Lecture 3: Logistic regression

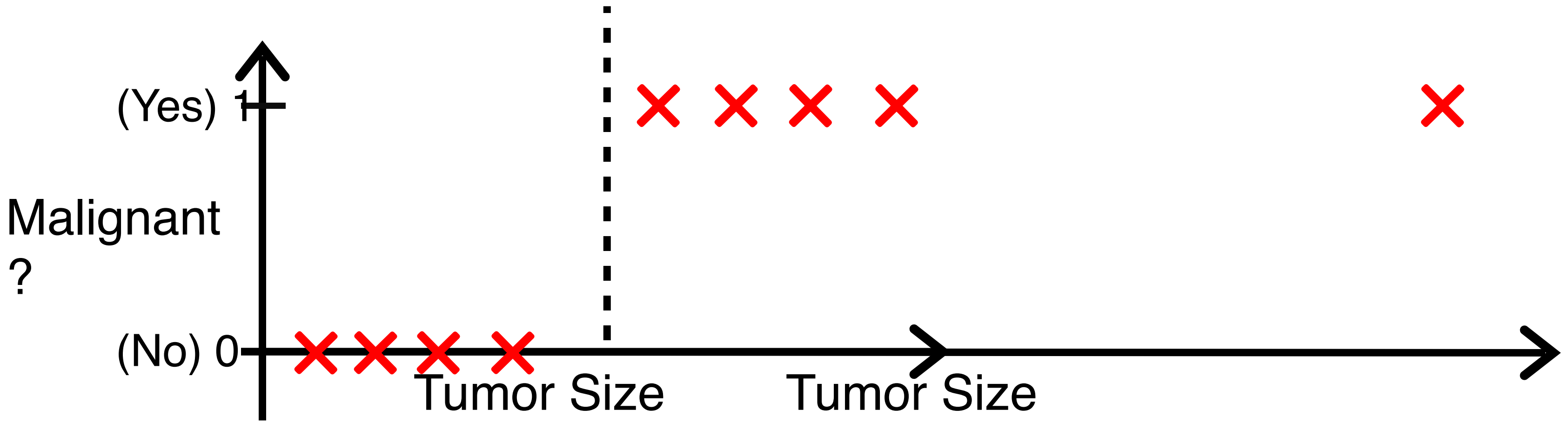
Classification

- Email: Spam / Not Spam?
- Online Transactions: Fraudulent (Yes / No)?
- Tumor: Malignant / Benign ?

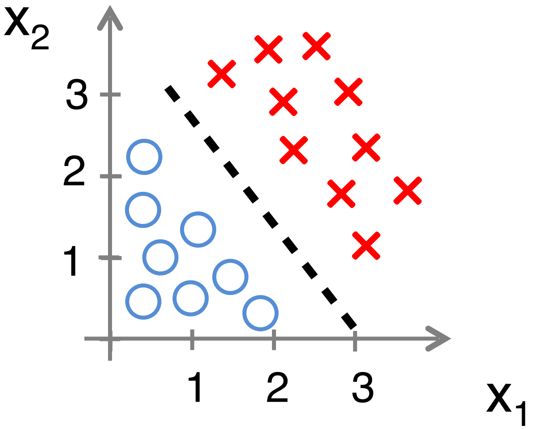
$$y \in \{0, 1\}$$

0: “Negative Class” (e.g., benign tumor)

1: “Positive Class” (e.g., malignant tumor)



Threshold classifier output


x2
3
2
1
1
2
3
x1

Perceptron

A perceptron, first introduced in 1958, is a deterministic binary classifier of the following form:

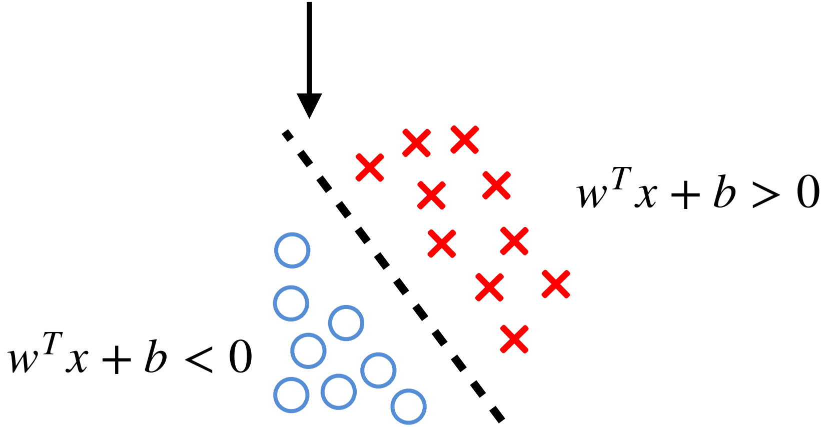
$$h_\theta(x) = \mathbb{I}(w^T x + b > 0)$$

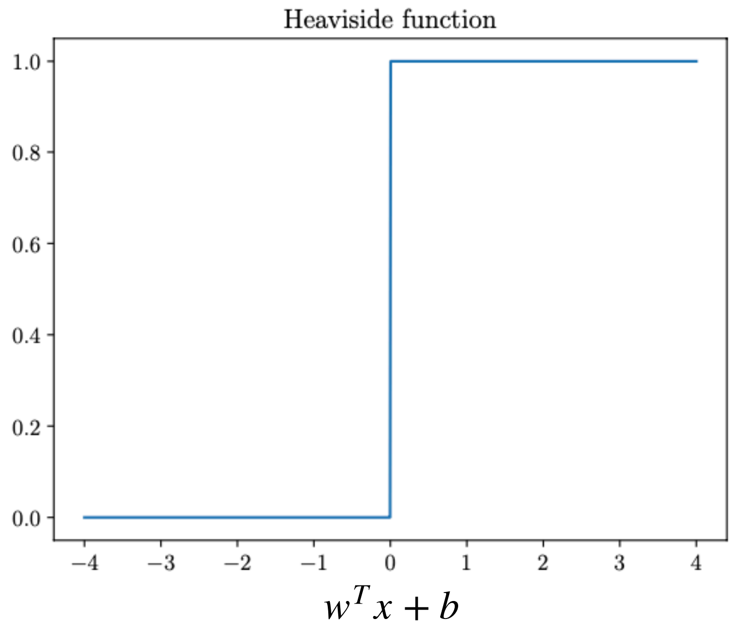
Recall that $w^T x = [w_0 \quad w_1 \quad w_3 \quad \cdots] \begin{bmatrix} x_0 \\ x_1 \\ x_3 \\ \vdots \end{bmatrix} = \sum_i w_i x_i$

b: bias term

$\mathbb{I}$: Heaviside function

Decision boundary: a linear ‘subspace’ where $w^T x + b = 0$





$$h_\theta(x) = \mathbb{I}(w^T x + b > 0)$$

Probabilistic formulation

- Bernoulli distribution for binary random variables (e.g. outcomes of flipping a coin)
- ψ: probability of getting the outcome ‘1’ (malignant)

$$\text{Ber}(y\,|\,\psi) = \begin{cases} 1-\psi & \text{if } y=0 \\ \psi & \text{if } y=1 \end{cases}$$

- This can be concisely expressed as

$$\text{Ber}(y\,|\,\psi) = \psi^{y}(1-\psi)^{1-y}$$

- Model the probability ψ as a function of the input X

- Perceptron: y = 0 or 1

$h_\theta(x)$ can be 1 or 0

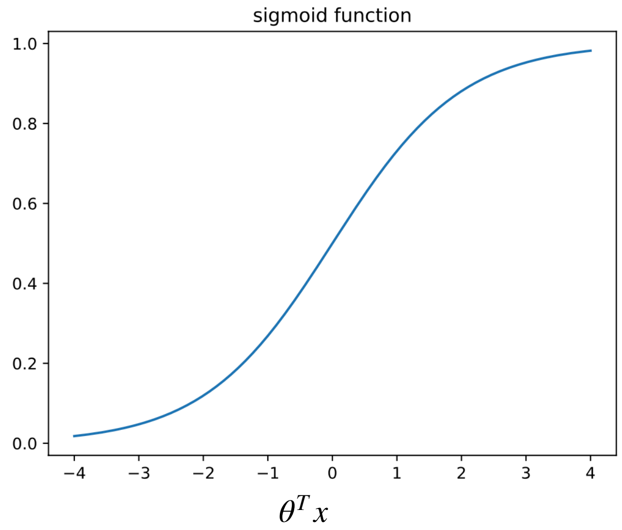
- Logistic Regression: $0 \leq h_\theta(x) \leq 1$

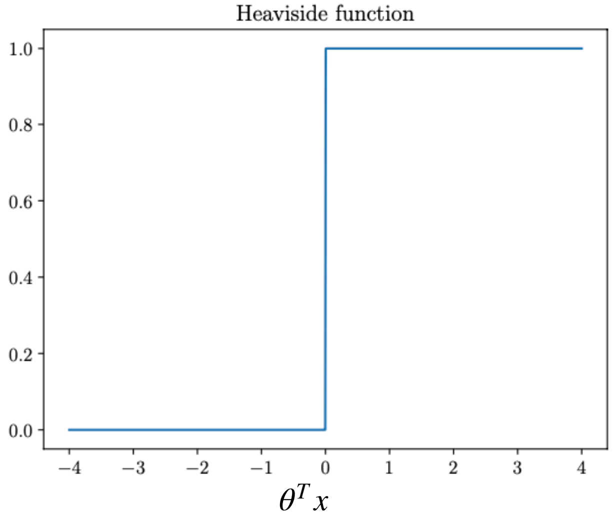
Logistic function

- We want $0 \leq h_\theta(x) \leq 1$
- We do this using the logistic function

$$h_\theta(x) = g(w^T x + b) = g(\theta^T x)$$

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





Interpretation of Hypothesis Output

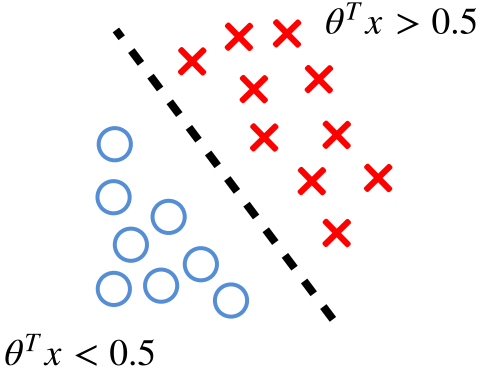
$h_\theta(x)$ = estimated probability that y = 1 on input x

Example: If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$

$$h_\theta(x) = 0.7$$

Tell patient that 70% chance of tumor being malignant

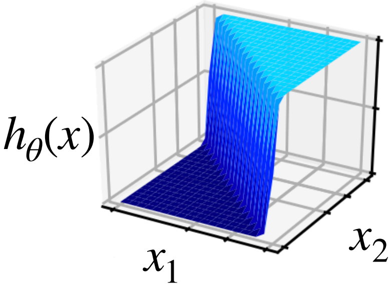
Decision Boundary



$$h_\theta(x) = g(\theta^T x)$$

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

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



predict " $y = 1$ " if $h_\theta(x) \geq 0.5$

predict " $y = 0$ " if $h_\theta(x) < 0.5$

Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

m examples

$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \qquad x_0 = 1, y \in \{0, 1\}$$

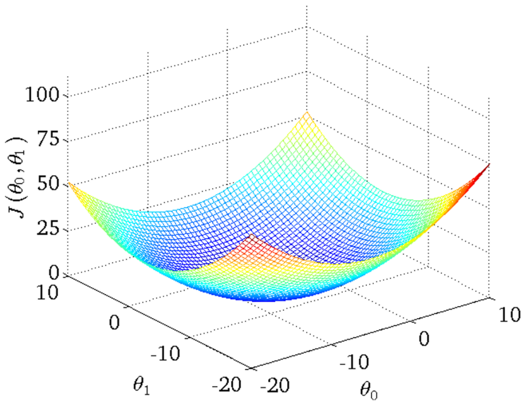
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

How to choose the parameters θ?

Cost function

Linear regression: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

$$\mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$



Logistic regression cost function - Likelihood

Likelihood: probability of observing the data given the model parameters

Likelihood for observing data point i:

$$\text{Ber}\left(y_i \,|\, h_\theta(x_i)\right) = h_\theta(x_i)^{y_i} \times (1 - h_\theta(x_i))^{1-y_i}$$

$$\text{Ber}(y\,|\,\psi) = \begin{cases} 1-\psi & \text{if } y = 0 \\ \psi & \text{if } y = 1 \end{cases}$$

$$\text{Ber}(y\,|\,\psi) = \psi^y(1-\psi)^{1-y}$$

Logistic regression cost function - Likelihood

Likelihood for observing data point i:

$$\text{Ber}\left(y_i \,|\, h_\theta(x_i)\right) = h_\theta(x_i)^{y_i} \times (1 - h_\theta(x_i))^{1-y_i}$$

Likelihood for observing all m data points

$$\prod_{i}^{m} h_\theta(x_i)^{y_i} \times (1 - h_\theta(x_i))^{1-y_i}$$

Logistic regression cost function - Likelihood

Likelihood for observing all m data points

$$\prod_i^m h_\theta(x_i)^{y_i} \times (1 - h_\theta(x_i))^{1-y_i}$$

We usually work with the negative log likelihood

$$\mathrm{NLL}(\theta) = -\frac{1}{m}\sum_i^m \left[y_i \log h_\theta(x_i) + (1 - y_i)\log(1 - h_\theta(x_i))\right]$$

Maximum likelihood

We find the best model parameters $\widehat{\theta}$ by maximizing the NLL

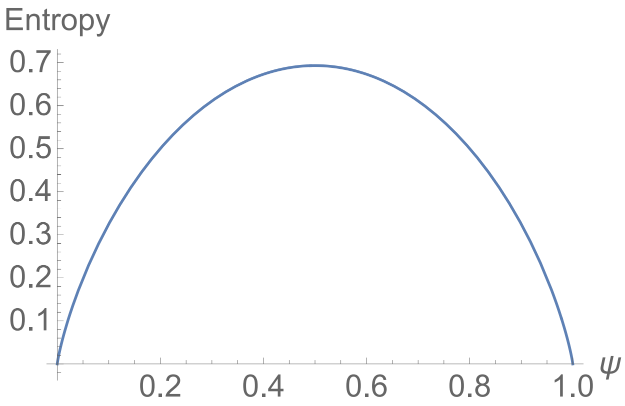
$$\text{NLL}(\theta) = -\frac{1}{m}\sum_{i}^{m}\left[y_i \log h_\theta(x_i) + (1 - y_i)\log(1 - h_\theta(x_i))\right]$$

$$\widehat{\theta} = \text{argmin}_\theta - \frac{1}{m}\sum_{i}^{m}\left[y_i \log h_\theta(x_i) + (1 - y_i)\log(1 - h_\theta(x_i))\right]$$

Negative log likelihood is equal to **binary cross entropy**

- The entropy of a distribution

$$\mathbb{H}(p) = -\sum_x p(x)\log(p(x))$$



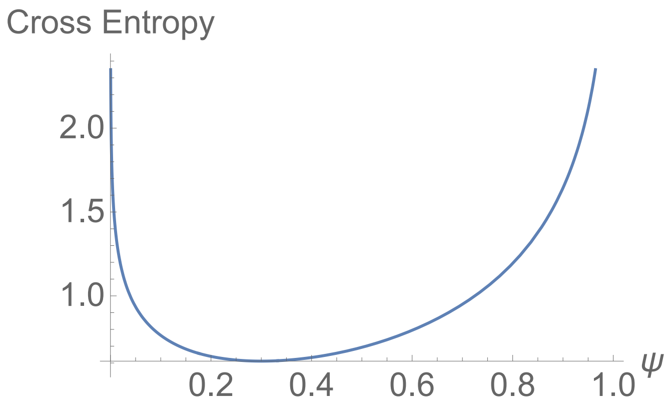
Negative log likelihood is equal to **binary cross entropy**

- The cross entropy between two probability distributions

$$\mathbb{H}(p, q) = -\sum_x p(x)\log(q(x))$$

- Cross entropy for two Bernoulli distributions

$$\mathbb{H}(p, q) = -\left[p\log q + (1-p)\log(1-q)\right]$$



cross entropy of Ber(ψ) relative to Ber(0.3)

Negative log likelihood is equal to **binary cross entropy**

- Equal to the entropy of p and the KL divergence between p and q

$$\mathbb{H}(p, q) = H(p) + D_{KL}(p \| q)$$

$$\begin{aligned}
\mathbb{H}(p, q) &= -\sum_x p(x)\log(q(x)) \\
&= -\sum_x p(x)\big(\log(q(x)) + \log(p(x) - \log(p(x))\big) \\
&= -\sum_x p(x)\log(p(x) - \sum_x p(x)\big(\log(q(x)) - \log(p(x))\big) \\
&= \mathbb{H}(p) + \sum_x p(x)\frac{\log(p(x)}{\log(q(x))}
\end{aligned}$$

$$D_{KL}(p \| q) = \sum_x p(x)\log\left(\frac{p(x)}{q(x)}\right)$$

Negative log likelihood is equal to **binary cross entropy**

- Cross entropy for two Bernoulli distributions

$$\mathbb{H}(p, q) = -\left[p \log q + (1-p)\log(1-q)\right]$$

- Consider the data point i as the distribution p
- Distribution q is the modeled distribution with probability $h_\theta(x_i)$

$$\mathbb{H}(p, q) = -\left[y_i \log h_\theta(x_i) + (1-y_i)\log(1-h_\theta(x_i))\right]$$

$$\text{NLL}(\theta) = -\frac{1}{m}\sum_i^m \mathbb{H}(y_i, h_\theta(x_i))$$

No close form solution exists for the logistic regression cost

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$
$$= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log (1-h_\theta(x^{(i)}))]$$

To fit parameters θ :

$$\min_\theta J(\theta)$$

To make a prediction given new x:

Output $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

Gradient Descent

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Want $\min_\theta J(\theta)$:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

(simultaneously update all θ_j)

$\}$

Gradient Descent

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Want $\min_\theta J(\theta)$:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update all θ_j)

$\}$

Algorithm looks identical to linear regression!

Multiclass case

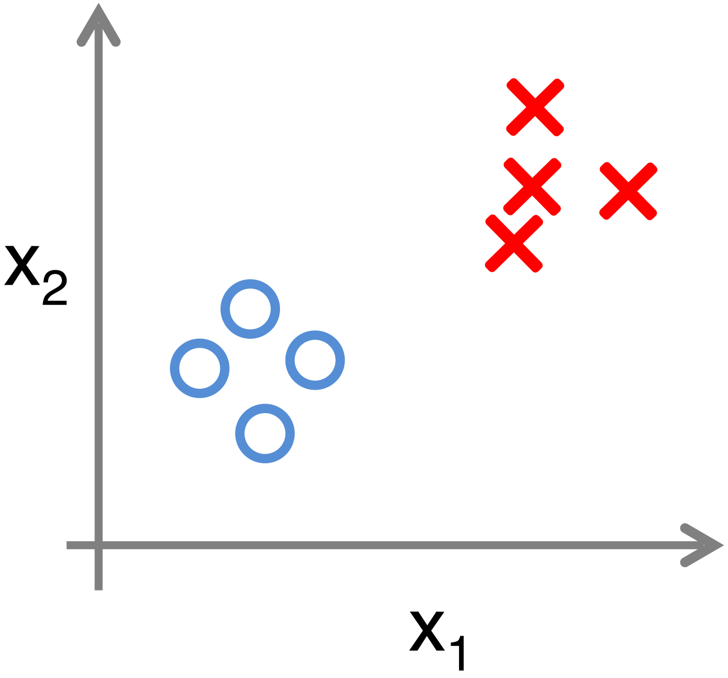
Multiclass classification

Email foldering/tagging: Work, Friends, Family, Hobby

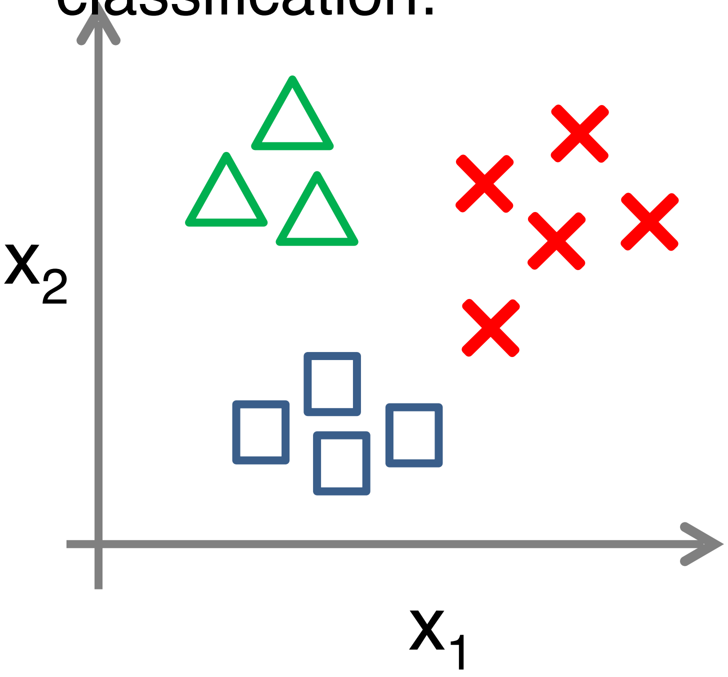
Medical diagrams: Not ill, Cold, Flu

Weather: Sunny, Cloudy, Rain, Snow

Binary classification:



Multi-class classification:



- Recall in logistic regression

$$h_\theta(x) = g(\theta^T x)$$

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

This can be considered as a procedure where we assign two numbers to the two outcomes

$$y = 0 \leftarrow 0$$

$$y = 1 \leftarrow \theta^T x$$

And renormalized such that their sum is one

$$p(y=0) = \frac{e^0}{e^0 + e^{\theta^T x}} \text{ and } p(y=1) = \frac{e^{\theta^T x}}{e^0 + e^{\theta^T x}} = \frac{1}{e^{-\theta^T x} + 1}$$

- The **softmax** function generalizes the logistic function

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \,.$$

- We can build a model for multi-label classification using the softmax function

$$a = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{C-1} \end{bmatrix} = \begin{bmatrix} 0 \\ w_1^T x \\ \vdots \\ w_{C-1}^T x \end{bmatrix} = Wx$$

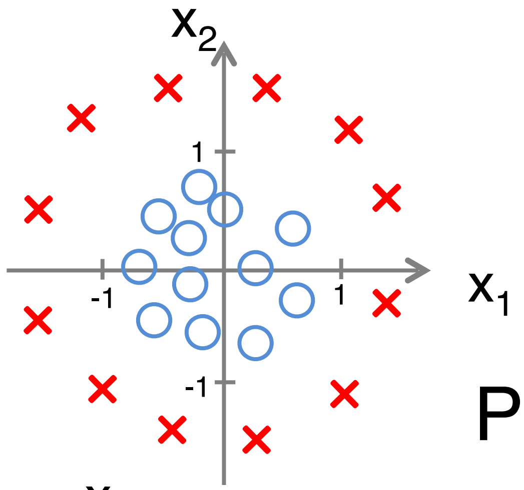
W is the $C \times D$ dimensional weight matrix

x is the D dimensional input vector

- And model the output probability using

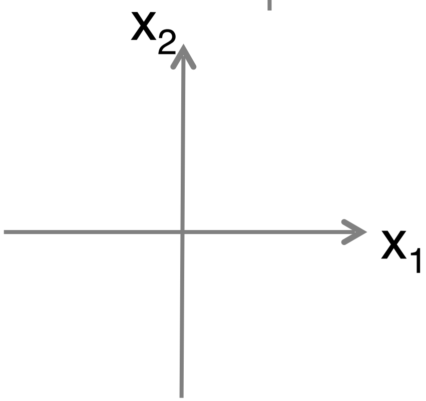
$$p(y = c|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{e^{a_c}}{\sum_{c'=1}^{C} e^{a_{c'}}}$$

Non-linear decision boundaries



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \ldots)$$