

# Logistic regression

## Machine Learning (CSCI 5770G)

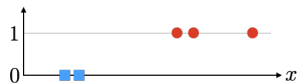
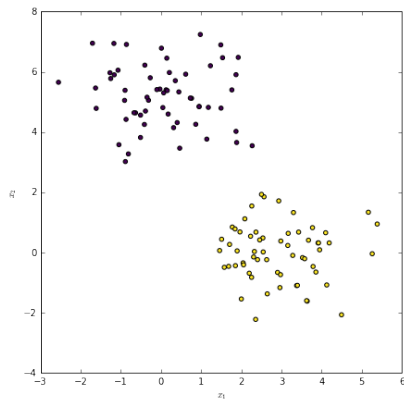
Faisal Z. Qureshi

<http://vclab.science.ontariotechu.ca>



# Logistic regression

- ▶ Logistic regression is for **binary classification**
- ▶ The target variable  $y$  takes on values in  $\{0, 1\}$



## Binary classification

The goal of binary classification is to learn  $h_{\theta}(\mathbf{x})$ , which can be used to assign a label  $y \in \{0, 1\}$  to the input  $\mathbf{x}$ . Label  $y$  takes values in  $\{0, 1\}$ , so we can use Bernoulli distribution to specify its probability distribution. Specifically

$$\Pr(y = 1) = h_{\theta}(\mathbf{x})$$

$$\Pr(y = 0) = 1 - h_{\theta}(\mathbf{x})$$

Or more succinctly

$$\Pr(y) = h_{\theta}(\mathbf{x})^y (1 - h_{\theta}(\mathbf{x}))^{1-y}$$

## Bernoulli distribution

A Bernoulli random variable  $X$  takes values in  $\{0, 1\}$

$$\begin{aligned}\Pr(X|\theta) &= \begin{cases} \theta & \text{if } X = 1 \\ 1 - \theta & \text{otherwise} \end{cases} \\ &= \theta^X (1 - \theta)^{1-X}\end{aligned}$$

### Example usage

Bernoulli distribution  $\text{Ber}(X|\theta)$  can be used to model coin tosses.

## Likelihood for binary classification

Under the assumption that data is independent and identically distributed (i.e., i.i.d.) the likelihood for the entire data is

$$\Pr(y|\mathbf{X}, \theta) = \prod_{i=1}^N h_{\theta}(\mathbf{x}^{(i)})^{y^{(i)}} \left(1 - h_{\theta}(\mathbf{x}^{(i)})\right)^{1-y^{(i)}}$$

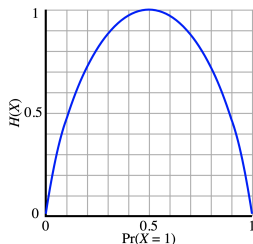
What form should  $h_{\theta}(\cdot)$  take?

# Entropy

- ▶ Average level of information in a random variable.
- ▶ Given a discrete random variable  $X$ , which takes values in the alphabet  $\mathcal{X}$  and is distributed according to  $p : \mathcal{X} \rightarrow [0, 1]$ :

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) = \mathbb{E}_{x \sim p(x)}[-\log p(x)]$$

- ▶ Choice of base for log varies with applications
  - ▶ Base **2** gives the unit of **bits** or **shannons**
  - ▶ Base  **$e$**  gives units of **nats**
  - ▶ Base **10** gives units of **dits**, **bans**, or **hartley**

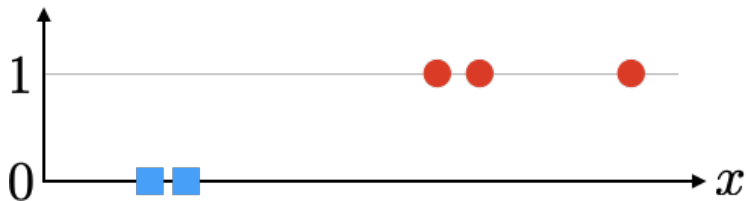


## Cross entropy

- ▶ Cross-entropy between two distributions  $p$  and  $q$  is a measure of the average number of bits needed to identify an event from a set  $\mathcal{X}$  with true distribution  $p$  when the coding scheme used for the set is optimized for an estimated probability distribution  $q$

$$H(p, q) = - \sum_{x \in \mathcal{X}} p(x) \log q(x) = -\mathbb{E}_{x \sim p(x)}[\log q(x)]$$

Lets consider a simple 1D case for binary classification

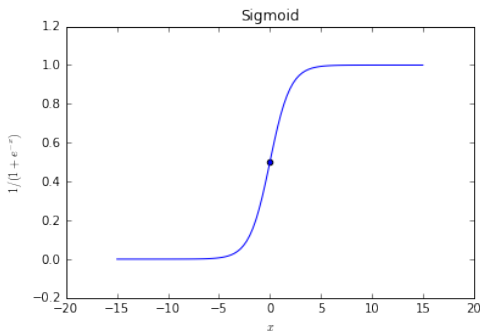




# Sigmoid function

$\text{sigm}(x)$  refers to a *sigmoid* function, also known as the *logistic* or *logit* function.

$$\text{sigm}(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$



## Logistic regression

For logistic regression, we set  $h_{\theta}(\mathbf{x}) = \text{sigm}(\mathbf{x}^T \theta)$ . So

$$\Pr(y|\mathbf{X}, \theta) = \prod_{i=1}^N \left[ \frac{1}{1 + e^{-\mathbf{x}^{(i)T} \theta}} \right]^{y^{(i)}} \left[ 1 - \frac{1}{1 + e^{-\mathbf{x}^{(i)T} \theta}} \right]^{1-y^{(i)}}$$

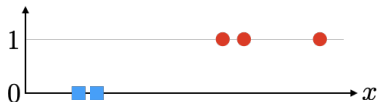
where

$$\mathbf{x}^T \theta = \theta_0 + \sum_{i=1}^M \theta_i \mathbf{x}_i$$

# Sigmoid function

$$\Pr(y|x, \theta) = \left[ \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}} \right]^y \left[ 1 - \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}} \right]^{1-y}$$

- ▶  $\theta = (\theta_0, \theta_1)$  are model parameters.
- ▶  $\theta_0$  controls the shift.
- ▶  $\theta_1$  controls the scale (how steep is the slope of the sigmoid function).



# MLE for logistic regression (1)

## Likelihood

$$L(\theta) = \Pr(y|\mathbf{X}, \theta)$$

## Negative log-likelihood

$$\begin{aligned} l(\theta) &= -\log L(\theta) \\ &= -\sum_{i=1}^N y^{(i)} \log h_{\theta}(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \log(1 - h_{\theta}(\mathbf{x}^{(i)})) \end{aligned}$$

*We prefer to work in the log domain for mathematical convenience. Plus there are numerical advantages of working in the log domain.*

## MLE for logistic regression (2)

### Goal

Our goal is to find parameters  $\theta$  that maximize the likelihood (or minimize the negative log-likelihood).

$$\theta^* = \arg \min_{\theta} l(\theta)$$

## Derivative of sigmoid

$$\begin{aligned}\frac{d}{dx}\text{sigm}(x) &= \frac{d}{dx} \frac{1}{1 + e^{-x}} \\ &= \frac{-(-1)e^{-x}}{(1 + e^{-x})^2} \\ &= \left( \frac{e^{-x}}{1 + e^{-x}} \right) \left( \frac{1}{1 + e^{-x}} \right) \\ &= \left( \frac{1 - 1 + e^{-x}}{1 + e^{-x}} \right) \left( \frac{1}{1 + e^{-x}} \right) \\ &= \left( 1 - \frac{1}{1 + e^{-x}} \right) \left( \frac{1}{1 + e^{-x}} \right) \\ &= (1 - \text{sigm}(x)) \text{sigm}(x)\end{aligned}$$

## Gradient of a sigmoid w.r.t. $\theta$

We know that

$$\frac{d}{dx} \text{sigm}(x) = (1 - \text{sigm}(x)) \text{sigm}(x)$$

It follows

$$\frac{d}{d\theta} \text{sigm}(\mathbf{x}^T \theta) = (1 - \text{sigm}(\mathbf{x}^T \theta)) \text{sigm}(\mathbf{x}^T \theta) \mathbf{x}$$

## MLE for logistic regression

Negative log likelihood contribution by sample  $i$

$$\begin{aligned}l^{(i)}(\theta) &= -y^{(i)} \log h_{\theta}(\mathbf{x}^{(i)}) \\ &\quad - (1 - y^{(i)}) \log(1 - h_{\theta}(\mathbf{x}^{(i)})) \\ &= -y^{(i)} \log \text{sigm}(\mathbf{x}^{(i)T} \theta) \\ &\quad - (1 - y^{(i)}) \log(1 - \text{sigm}(\mathbf{x}^{(i)T} \theta))\end{aligned}$$

Gradient of  $l^{(i)}(\theta)$ :

$$\nabla_{\theta} l^{(i)} = ?$$



# MLE for logistic regression

## Notation change

- ▶ Replacing  $\text{sigm}(\mathbf{x}^{(i)T})$  with  $s$
- ▶ Replacing  $y^{(i)}$  with  $y$
- ▶ Replacing  $\mathbf{x}^{(i)}$  with  $\mathbf{x}$

$$\begin{aligned}\nabla_{\theta} l^{(i)} &= \nabla_{\theta} [-y \log s - (1 - y) \log(1 - s)] \\ &= -y \frac{s(1-s)\mathbf{x}}{s} - (1-y) \frac{s(1-s)\mathbf{x}}{1-s} \\ &= -y\mathbf{x} + ys\mathbf{x} - s\mathbf{x} - ys\mathbf{x} \\ &= -y\mathbf{x} - s\mathbf{x} \\ &= -\mathbf{x}(y - s)\end{aligned}$$

Therefore (after fixing the notation),

$$\nabla_{\theta} l^{(i)} = -\mathbf{x}^{(i)}(y^{(i)} - h_{\theta}(\mathbf{x}^{(i)}))$$

# MLE for logistic regression

Gradient of  $l(\theta)$  for  $i$ th example

$$\nabla_{\theta} l^{(i)} = -\mathbf{x}^{(i)} (y^{(i)} - h_{\theta}(\mathbf{x}^{(i)}))$$

Stochastic gradient descent rule

$$\begin{aligned}\theta^{(k+1)} &= \theta^{(k)} - \eta \nabla l^{(i)}(\theta) \\ &= \theta^{(k)} + \eta \mathbf{x}^{(i)} (y^{(i)} - h_{\theta}(\mathbf{x}^{(i)})) \\ &= \theta^{(k)} + \eta \mathbf{x}^{(i)} (y^{(i)} - \text{sigm}(\mathbf{x}^{(i)T} \theta)),\end{aligned}$$

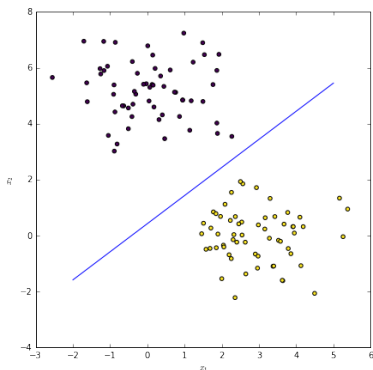
where  $\eta$  is the **learning rate** and  $k$  refers to the gradient descent iteration (step).

# Logistic regression for binary classification

Given a point  $\mathbf{x}^{(*)}$ , classify using the following rule

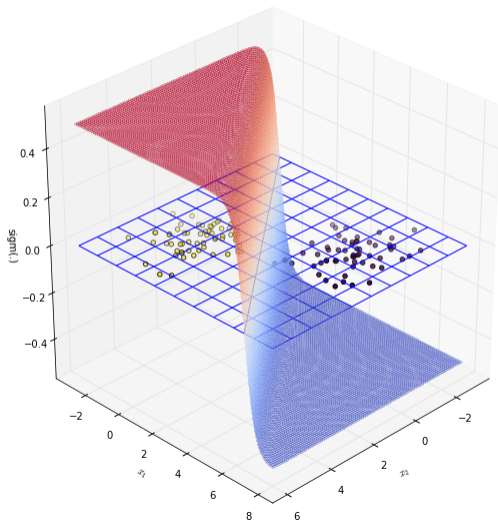
$$y^{(*)} = \begin{cases} 1 & \text{if } \Pr(y|\mathbf{x}^{(*)}, \theta) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

The decision boundary is  $\mathbf{x}^T \theta = 0$ .  
Recall that this is where the sigmoid function is 0.5.



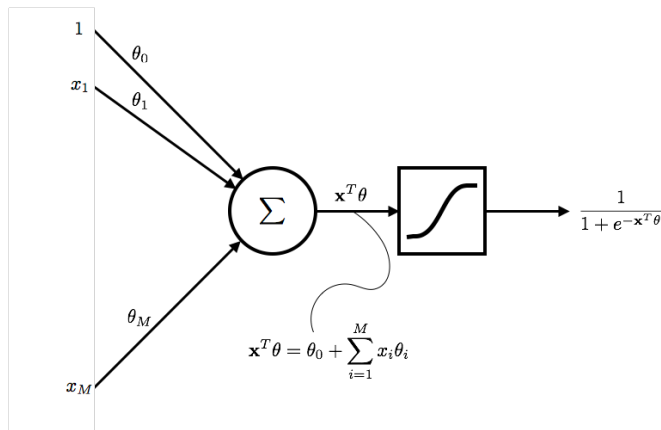
# Logistic regression for binary classification

- ▶ The decision boundary is  $\mathbf{x}^T \boldsymbol{\theta} = 0$ 
  - ▶ This is where sigmoid function is 0.5



## Network view of logistic regression

- ▶ By changing the activation function to sigmoid and using the cross-entropy loss instead the least-squares loss that we use for linear regression, we are able to perform binary classification.



Artificial neuron

# Summary

- ▶ We looked at logistic regression, a binary classifier.
- ▶ Bernoulli distribution
- ▶ Linear regression and logistic regression topics provide an excellent opportunity to study and understand the concepts underpinning neural networks

# Copyright and License

©Faisal Z. Qureshi



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.