Logistic Regression

Logisitic regression is a supervised learning algorithm for classification problems that can be used to estimate the probability of a sample belonging to a particular class.

The hypothesis function employed in logistic regression is a type of sigmoid function called the logistic function:

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

The logistic regression model outputs the logistic of the weighted sum of the features plus a bias term. If we form the parameter vector $\mathbf{\Theta}$, and the vector of features, $\mathbf{x}$,

$$\mathbf{\Theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{2}$$

where the bias term $x_0 = 1$, we can compute

$$\mathbf{\Theta}^T\mathbf{x} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + ... + \theta_n x_n \tag{3}$$

The hypothesis function is defined as

$$h_\Theta(\mathbf{x}) = \frac{1}{1 + exp(-\mathbf{\Theta}^T\mathbf{x})} \tag{4}$$

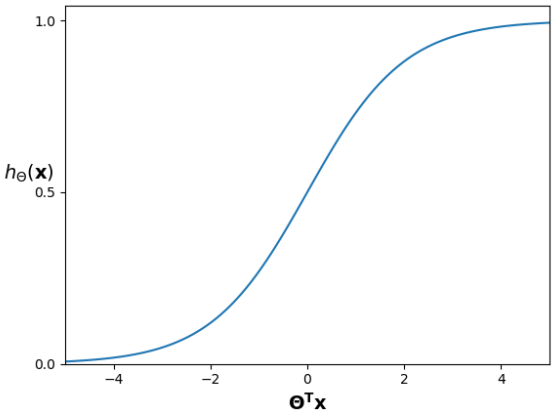


Figure 1: The hypothesis function, $h_\theta(\mathbf{x})$, plotted against the weighted sum of the features plus bias term, $\mathbf{\Theta}^T\mathbf{x}$.

Figure 1 is a visualisation of the logistic regression hypothesis function. As $\mathbf{\Theta}^T\mathbf{x} \to \infty$, $h_\Theta(\mathbf{x}) \to 1$, and as $\mathbf{\Theta}^T\mathbf{x} \to -\infty$, $h_\Theta(\mathbf{x}) \to 0$. At $\mathbf{\Theta}^T\mathbf{x} = 0$, $h_\Theta(\mathbf{x}) = 0.5$. If the output of a sample is greater than or equal to 0.5, then the class is predicted positive, and if the output of a sample is less than 0.5 then the class is predicted negative. More formally, this is

$$y = \begin{cases} 0 & for \quad h_\Theta(\mathbf{x}) < 0.5 \\ 1 & for \quad h_\Theta(\mathbf{x}) \geq 0.5 \end{cases} \tag{5}$$

The hypothesis output can be interpreted as the probability of a sample belonging to the positive ($y = 1$) class. For example, if the output for a sample is 0.83, there is an 83% probability that this sample belongs to the positive class.

The logistic regression decision boundary is defined by $h_\Theta(\mathbf{x}) = 0.5$, which corresponds to $\mathbf{\Theta}^T\mathbf{x} = 0$. If we have two features in our dataset, the decision boundary can be plotted as a line. For more than two features this decision boundary will be a hyperplane. The decision boundary separates the two classes and indicates which class will be predicted for a new sample. It is determined entirely by the training data.

For example, say we have the dataset of two features, x_1 and x_2, shown in Figure 2, and have found the parameters θ_i, such that

$$\mathbf{\Theta} = \begin{pmatrix} -2 \\ 1 \\ 1 \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$$

We can plot the decision boundary on Figure 2:

$$\mathbf{\Theta}^T\mathbf{x} = -2 + x_1 + x_2 = 0$$

Therefore, the decision boundary is the straight line given by $x_1 + x_2 = 2$. This means that for a new sample, y will be predicted as

$$y = \begin{cases} 0 & for \quad x_1 + x_2 < 2 \\ 1 & for \quad x_1 + x_2 \geq 2 \end{cases}$$

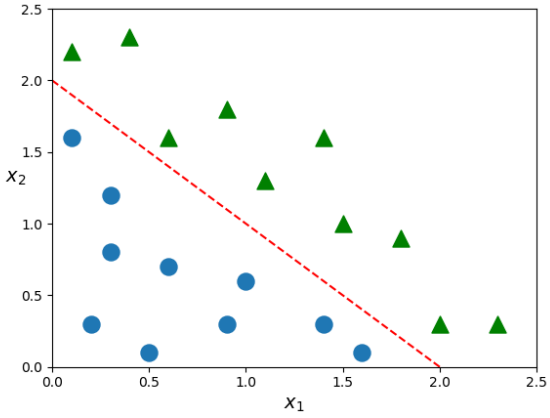


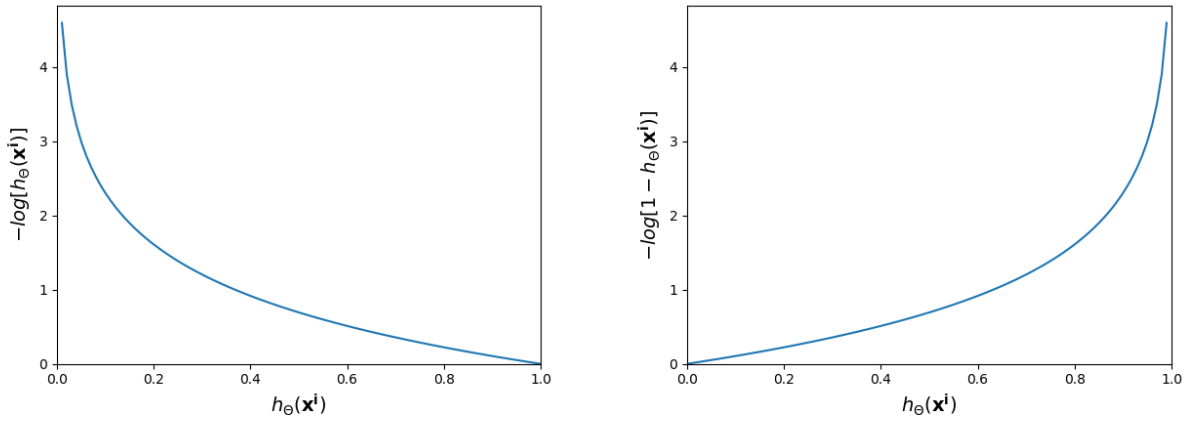
Figure 2: In this dataset, the negative samples are represented by the blue circles, and the positive samples by the green triangles. The decision boundary (red dotted-line) separates the classes and is defined by the equation $\mathbf{\Theta}^T\mathbf{x} = 0$.

The Cost Function

The logistic regression cost function is given by

$$C(\boldsymbol{\Theta}) = -\frac{1}{m}\sum_{i=1}^{m}\left\{y^i\ log[h_\Theta(\mathbf{x}^i)] + (1-y^i)\ log[1-h_\Theta(\mathbf{x}^i)\right\} \tag{6}$$

(see Appendix for derivation) where the notation $\mathbf{x}^i$ denotes the i^{th} training sample and y^i the corresponding value of y. $h_\Theta(\mathbf{x}^i)$ is the hypothesis of the i^{th} training sample.



(a) The $y = 1$ term.

(b) The $y = 0$ term.

Figure 3: The two terms of the cost function.

For the i^{th} sample, if $y^i = 1$ then the second term in the cost function vanishes. The remaining term (Figure 3(a)) tends to zero as $h_\Theta(\mathbf{x}^i)$ approaches one. Therefore, the cost function is minimised when the hypothesis output is close to the actual value $y^i = 1$. Similarly, if $y^i = 0$ the first term in the cost function vanishes and the remaining term (Figure 3(b)) tends to zero as $h_\Theta(\mathbf{x}^i)$ approaches zero. So again, we see that the cost function is minimised when the hypothesis output is close to the actual value $y^i = 0$.

Now let's show an example of how the cost function is calculated explicitly to see how it works:

Say we have a dataset of only five training samples which have the classifications $y^0 = 1$, $y^1 = 0$, $y^2 = 1$, $y^3 = 1$ and $y^4 = 0$, so that $y = (1, 0, 1, 1, 0)^T$.

The cost function becomes

$$C(\boldsymbol{\Theta}) = -\frac{1}{5}\{y^0 log[h_\Theta(\mathbf{x}^0)] + (1-y^1)log[1-h_\Theta(\mathbf{x}^1)] + y^2 log[h_\Theta(\mathbf{x}^2)] + y^3 log[h_\Theta(\mathbf{x}^3)] \\ +(1-y^4)log[h_\Theta(\mathbf{x}^4)]\}$$

For some initial $\boldsymbol{\Theta} = (\theta_0, \theta_1, \theta_2, \theta_3, \theta_4)^T$, let's say the cost function yields

$$\begin{aligned} C(\boldsymbol{\Theta}) &= -\frac{1}{5}[log(0.72) + log(1 - 0.31) + log(0.68) + log(0.8) + log(1 - 0.29)] \\ &= -\frac{1}{5}(-0.143 - 0.161 - 0.168 - 0.097 - 0.149) \\ &= 0.143 \end{aligned}$$

Now, the θ_i are refined (we will see how in the next section), and the cost function yields, let's say

$$\begin{aligned} C(\boldsymbol{\Theta}) &= -\frac{1}{5}[log(0.76) + log(1 - 0.26) + log(0.71) + log(0.84) + log(1 - 0.22)] \\ &= -\frac{1}{5}(-0.119 - 0.131 - 0.149 - 0.076 - 0.108) \\ &= 0.117 \end{aligned}$$

As the θ_i are further refined, $-log[1-h_\Theta(\mathbf{x}^i)]$ is reduced for the $y = 0$ terms, and $-log[h_\Theta(\mathbf{x}^i)]$ is reduced for the $y = 1$ terms. Consequently, the cost function decreases.

Gradient Descent

We have seen that the logistic regression cost function is minimised when the outputs, $h_\Theta(\mathbf{x}^i)$, are close in value to the y^i. Therefore, our goal in producing an accurate classifier should be to minimise the cost function. Fortunately, the cost function is convex (see Appendix) and has a global minimum. Unfortunately, there does not exist a direct analytic solution for the θ_i analogous to the normal equation in linear regression. However, we can employ gradient descent to minimise the logistic regression cost function as in linear regression. The idea behind gradient descent is to descend the surface of the cost function in the direction opposite to the gradient vector in a series of iterative steps until arriving at the global minimum (see Linear Regression for further discussion).

To this end we repeatedly apply the following step simultaneously for all θ_i:

$$\theta_i \to \theta_i - \alpha \frac{\partial}{\partial \theta_i}[C(\boldsymbol{\Theta})] \tag{7}$$

where α is the learning rate which determines the size of the step taken. The gradient vector is given by

$$\boldsymbol{\nabla} C = \begin{pmatrix} \frac{\partial C}{\partial \theta_0} \\ \frac{\partial C}{\partial \theta_1} \\ \vdots \\ \frac{\partial C}{\partial \theta_n} \end{pmatrix} \tag{8}$$

Therefore, applying the step simultaneously for all θ_i is equivalent to applying the step

$$\boldsymbol{\Theta} \to \boldsymbol{\Theta} - \alpha \boldsymbol{\nabla} C \tag{9}$$

$\boldsymbol{\nabla} C$ points in the direction of steepest ascent of the surface of C. Hence, the step made in gradient descent is made in the direction of steepest descent of the surface. When the minimum is reached $\boldsymbol{\nabla} C = 0$ and gradient descent stops.

To write (7) explicitly we need to compute the partial derivatives of C. The partial derivative of C with respect to θ_0 is

$$\begin{aligned}\frac{\partial C}{\partial \theta_0} &= \frac{1}{m}\sum_{i=1}^{m}\left\{-y^i\frac{\partial}{\partial \theta_0}\left[log\left(\frac{1}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}\right)\right] - (1-y^i)\frac{\partial}{\partial \theta_0}\left[log\left(1-\frac{1}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}\right)\right]\right\} \\ &= \frac{1}{m}\sum_{i=1}^{m}\left\{-y^i\frac{\partial}{\partial \theta_0}\left[log\left(\frac{1}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}\right)\right] - (1-y^i)\frac{\partial}{\partial \theta_0}\left[log\left(\frac{e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}\right)\right]\right\}\end{aligned}$$

The partial derivative terms are calculated as

$$\begin{aligned}\frac{\partial}{\partial \theta_0}\left[log\left(\frac{1}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}\right)\right] &= (1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i})(-1)(1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i})^{-2}(-e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}) \\ &= \frac{e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial \theta_0}\left[log\left(\frac{e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}\right)\right] &= \frac{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}\left[\frac{-e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}} + e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}(-1)(1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i})^{-2}(-e^{-\boldsymbol{\Theta}^T\mathbf{x}^i})\right] \\ &= \frac{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}\left[\frac{-e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}} + \frac{e^{-2\boldsymbol{\Theta}^T\mathbf{x}^i}}{(1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i})^2}\right] \\ &= \frac{e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}} - 1\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}\frac{\partial C}{\partial \theta_0} &= \frac{1}{m}\sum_{i=1}^{m}\left[-y^i\frac{e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}} - (1-y^i)\left(\frac{e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}} - 1\right)\right] \\ &= \frac{1}{m}\sum_{i=1}^{m}\left[1 - \frac{e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}} - y^i\right] \\ &= \frac{1}{m}\sum_{i=1}^{m}\left[\frac{1}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}} - y^i\right] \\ &= \frac{1}{m}\sum_{i=1}^{m}\left[h_\Theta(\mathbf{x}^i) - y^i\right] \qquad (10)\end{aligned}$$

The partial derivative of C with respect to θ_j is

$$\frac{\partial C}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left\{-y^i\frac{\partial}{\partial \theta_j}\left[log\left(\frac{1}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}\right)\right] - (1-y^i)\frac{\partial}{\partial \theta_j}\left[log\left(\frac{e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}{1+e^{-\boldsymbol{\Theta}^T\mathbf{x}^i}}\right)\right]\right\}$$

The partial derivative terms are calculated as

$$\frac{\partial}{\partial\theta_j}\left[log\left(\frac{1}{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}}\right)\right] = (1+e^{-\mathbf{\Theta}^T\mathbf{x}^i})(-1)(1+e^{-\mathbf{\Theta}^T\mathbf{x}^i})^{-2}(-{x_j}^i e^{-\mathbf{\Theta}^T\mathbf{x}^i})$$
$$= \frac{{x_j}^i e^{-\mathbf{\Theta}^T\mathbf{x}^i}}{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}}$$

$$\frac{\partial}{\partial\theta_0}\left[log\left(\frac{exp(\mathbf{\Theta}^T\mathbf{x}^i)}{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}}\right)\right] = \frac{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}}{e^{-\mathbf{\Theta}^T\mathbf{x}^i}}\left[\frac{-{x_j}^i e^{-\mathbf{\Theta}^T\mathbf{x}^i}}{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}} + e^{-\mathbf{\Theta}^T\mathbf{x}^i}(-1)(1+e^{-\mathbf{\Theta}^T\mathbf{x}^i})^{-2}(-{x_j}^i e^{-\mathbf{\Theta}^T\mathbf{x}^i})\right]$$
$$= \frac{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}}{e^{-\mathbf{\Theta}^T\mathbf{x}^i}}\left[\frac{-{x_j}^i e^{-\mathbf{\Theta}^T\mathbf{x}^i}}{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}} + \frac{{x_j}^i e^{-2\mathbf{\Theta}^T\mathbf{x}^i}}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}^i})^2}\right]$$
$$= {x_j}^i\left[\frac{e^{-\mathbf{\Theta}^T\mathbf{x}^i}}{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}} - 1\right]$$

Therefore, we obtain

$$\frac{\partial C}{\partial\theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left\{-y^i\frac{{x_j}^i e^{-\mathbf{\Theta}^T\mathbf{x}^i}}{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}} - (1-y^i){x_j}^i\left[\frac{e^{-\mathbf{\Theta}^T\mathbf{x}^i}}{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}} - 1\right]\right\}$$
$$= \frac{1}{m}\sum_{i=1}^{m}\left\{{x_j}^i\left[1 - \frac{e^{-\mathbf{\Theta}^T\mathbf{x}^i}}{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}} - y^i\right]\right\}$$
$$= \frac{1}{m}\sum_{i=1}^{m}\left\{{x_j}^i\left[\frac{1}{1+e^{-\mathbf{\Theta}^T\mathbf{x}^i}} - y^i\right]\right\}$$
$$= \frac{1}{m}\sum_{i=1}^{m}[h_\Theta(\mathbf{x}^i) - y^i]{x_j}^i \tag{11}$$

As ${x_0}^i = 1$, we can say that

$$\frac{\partial}{\partial\theta_j}[C(\mathbf{\Theta})] = \frac{1}{m}\sum_{i=1}^{m}[h_\Theta(\mathbf{x}^i) - y^i]{x_j}^i, \qquad for\ j = 0, 1, 2, ... \tag{12}$$

Hence, (7) becomes

$$\theta_j \to \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}[h_\Theta(\mathbf{x}^i) - y^i]{x_j}^i, \qquad for\ j = 0, 1, 2, ... \tag{13}$$

The Regularised Cost Function

As with other Machine Learning methods, logistic regression models can suffer from overfitting. Large feature coefficients are characteristic of overfitted models; small variations in the value of a feature multiplied by a large coefficient result in significant variations in the output and a model that is susceptible to learning any noise in the feature. Regularisation is a common strategy employed to reduce model variance by means of minimising the feature coefficients (see Linear Regression for further discussion). As in linear regression, we can add a regularisation term to the cost function which becomes

$$C(\boldsymbol{\Theta}) = -\frac{1}{m}\sum_{i=1}^{m}\left\{y^i\ log[h_\Theta(\mathbf{x}^i)] + (1-y^i)\ log[1-h_\Theta(\mathbf{x}^i)\right\} + \frac{\lambda}{2m}\sum_{j=1}^{n}{\theta_j}^2 \tag{14}$$

The regularisation term is (strictly) convex (see Linear Regression Appendix), and as two or more convex functions added together produce another convex function, the cost function remains convex. The partial derivatives of the regularised cost function are

$$\frac{\partial C}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}[h_\Theta(\mathbf{x}^i) - y^i], \qquad \frac{\partial C}{\partial \theta_j} = \frac{1}{m}\left\{\sum_{i=1}^{m}[h_\Theta(\mathbf{x}^i) - y^i]{x_j}^i + \lambda\theta_j\right\} \tag{15}$$

It follows that the gradient descent algorithm becomes

$$\theta_0 \to \theta_0 - \frac{\alpha}{m}\sum_{i=1}^{m}[h_\Theta(\mathbf{x}^i) - y^i], \qquad \theta_j \to \theta_j - \frac{\alpha}{m}\left\{\sum_{i=1}^{m}[h_\Theta(\mathbf{x}^i) - y^i]{x_j}^i + \lambda\theta_j\right\} \tag{16}$$

repeated simultaneously for all θ_j.

References

Jurafsky, D., Martin, J. (2020). *Speech and Language Processing*, third edition (draft).

Ng, A. *Machine Learning*. Offered by Stanford University on Coursera.

Appendix

0.1 Deriving the logistic regression cost function

Let's say the variable x can take only the values 0 or 1. x takes the value 1 with probability $h_\Theta(x)$, which means it takes the value 0 with probability $1 - h_\Theta(x)$. We can write the probability density function as

$$P(y) = \begin{cases} 1 - h_\Theta(x) & for \quad y = 0 \\ h_\Theta(x) & for \quad y = 1 \end{cases} \tag{17}$$

We can see that encouraging the correct classification of a sample is equivalent to maximising $P(y)$; if $y = 0$ then $P(y)$ is largest when $h_\Theta(x) = 0$, and if $y = 1$, $P(y)$ is largest when $h_\Theta(x) = 1$.

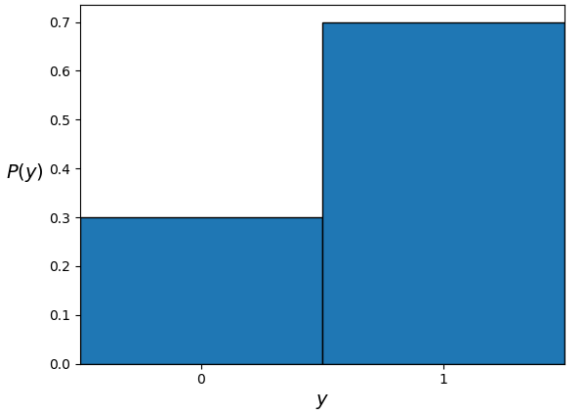


Figure 4: $P(y)$ for $h_\Theta(x) = 0.7$.

$P(y)$ is the Bernoulli distribution and can equivalently be written as

$$P(y) = h_\Theta(x)^y [1 - h_\Theta(x)]^{1-y} \tag{18}$$

It turns out to be be advantageous to maximise the log of this expression rather than the expression itself, and fortunately, whichever values maximise $P(y)$ will also maximise $log[P(y)]$. Taking the log of both sides yields

$$log[P(y)] = log[h_\Theta(x)^y] + log\left\{[1 - h_\Theta(x)]^{1-y}\right\} \tag{19}$$

$$= ylog[h_\Theta(x)] + (1 - y)log[1 - h_\Theta(x)] \tag{20}$$

Instead of maximising this result we choose to minimise its negative. Hence, we obtain the cost function

$$C = -ylog[h_\Theta(x)] - (1 - y)log[1 - h_\Theta(x)] \tag{21}$$

which is minimised to encourage the correct classification of a sample.

0.2 The logistic regression cost function is convex

The cost function is

$$C(\mathbf{\Theta}) = -y\ log[h_\Theta(\mathbf{x})] - (1-y)\ log[1-h_\Theta(\mathbf{x})] \tag{22}$$

where, let's say

$$\mathbf{\Theta}^T\mathbf{x} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \tag{23}$$

From equations (10) and (11) we can calculate the second partial derivatives of C:

$$\frac{\partial^2 C}{\partial\theta_0^2} = \frac{e^{-\mathbf{\Theta}^T\mathbf{x}}}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}})^2}, \quad \frac{\partial^2 C}{\partial\theta_1^2} = \frac{{x_1}^2 e^{-\mathbf{\Theta}^T\mathbf{x}}}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}})^2}, \quad \frac{\partial^2 C}{\partial\theta_2^2} = \frac{{x_2}^2 e^{-\mathbf{\Theta}^T\mathbf{x}}}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}})^2}$$

$$\frac{\partial^2 C}{\partial\theta_0\partial\theta_1} = \frac{\partial^2 C}{\partial\theta_1\partial\theta_0} = \frac{x_1 e^{-\mathbf{\Theta}^T\mathbf{x}}}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}})^2}, \quad \frac{\partial^2 C}{\partial\theta_0\partial\theta_2} = \frac{\partial^2 C}{\partial\theta_2\partial\theta_0} = \frac{x_2 e^{-\mathbf{\Theta}^T\mathbf{x}}}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}})^2}$$

$$\frac{\partial^2 C}{\partial\theta_1\partial\theta_2} = \frac{\partial^2 C}{\partial\theta_2\partial\theta_1} = \frac{x_1 x_2 e^{-\mathbf{\Theta}^T\mathbf{x}}}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}})^2}$$

The Hessian matrix of C is

$$H = \begin{pmatrix} \frac{\partial^2 C}{\partial\theta_0^2} & \frac{\partial^2 C}{\partial\theta_0\partial\theta_1} & \frac{\partial^2 C}{\partial\theta_0\partial\theta_2} \\ \frac{\partial^2 C}{\partial\theta_1\partial\theta_0} & \frac{\partial^2 C}{\partial\theta_1^2} & \frac{\partial^2 C}{\partial\theta_1\partial\theta_2} \\ \frac{\partial^2 C}{\partial\theta_2\partial\theta_0} & \frac{\partial^2 C}{\partial\theta_2\partial\theta_1} & \frac{\partial^2 C}{\partial\theta_2^2} \end{pmatrix} \tag{24}$$

$$= \frac{1}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}})^2} \begin{pmatrix} e^{-\mathbf{\Theta}^T\mathbf{x}} & x_1 e^{-\mathbf{\Theta}^T\mathbf{x}} & x_2 e^{-\mathbf{\Theta}^T\mathbf{x}} \\ x_1 e^{-\mathbf{\Theta}^T\mathbf{x}} & {x_1}^2 e^{-\mathbf{\Theta}^T\mathbf{x}} & x_1 x_2 e^{-\mathbf{\Theta}^T\mathbf{x}} \\ x_2 e^{-\mathbf{\Theta}^T\mathbf{x}} & x_1 x_2 e^{-\mathbf{\Theta}^T\mathbf{x}} & {x_2}^2 e^{-\mathbf{\Theta}^T\mathbf{x}} \end{pmatrix} \tag{25}$$

To find the eigenvalues of this Hessian matrix we need to solve the equation

$$\frac{1}{(1+e^{-\mathbf{\Theta}^T\mathbf{X}})^2} \begin{vmatrix} e^{-\mathbf{\Theta}^T\mathbf{x}} - \lambda & x_1 e^{-\mathbf{\Theta}^T\mathbf{x}} & x_2 e^{-\mathbf{\Theta}^T\mathbf{x}} \\ x_1 e^{-\mathbf{\Theta}^T\mathbf{x}} & {x_1}^2 e^{-\mathbf{\Theta}^T\mathbf{x}} - \lambda & x_1 x_2 e^{-\mathbf{\Theta}^T\mathbf{x}} \\ x_2 e^{-\mathbf{\Theta}^T\mathbf{x}} & x_1 x_2 e^{-\mathbf{\Theta}^T\mathbf{x}} & {x_2}^2 e^{-\mathbf{\Theta}^T\mathbf{x}} - \lambda \end{vmatrix} = 0 \tag{26}$$

Computing the determinant we obtain

$$\lambda^2 \left[\lambda - \frac{e^{-\mathbf{\Theta}^T\mathbf{x}}}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}})^2} - \frac{{x_1}^2 e^{-\mathbf{\Theta}^T\mathbf{x}}}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}})^2} - \frac{{x_2}^2 e^{-\mathbf{\Theta}^T\mathbf{x}}}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}})^2} \right] = 0 \tag{27}$$

This equation has the solutions

$$\lambda = 0, \qquad \lambda = \frac{e^{-\mathbf{\Theta}^T\mathbf{x}}}{(1+e^{-\mathbf{\Theta}^T\mathbf{x}})^2} \left(1 + {x_1}^2 + {x_2}^2\right) \tag{28}$$

These are the eigenvalues of the Hessian matrix. Note that the second eigenvalue is always positive. A matrix is positive semidefinite if it is symmetric and all its eigenvalues are non-negative, with at least one equal to zero. Therefore, the Hessian matrix of C is positive semidefinite. A convex function has the property that its Hessian matrix is positive semidefinite. Hence, C is a convex function. This holds for any number of x_n.