eMasters in Communication Systems

Prof. Aditya Jagannatham

Elective Module:

Advanced ML Techniques

Chapter 3

Logistic Regression

Linear vs Logistic Regression

- ***Linear regression*** is well suited…
  - *when the response variable y is*

    CONTINUOS ________________

Linear vs Logistic Regression

- ***Linear regression*** is well suited…
  - *when the response variable y is* <u>**continuous**</u>

Linear vs Logistic Regression

DISCRETE

- What about when y is **discrete**?

$y = 0$
$y = 1$

  - *Example: y is **binary**, i.e., $y \in \{0,1\}$*
  - *This is precisely handled by*

LOGISTIC REGRESSION

Linear vs Logistic Regression

- What about when y is **discrete**?
  - *Example: y is **binary**, i.e., $y \in \{0,1\}$*
  - *This is precisely handled by **Logistic Regression***

Linear vs Logistic Regression

- Example
  - ***Image/ video***: Person present or absent
  - ***Medical imaging***: Disease present or absent.

Linear vs Logistic Regression

- Example
  - ***Image/ video***: *Person present/ absent*
  - ***Medical imaging***: *Disease present/ absent*

Logistic function

- The **logistic function** is given below

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

- Also termed the Sigmoid.

Logistic function

- The **logistic function** is given below

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

- Also termed the **sigmoid function**

Logistic function

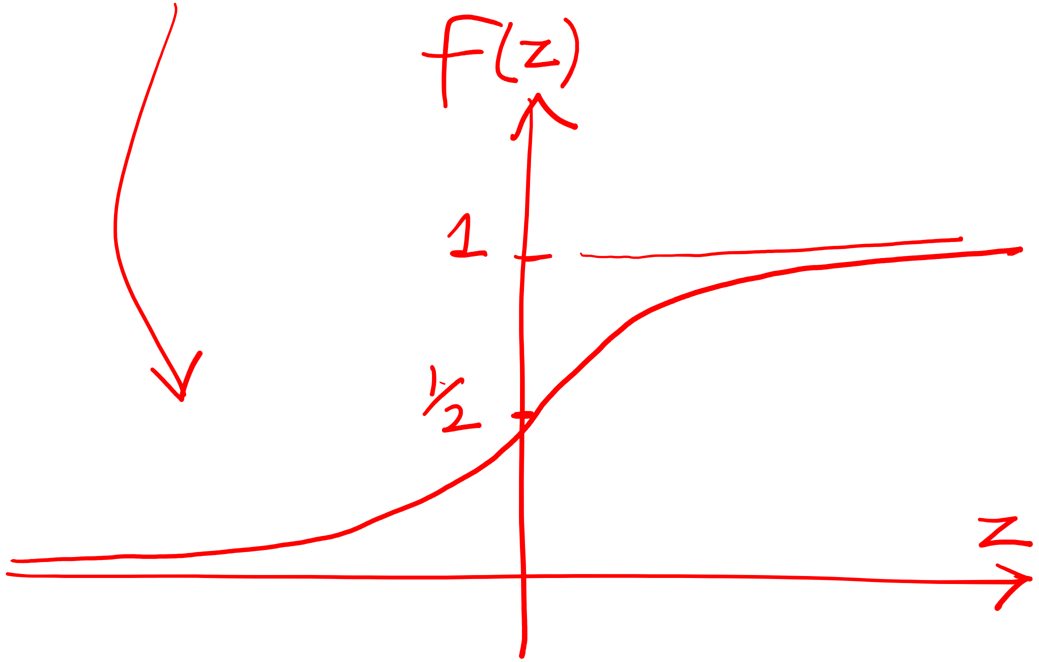
- Observe

$$\lim_{z\to\infty} \frac{1}{1+e^{-z}} \to 1$$

$$\lim_{z\to-\infty} \frac{1}{1+e^{-z}} \to 0$$

$$z=0 \quad f(z)=\frac{1}{1+1}=\frac{1}{2}$$

Plot of Logistic Function

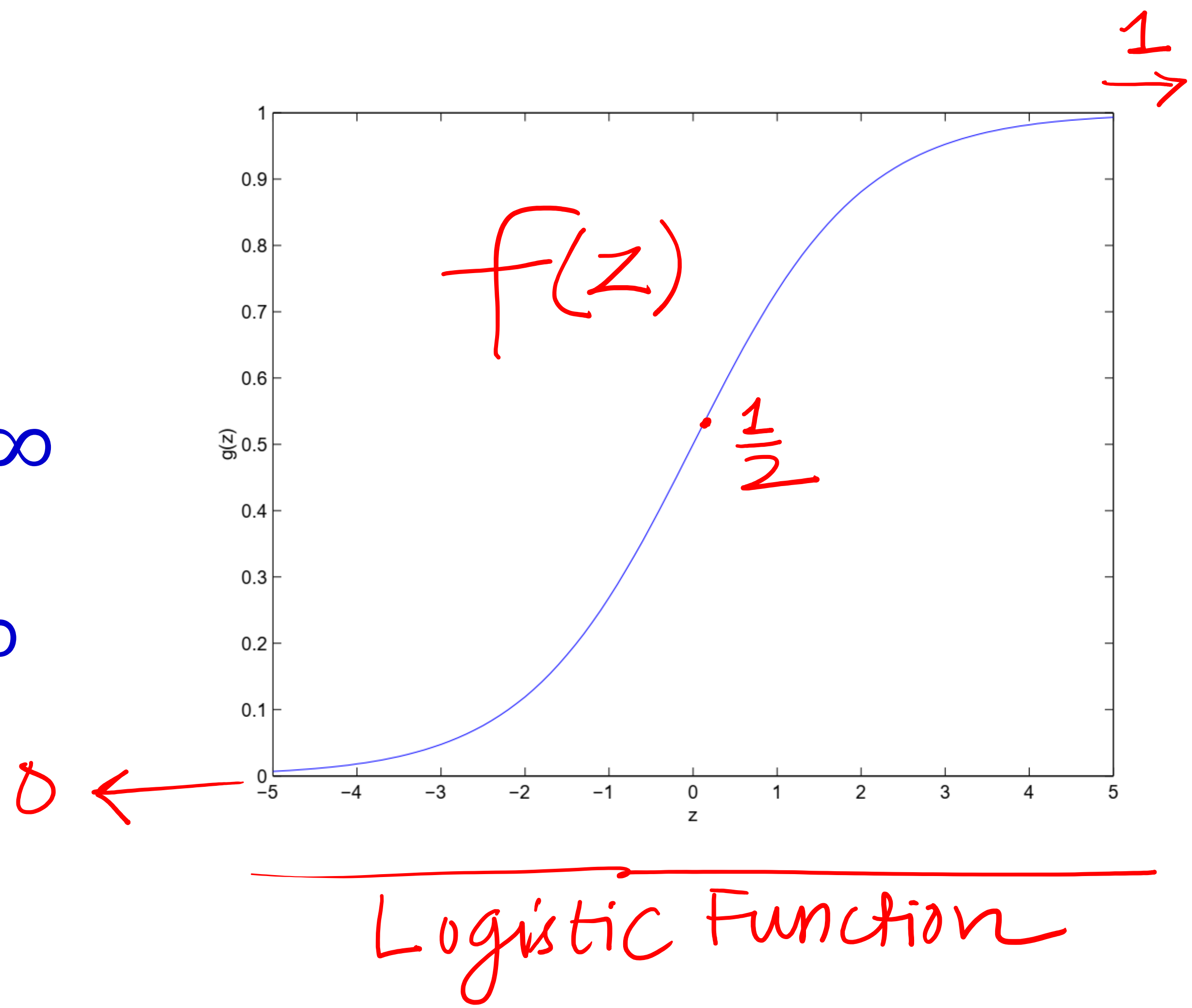


Logistic function

- Observe

$$\frac{1}{1+e^{-z}} \to 0 \text{ as } z \to -\infty$$

$$\frac{1}{1+e^{-z}} \to 1 \text{ as } z \to \infty$$



Probability

$\bar{h}$ = Regression coefficient vector

- In ***Logistic Regression***, the probabilities are given as

$$p(y=1|\bar{x}) = \frac{1}{1+e^{-\bar{x}^T\bar{h}}} = g(\bar{x})$$

$$p(y=0|\bar{x}) = 1 - p(y=1|\bar{x})$$

$$= \frac{e^{-\bar{x}^T\bar{h}}}{1+e^{-\bar{x}^T\bar{h}}}$$

Probability

- In ***Logistic Regression***, the probabilities are given as

Modeling probabilities.
$y=1$
$y=0$.

$$P(y=1|\bar{\mathbf{x}}) = \frac{1}{1+e^{-\bar{\mathbf{x}}^T\bar{\mathbf{h}}}} = g(\bar{\mathbf{x}})$$

Probability

$$P(y = 0|\bar{\mathbf{x}}) = \frac{e^{-\bar{\mathbf{x}}^T\bar{\mathbf{h}}}}{1 + e^{-\bar{\mathbf{x}}^T\bar{\mathbf{h}}}} = 1 - g(\bar{\mathbf{x}})$$

$1 - p(y=1|\bar{x})$.

Probability

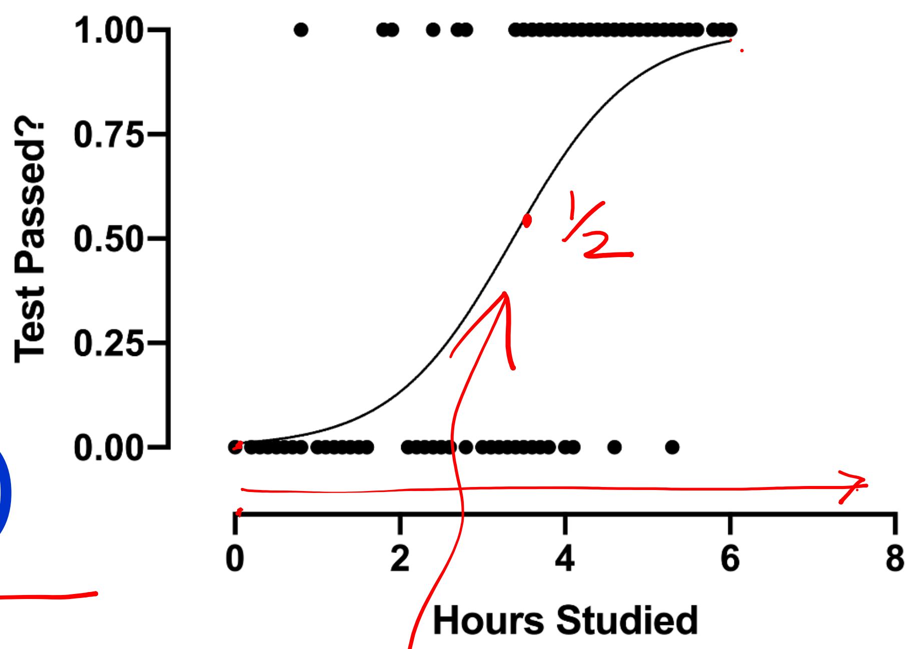
- **Example:**

Binary Response

Discrete Response.

$p(y=1|x)$

$P(\text{Pass}|\text{Hours studied})$



Logistic Function

Response = y

Pass = 1

Fail = 0

x = number of hours studied.

Likelihood

Training Data

Supervised Learning

- How to determine the **regression parameter** $\bar{\mathbf{h}}$ in this case?
  - We use the *the* Maximum Likelihood.

Likelihood

- How to determine the **regression parameter** $\bar{\mathbf{h}}$ in this case?
  - *We use the* ***Maximum Likelihood*** (ML) *technique*

Likelihood

- The **likelihood** of $\big(y(k), \bar{\mathbf{x}}(k)\big)$ can be written as

$$p\big(y(k)=1 \mid \bar{x}(k)\big) = g\big(\bar{x}(k)\big)$$

$$p\big(y(k) \mid \bar{x}(k)\big) = \Big(\underbrace{g\big(\bar{x}(k)\big)}_{p(y(k)=1)}\Big)^{y(k)} \Big(\underbrace{1 - g\big(\bar{x}(k)\big)}_{p(y(k)=0)}\Big)^{1-y(k)}$$

Probability of Response.

Likelihood

Response

Regressor

input.

- The **likelihood** of $\big(y(k), \bar{\mathbf{x}}(k)\big)$ can be written as

Training pair

$$\Big(g\big(\bar{\mathbf{x}}(k)\big)\Big)^{y(k)}\Big(1 - g\big(\bar{\mathbf{x}}(k)\big)\Big)^{1-y(k)}$$

Likelihood

- The **joint likelihood** of all outputs/ responses is given as

independent

Product of the probabilities.

$$L(\bar{\mathbf{h}}) = \prod_{k=1}^{M} p(y(k) \mid \bar{x}(k))$$

M = Number of training points.

$$= \underbrace{\prod_{k=1}^{M} \left(g(\bar{x}(k))\right)^{y(k)} \left(1-g(\bar{x}(k))\right)^{1-y(k)}}_{\text{Likelihood function}}$$

Likelihood

- The **joint likelihood** of all outputs/ responses is given as

$$\underbrace{\mathbf{L}(\bar{\mathbf{h}}) = \prod_{k=1}^{M}\left(g\left(\bar{\mathbf{x}}(k)\right)\right)^{y(k)}\left(\overbrace{1-g\left(\bar{\mathbf{x}}(k)\right)}^{\frac{e^{-\bar{x}^T\bar{h}}}{1+e^{-\bar{x}^T\bar{h}}}}\right)^{1-y(k)}}_{\text{Likelihood of } \bar{h}}$$

Direct maximization challenging!

$$g(\bar{x}(k)) = \frac{1}{1+e^{-\bar{x}^T\bar{h}}}$$

Likelihood

- The **log-likelihood** is given as

$$\ln L(\bar{\mathbf{h}}) = l(\bar{\mathbf{h}})$$

$$= \ln \prod_{k=1}^{M} g(\bar{x}(k))^{y(k)} \cdot \left(1 - g(\bar{x}(k))\right)^{(1-y(k))}$$

$$= \sum_{k=1}^{M} y(k) \ln g(\bar{x}(k)) + (1-y(k)) \ln(1 - g(\bar{x}(k)))$$

Likelihood

- The **log-likelihood** is given as

maximize loglikelihood.

$$\underbrace{\ln L(\bar{\mathbf{h}}) = l(\bar{\mathbf{h}})}_{\text{No closed form solution.}}$$

$$\underbrace{= \sum_{k=1}^{M} y(k) \ln g(\bar{\mathbf{x}}(k)) + (1 - y(k)) \ln\left(1 - g(\bar{\mathbf{x}}(k))\right)}_{\text{Log Likelihood.}}$$

concave.

Maximum Likelihood

- To maximize the **log-likelihood**, one can employ the ***gradient ascent*** technique

$$\overbrace{\bar{h}(k+1) = \bar{h}(k) + \frac{\eta}{2} \cdot \nabla \ell(\bar{h})}^{\text{Gradient ascent.}}$$

Gradient of Log likelihood.

Maximum Likelihood

- The **update rule** reduces to

$$\bar{h}(k+1) = \bar{h}(k) + \eta\, e(k+1)\, \bar{x}(k+1)$$

LMS Rule !

error

$$e(k+1) = y(k+1) - g(\bar{x}(k+1))$$

Observed Response.

Predicted Response.

Maximum Likelihood

- The **update rule** reduces to

identical to
~~LMS~~ rule
Least mean Squares.

$$\bar{\mathbf{h}}(k+1) = \bar{\mathbf{h}}(k) + \eta\Big(y(k+1) - g\big(\bar{\mathbf{x}}(k+1)\big)\Big)\bar{\mathbf{x}}(k+1)$$

Logistic Regression
update Rule.

Maximum Likelihood

Complexity very low.

- Observe this is similar to the **LMS update rule**!

Stochastic Gradient ascent.

$$\bar{h}(k+1) = \bar{h}(k) + \eta\overbrace{\left(y(k+1) - g\left(\bar{x}(k+1)\right)\right)}^{e(k+1)}\bar{x}(k+1)$$

$$\underbrace{\bar{h}(k+1) = \bar{h}(k) + \eta\, e(k+1)\,\bar{x}(k+1)}_{\text{Update Rule}} \quad \text{Identical to LMS Rule}$$

Update Rule.

Online Algorithm

Perceptron Learning Algorithm

- In the **perceptron learning algorithm**, g is given as the threshold function

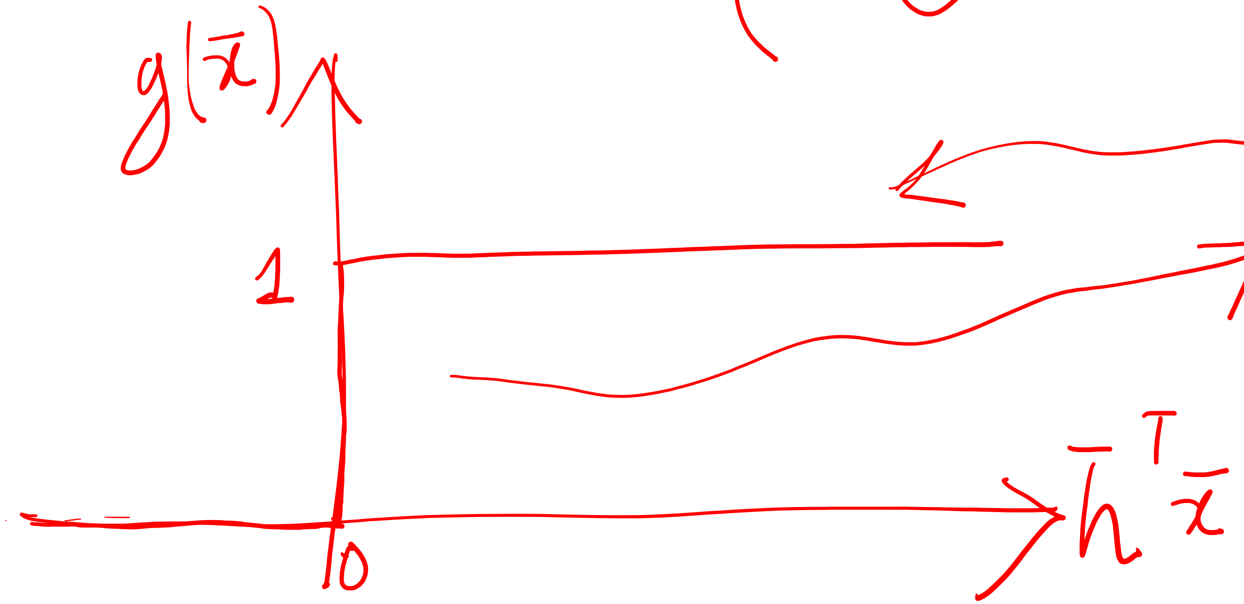
$$F(z) = \frac{1}{1+e^{-z}}$$

Smooth $\Rightarrow$ differentiable.

$$g(\bar{\mathbf{x}}) = \begin{cases} 1 & \text{if } \bar{h}^T \bar{x} \geq 0 \\ 0 & \text{if } \bar{h}^T \bar{x} < 0 \end{cases}$$

unit Step Function

Not smooth.



Perceptron Learning Algorithm

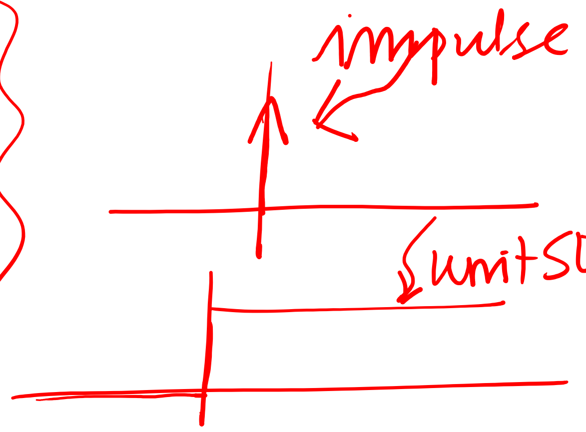
- In the **perceptron learning algorithm**, g is given as the threshold function — UNIT STEP.

$$g(\bar{\mathbf{x}}) = \begin{cases} 1 & \bar{\mathbf{h}}^T \bar{\mathbf{x}} \geq 0 \\ 0 & \bar{\mathbf{h}}^T \bar{\mathbf{x}} < 0 \end{cases}$$

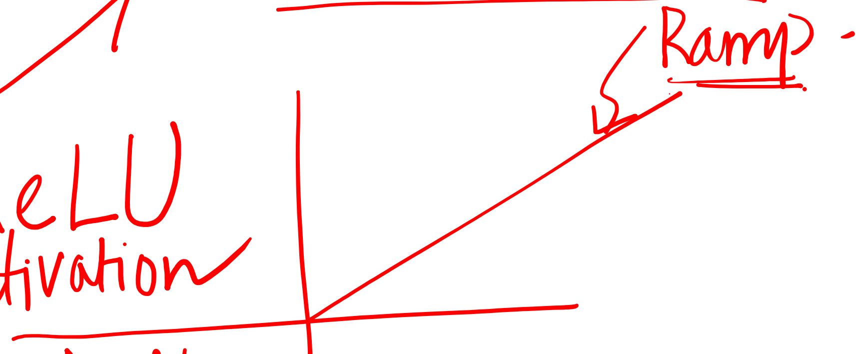
Activation Function

Neuron Firing model.

Threshold = 0.



Unit step (threshold)



ReLU Activation

Rectified Linear Unit.

Perceptron Learning Algorithm

- The **update rule** is once again given as

$$\bar{h}(k+1) = \bar{h}(k) + \eta \cdot e(k+1)\,\bar{x}(k+1)$$

$$e(k+1) = y(k+1) - g(\bar{x}(k+1))$$

Perceptron Learning Algorithm

Initial model for neuron

- The **update rule** is once again given as

$$\bar{\mathbf{h}}(k+1) = \bar{\mathbf{h}}(k) + \eta\,\underbrace{\Big(y(k+1) - g\big(\bar{\mathbf{x}}(k+1)\big)}_{e(k+1)}\Big)\,\bar{\mathbf{x}}(k+1)$$

Same update rule as LMS!

Perceptron Learning Algorithm

- This is termed the Perceptron Learning Algorithm
  - *It was developed as an approximate model for the* ***neurons*** *in the* Human brain.

Instructors may use this white area (14.5 cm / 25.4 cm) for the text.
Three options provided below for the font size.

Font: Avenir (Book), Size: 32, Colour: Dark Grey

Font: Avenir (Book), Size: 28, Colour: Dark Grey

Font: Avenir (Book), Size: 24, Colour: Dark Grey

Do not use the space below.