50.007 Machine Learning

Homework 4

Logistic Regression

Berrak Sisman

Assistant Professor, ISTD Pillar, SUTD

Graded by TA

Question 1.1 [20 pts]

Please indicate whether the following statements are true (T) or false (F).

a) Logistic regression is a supervised machine learning algorithm. True [5 pts]

b) Logistic regression is mainly used for regression, not classification. False [5 pts]

c) Logistic regression outputs a probability or confidence score, i.e., a value between 0 and 5. False [5 pts]

d) It is possible to apply a logistic regression algorithm on a 3-class classification problem. True [5 pts]

Question 1.2 [20 pts]

Suppose that you have trained a logistic regression classifier, and it outputs on a new example a prediction $h_\theta(x) = 0.48$. This means (check all that apply):

1) Our estimate for $P(y = 0|x; \theta)$ is 0.52 **True [5 pts]**

2) Our estimate for $P(y = 0|x; \theta)$ is 0.48 **False [5 pts]**

3) Our estimate for $P(y = 1|x; \theta)$ is 0.52 **False [5 pts]**

4) Our estimate for $P(y = 1|x; \theta)$ is 0.48 **True [5 pts]**

Please explain your answer.

- **Note that $h_\theta(x)$ is the estimated probability that $y = 1$ on input x. If $h_\theta(x)$=0.48, it means that $P(y = 1|x; \theta) = 0.48$.**
- **Please also note that $p(y = 1|x; \theta)$+$p(y = 0|x; \theta) = 1$. So $p(y = 0|x; \theta)$ will be equal to 0.52.**

Question 1.3 [20 pts]

Suppose you train a logistic classifier $h_\theta(x) = g(\theta_0 + x_1\theta_1 + x_2\theta_2)$, and obtain $\theta = [6 \ -6 \ 2]^T$. Please formulate the decision boundary of your classifier. Note that this is a binary classification problem, which means class label y can be 0 or 1.

Answer:

$$\theta^T x = [6 \quad -6 \quad 2] \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = 6 - 6x_1 + 2x_2$$

Let's try to figure out where the hypothesis ends of predicting $y = 0$ and $y = 1$

$y = 1$ if $6 - 6x_1 + 2x_2 \geq 0$ $\qquad$ $y = 0$ if $6 - 6x_1 + 2x_2 < 0$

$\boldsymbol{y = 1}$ **if** $\boldsymbol{3 \geq 3x_1 - x_2}$ $\qquad$ $\boldsymbol{y = 0}$ **if** $\boldsymbol{3 < 3x_1 - x_2}$

(Please note that decision boundary is a property of hypothesis and the parameters…)

[20 pts]

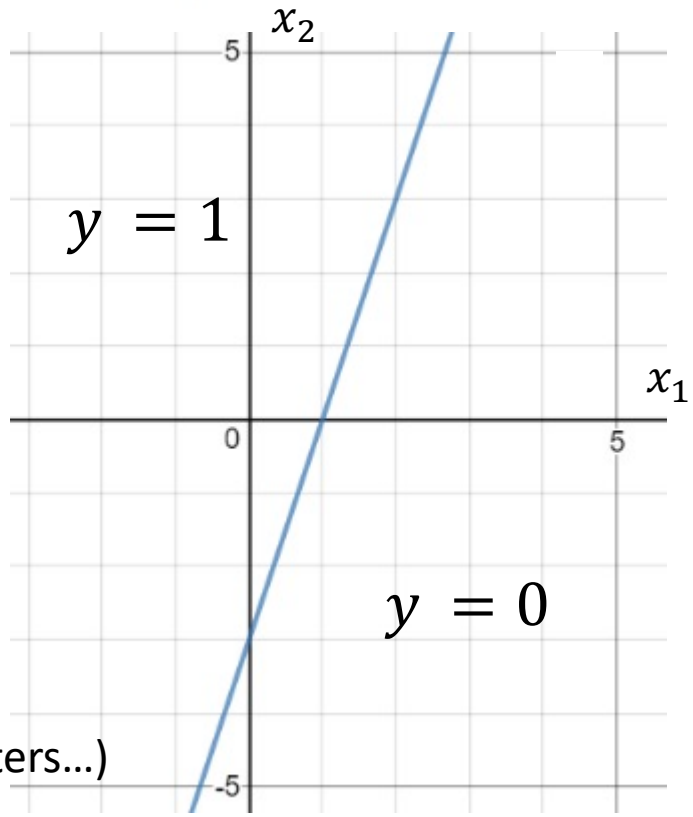


Figure is not compulsory

Question 1.4 [20 pts]

Suppose you train a logistic classifier $h_\theta(x) = g(\theta_0 + x_1\theta_1 + x_2\theta_2 + x_1^2\theta_3 + x_2^2\theta_4)$, and obtain $\theta = [-9\ 0\ 0\ 4\ 1]^T$. Please formulate the decision boundary of your classifier. Note that this is a binary classification problem, which means class label y can be 0 or 1.

Answer:

$$\theta^T x = [-9 \quad 0 \quad 0 \quad 4 \quad 1]\begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix} = 4x_1^2 + x_2^2 - 9$$

Let's try to figure out where the hypothesis ends of predicting $y = 0$ and $y = 1$

$y = 1$ if $-9 + 4x_1^2 + x_2^2 \geq 0$ $\qquad$ $y = 0$ if $-9 + 4x_1^2 + x_2^2 < 0$

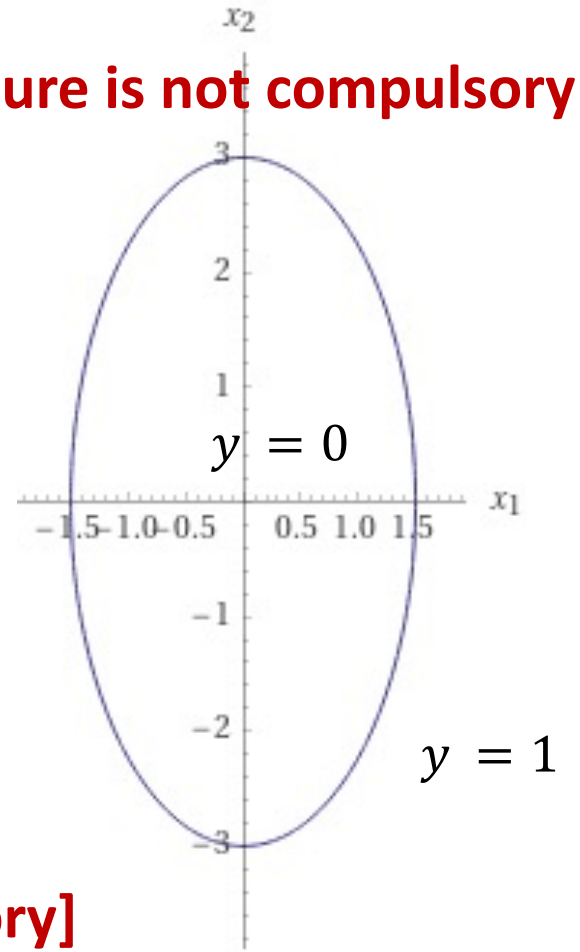
$y = 1$ if $4x_1^2 + x_2^2 \geq 9$ $\qquad$ $y = 0$ if $4x_1^2 + x_2^2 < 9$

(Please note that decision boundary is a property of hypothesis and the parameters…)

How does the decision boundary look like? It is an ellipse!

[not compulsory]

Figure is not compulsory



[20 pts]

Question 1.5 [20 pts]

In logistic regression, we find the parameters of a logistic (sigmoid) function that maximize the likelihood of a set of training examples. The likelihood is given as follows:

$$\prod_{i=1}^{n} P(y^{(i)}|x^{(i)}) \tag{1}$$

However, we re-express the problem of maximizing the likelihood as minimizing the following expression:

$$\frac{1}{n}\sum_{i=1}^{n} log\left(1 + exp\left(-y^{(i)}\left(\theta.x^{(i)} + \theta_0\right)\right)\right) \tag{2}$$

What is the benefit of optimizing the log-likelihood rather than the likelihood of the data? In other words, why is this expression computationally more "convenient"? *(Hint: try randomly generating, say, 1,000 probabilities in Python and multiplying them together as in Eq. 1.)*

Answer:

Progressively multiplying many probabilities together as in Equation 1 quickly gives a result that is too small to be representable in computer memory (this is known as an underflow problem). In contrast, Equation 2 uses a sum over terms that makes this problem less likely to occur.

[20 pts]