frac{-1}{(1 + e^{-x})^2} \cdot (-e^{-x}) \\
&= \frac{1}{1 + e^{-x}} \cdot \frac{e^{-x}}{1 + e^{-x}} \\
&= \sigma(x) \cdot (1 - \sigma(x))
\end{aligned}$$

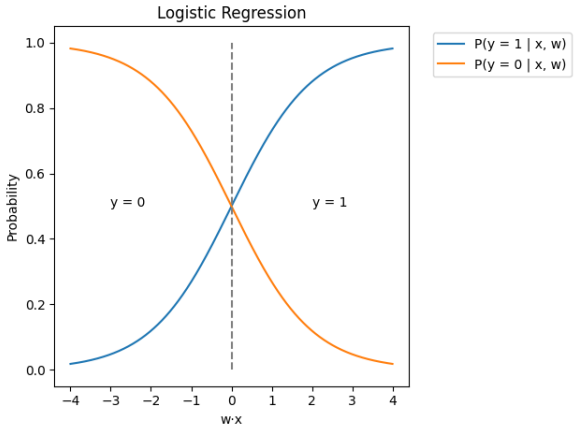
The derivative of the sigmoid function can hence be written in terms of the sigmoid function itself, a property we will use later.

2.2 Model

The model used in logistic regression outputs the probability of the input vector $\mathbf{x}$ having class label $y = 1$ as follows:

$$\boxed{P(y = 1|\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x})}$$

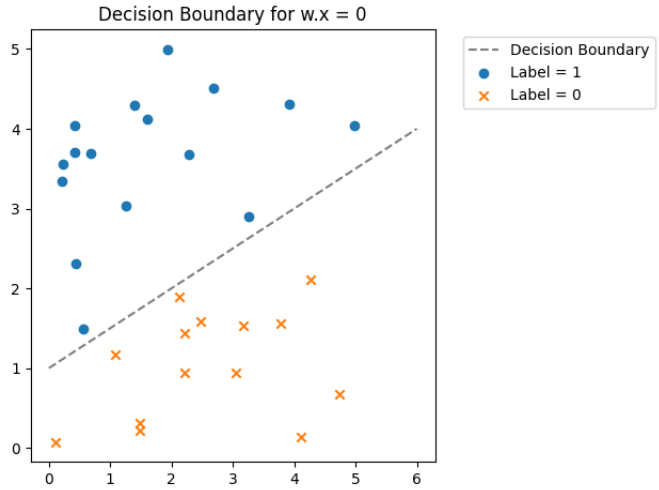
$$\boxed{P(y = 0|\mathbf{x}, \mathbf{w}) = 1 - P(y = 1|\mathbf{x}, \mathbf{w}) = 1 - \sigma(\mathbf{w}^T\mathbf{x})}$$



We then predict the label $\hat{y}$ as the label which has higher probability. Hence $\hat{y} = 1$ if:

$$\begin{aligned} & \frac{P(y=1|\mathbf{x},\mathbf{w})}{P(y=0|\mathbf{x},\mathbf{w})} > 1 \\ \implies & \frac{\sigma\left(\mathbf{w}^T\mathbf{x}\right)}{1-\sigma(\mathbf{w}^T\mathbf{x})} > 1 \\ \implies & \exp\left(\mathbf{w}^T\mathbf{x}\right) > 1 \\ \implies & \mathbf{w}^T\mathbf{x} > 0 \end{aligned}$$

Hence, the **hyperplane** $\mathbf{w}^T\mathbf{x} = 0$ acts as a **decision boundary** in the sense that all input vectors $\mathbf{x}$ lying "above" the boundary (that is $\mathbf{w}^T\mathbf{x} > 0$) are given label 1 and rest are given label 0.



Observe that logistic regression yields a linear decision boundary. Hence it can perfectly classify the training points for **linearly separable data**.

Linearly Separable Data: A data-set $\mathcal{D}$ is linearly separable if $\exists \mathbf{w}$ such that $\forall$ positive training points $(y = 1)$, $\mathbf{w}^T\mathbf{x} > 0$ and $\forall$ negative training points $(y = 0)$, $\mathbf{w}^T\mathbf{x} < 0$.

This puts a severe restriction on the data-sets which can be classified using logistic regression. Consider a simple example, where $\mathbf{x} = [x_1, x_2]^T \in \mathbb{R}^2$:

$$y = \begin{cases} 1 & \text{if} \quad x_1 \cdot x_2 > 0 \\ 0 & \text{otherwise} \end{cases}$$

Observe that a Logistic Regression Classifier will not be able to find a good linear decision boundary for the above data. However, adding a feature to get $\phi(\mathbf{x}) = [1, x_1, x_2, x_1x_2]^T$ makes the data linearly separable and enables the model to find a perfect weight vector $\mathbf{w} = [0, 0, 0, 1]^T$ to classify the data.

Hence, to get better results, we may have to add new features to the input vector before training the model.

2.3 Parameter Estimation

Since the output of model is a probability distribution on the labels, it is natural to estimate the parameters $\mathbf{w}$ as the **Maximum Likelihood Estimate** for the observed data (training data-set). Hence,

$$\begin{aligned}
\mathbf{w}^* &= \underset{\mathbf{w}}{\text{argmax}} \prod_{i=1}^{|\mathcal{D}_{\text{train}}|} P(y_i|\mathbf{x}_i, \mathbf{w}) \\
&= \underset{\mathbf{w}}{\text{argmax}} \sum_{i=1}^{|\mathcal{D}_{\text{train}}|} \log P(y_i|\mathbf{x}_i, \mathbf{w}) \quad \text{(Maximum Conditional Likelihood)} \\
&= \underset{\mathbf{w}}{\text{argmin}} \sum_{i=1}^{|\mathcal{D}_{\text{train}}|} -\log P(y_i|\mathbf{x}_i, \mathbf{w}) \quad \text{(Corresponding Loss Function)}
\end{aligned}$$

Here, $P(y_i|\mathbf{x}_i, \mathbf{w})$ is shorthand for $P(\hat{y}_i = y_i|\mathbf{x}_i, \mathbf{w})$, where $P(\hat{y}_i = 1|\mathbf{x}_i, \mathbf{w}) = \sigma\left(\mathbf{w}^T\mathbf{x}_i\right)$.
Hence, the loss associated with a single data point is:

$$L_{\mathbf{w}}(\mathbf{x}_i, y_i) = \begin{cases} -\log P(y_i = 1|\mathbf{x}_i, \mathbf{w}) & \text{if} \quad y_i = 1 \\ -\log\left(1 - P(y_i = 1|\mathbf{x}_i, \mathbf{w})\right) & \text{if} \quad y_i = 0 \end{cases}$$

Combining them,

$$\boxed{L(\mathbf{w}, \mathbf{x}_i, y_i) = -y_i \log P(y_i = 1|\mathbf{x}_i, \mathbf{w}) - (1 - y_i) \log\left(1 - P(y_i = 1|\mathbf{x}_i, \mathbf{w})\right)}$$

Total loss across the dataset then would be,

$$L(\mathbf{w}, \mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} \underbrace{-y_i \log P(y_i = 1|\mathbf{x}_i, \mathbf{w}) - (1 - y_i) \log\left(1 - P(y_i = 1|\mathbf{x}_i, \mathbf{w})\right)}_{\text{(Binary) Cross Entropy Loss}}$$

This loss is called the binary cross entropy loss because of the similar structure as the formula for cross-entropy loss in information theory. Some properties about the above defined cross entropy loss:

- It is a convex loss function
- It is differentiable
- There is no closed form solution for the optimal parameter $\mathbf{w}^*$. Hence techniques such as gradient descent are used to optimise the model.

To calculate the gradient, note that

$$\begin{aligned}\nabla_{\mathbf{w}}\sigma(\mathbf{w}^T\mathbf{x}_i) &= \sigma(\mathbf{w}^T\mathbf{x}_i)(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\nabla_{\mathbf{w}}\mathbf{w}^T\mathbf{x}_i \\ &= \sigma(\mathbf{w}^T\mathbf{x}_i)(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\mathbf{x}_i \\ \implies \nabla_{\mathbf{w}}\log\sigma(\mathbf{w}^T\mathbf{x}_i) &= (1-\sigma(\mathbf{w}^T\mathbf{x}_i))\mathbf{x}_i \\ \nabla_{\mathbf{w}}\log\big(1-\sigma(\mathbf{w}^T\mathbf{x}_i)\big) &= -\sigma(\mathbf{w}^T\mathbf{x}_i)\mathbf{x}_i\end{aligned}$$

$$\begin{aligned}\implies \nabla_{\mathbf{w}}L(\mathbf{w},\mathcal{D}) &= \sum_{i=1}^{|\mathcal{D}|} -y_i(1-\sigma(\mathbf{w}^T\mathbf{x}_i))\mathbf{x}_i + (1-y_i)\sigma(\mathbf{w}^T\mathbf{x}_i)\mathbf{x}_i \\ &= \sum_{i=1}^{|\mathcal{D}|}(\sigma(\mathbf{w}^T\mathbf{x}_i)-y_i)\mathbf{x}_i \\ &= \sum_{i=1}^{|\mathcal{D}|}(\hat{y}_i-y_i)\mathbf{x}_i\end{aligned}$$

The form of the gradient is very similar to that in linear regression, with $\hat{y} = \sigma(\mathbf{w}^T\mathbf{x}_i)$

This also shows convexity, since

$$\begin{aligned}\nabla^2_{\mathbf{w}}L(\mathbf{w},\mathcal{D}) &= \nabla_{\mathbf{w}}\cdot\nabla_{\mathbf{w}}L(\mathbf{w},\mathcal{D}) \\ &= \sum_{i=1}^{|\mathcal{D}|}\nabla_{\mathbf{w}}\cdot(\sigma(\mathbf{w}^T\mathbf{x}_i)-y_i)\mathbf{x}_i \\ &= \sum_{i=1}^{|\mathcal{D}|}\sigma(\mathbf{w}^T\mathbf{x}_i)(1-\sigma(\mathbf{w}^T\mathbf{x}_i))||\mathbf{x_i}||_2^2 \\ &> 0\end{aligned}$$

and hence, gradient descent would be good option to minimise loss and find optimal paramters.