Logistic Regression and Regularization

Regression vs Classification

Having worked on regression problems for a while, we now move on to classification problems.

Regression Problems => Real valued output in continuous range

Classification Problems => Discrete valued output in categories

Binary Classification

In binary classification, there are only 2 categories or classes, and This classes are usually represented as 0 or 1.

The model is therefore expected to produce an output value that is either 0 or 1.

When the number of classes is more than 2 the classification problem is termed **Multiclass classification**.

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

In binary classification, often times depending on the nature of the problem;

0 is taken as the ‘negative class’ and 1 the ‘positive class’, or
0 is taken as the ‘false class’ and 1 the ‘true class’, or
0 is taken as the ‘no class’ and 1 the ‘yes class’

You are free to use any representation whatsoever.

How do we tackle the binary classification problem ?

We all know how linear regression works right ?

Let’s use our knowledge of linear regression and build an intuition from that, taking note that in this case, our output is either 0 or 1.

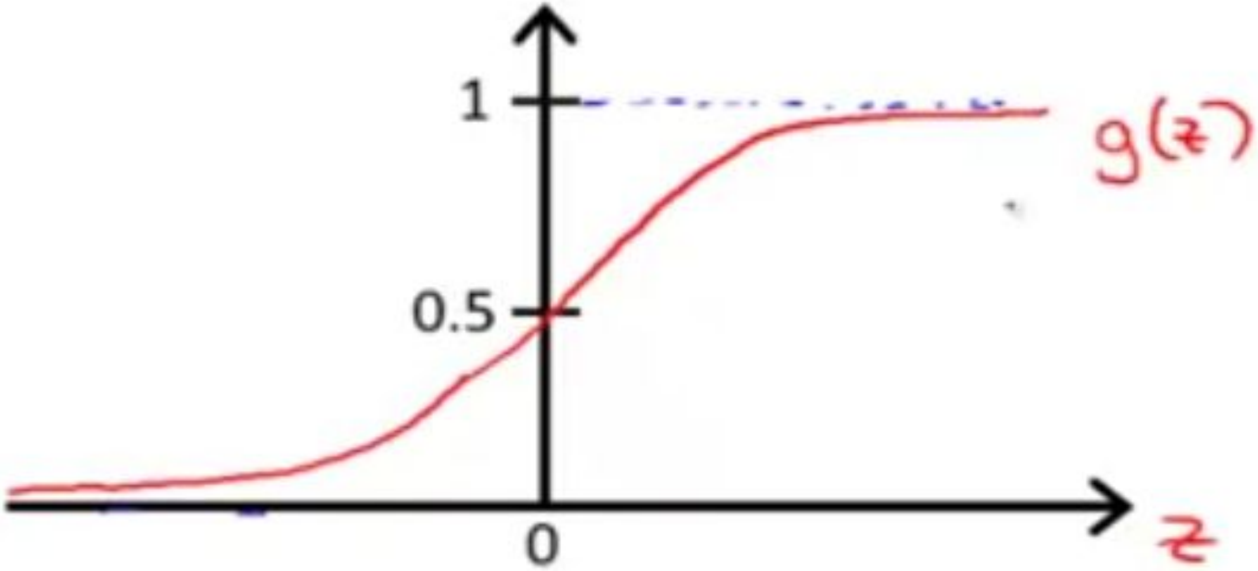
Hypothesis Representation

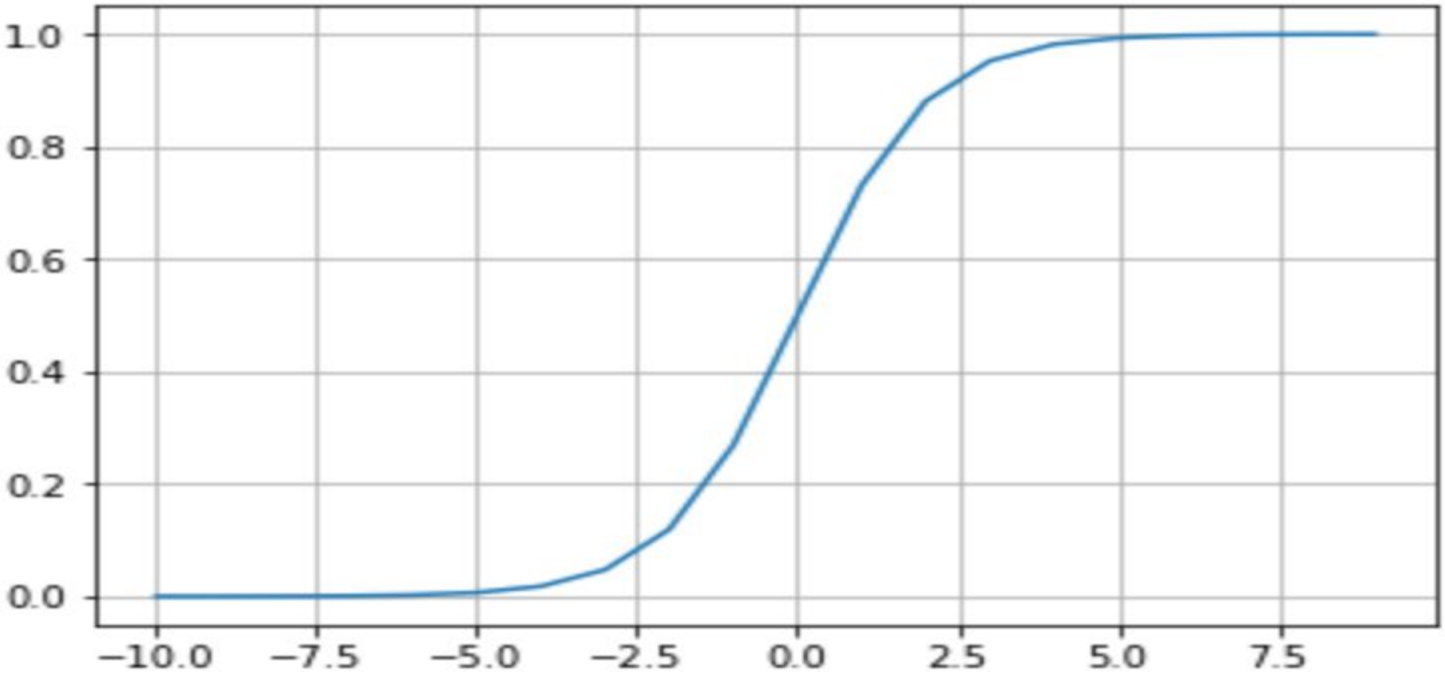
Since the target label is either 0 or 1, our hypothesis has to lie in that range;
$0 \leq h_\theta(x) \leq 1$ (this is what we want)

Linear regression hypothesis is gives as
$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n = \boldsymbol{\theta}^T \mathbf{x}$

And it varies as;
$-\infty \leq h_\theta(x) \leq -\infty$

So, how do we achieve this; $0 \leq h_\theta(x) \leq 1$


1
0.5
0
g(z)
z


1.0
0.8
0.6
0.4
0.2
0.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5

We can do this by passing $\boldsymbol{\theta}^{\top}\mathbf{x}$ into a function, g(z) called ‘Sigmoid function’ or ‘Logistic function’

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

This function takes in whatever input you pass into it and produces an output between 0 and 1.

This is now the hypothesis for logistic regression

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

And,

$$z = \theta^T x$$

The hypothesis is ranges between 0 and 1, and can be modelled as the probability that an output is 1.

E.g, $h_\theta(x) = 0.7$ is a 70% chance that our output is 1.

Therefore,

$$h_\theta(x) = P(y=1|x;\theta) = 1 - P(y=0|x;\theta)$$

And,

$$P(y=0|x;\theta) + P(y=1|x;\theta) = 1$$

Decision Boundary

The hypothesis gives values ranging from 0 to 1. In order to obtain the class of the prediction, we can establish a threshold in which values greater than the threshold are rounded up as 1 and lesser values are 0. We can choose 0.5

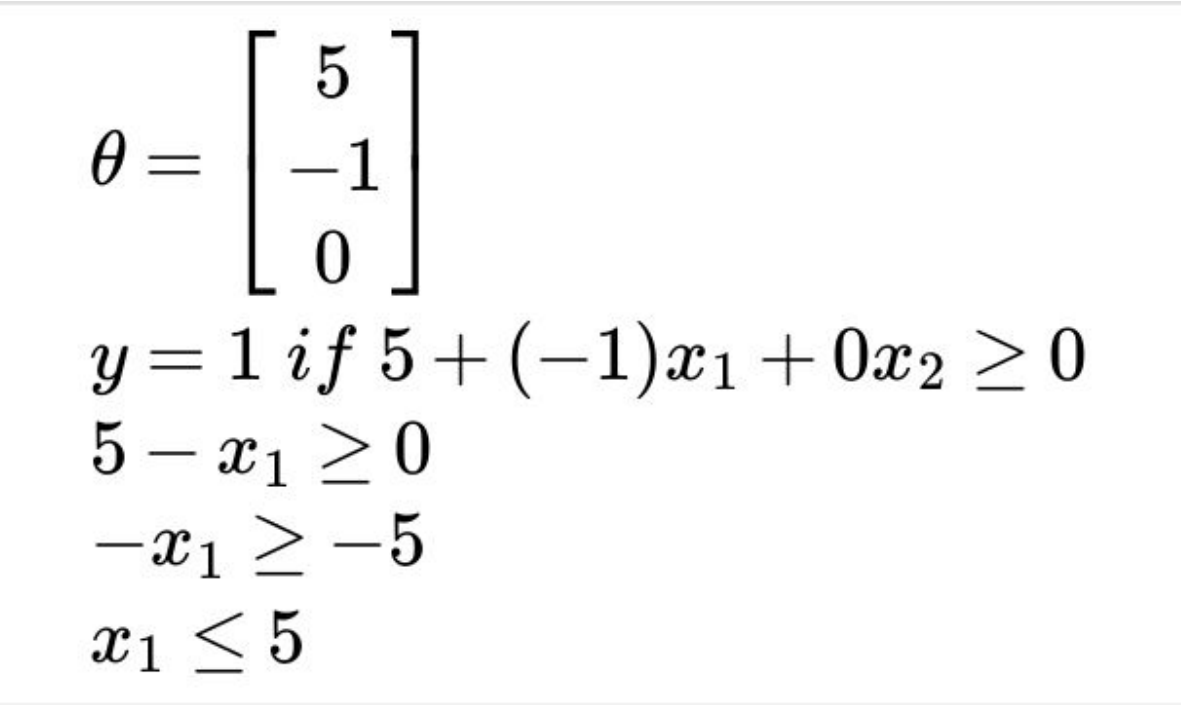
$$h_\theta(x) \geq 0.5 \rightarrow y = 1$$
$$h_\theta(x) < 0.5 \rightarrow y = 0$$

The logistic/sigmoid function has a very interesting behaviour. When its input is greater than or equal to 0, its output is greater than or equal to 0.5, and when its input is less than 0, its output is less than 0.5

$$g(z) \geq 0.5$$
$$when\ z \geq 0$$

$$\theta^T x \geq 0 \Rightarrow y = 1$$
$$\theta^T x < 0 \Rightarrow y = 0$$

The decision boundary is a line that separates the areas y=1 and y=0



$$\theta = \begin{bmatrix} 5 \\ -1 \\ 0 \end{bmatrix}$$

$$y = 1\ if\ 5 + (-1)x_1 + 0x_2 \geq 0$$

$$5 - x_1 \geq 0$$

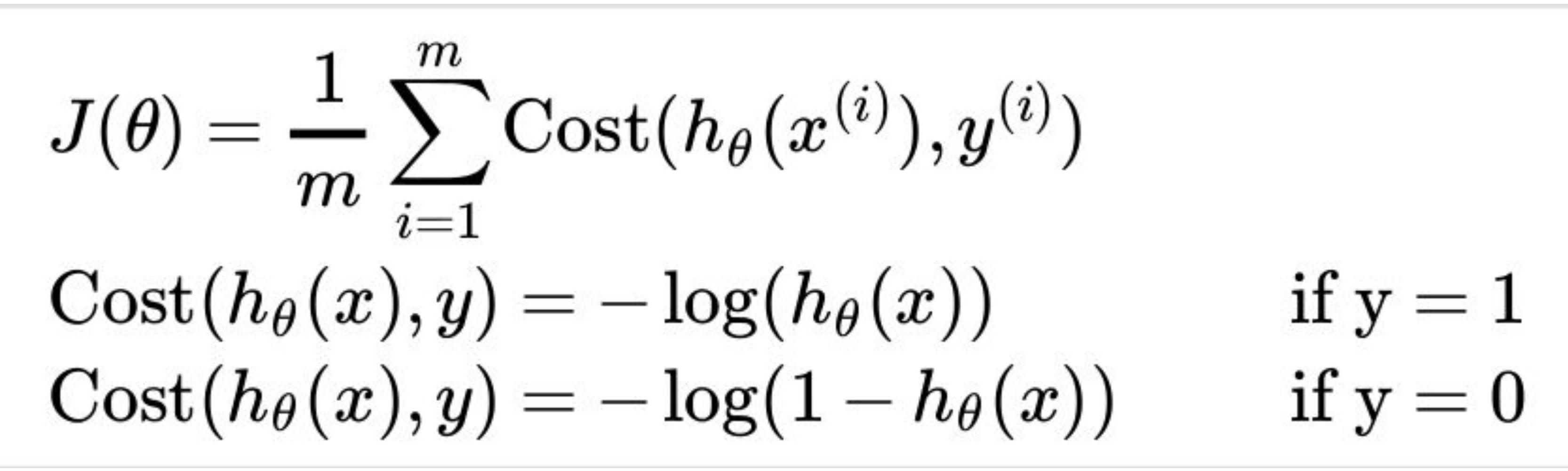
$$-x_1 \geq -5$$

$$x_1 \leq 5$$

Cost Function

What cost function should we use ?

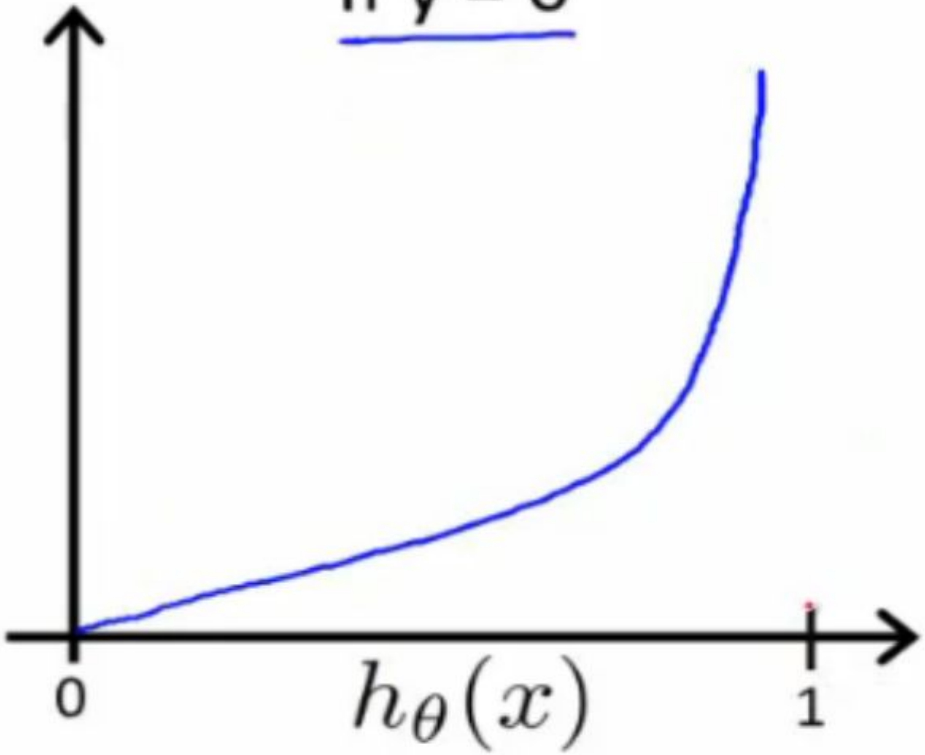
Should we use MSE as in Linear regression ?



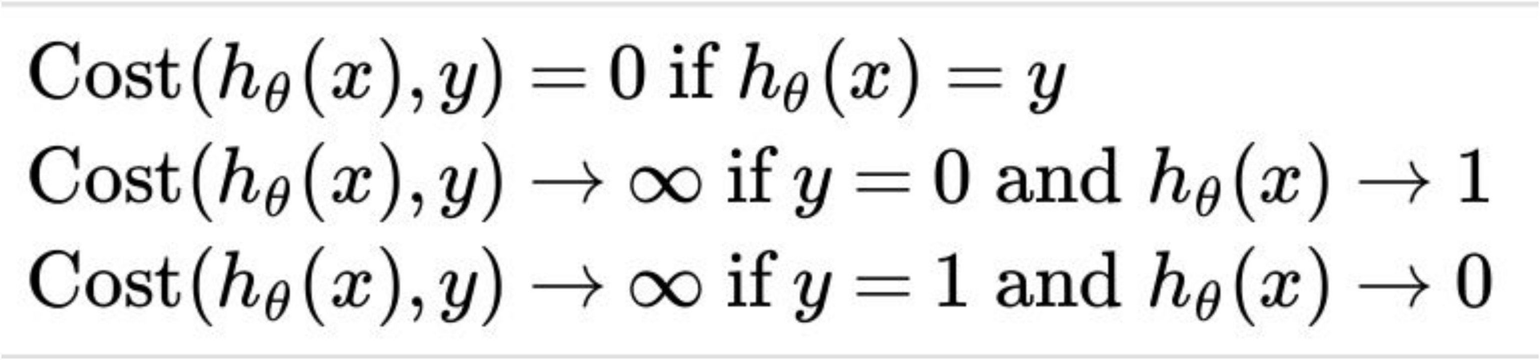
$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\begin{aligned} &\mathrm{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) && \text{if y} = 1 \\ &\mathrm{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) && \text{if y} = 0 \end{aligned}$$

If y = 0



If our hypothesis is far from y, the cost tends to infinity.
If our hypothesis is exactly the same as y, the cost is zero.



$$\mathrm{Cost}(h_\theta(x), y) = 0 \text{ if } h_\theta(x) = y$$
$$\mathrm{Cost}(h_\theta(x), y) \to \infty \text{ if } y = 0 \text{ and } h_\theta(x) \to 1$$
$$\mathrm{Cost}(h_\theta(x), y) \to \infty \text{ if } y = 1 \text{ and } h_\theta(x) \to 0$$

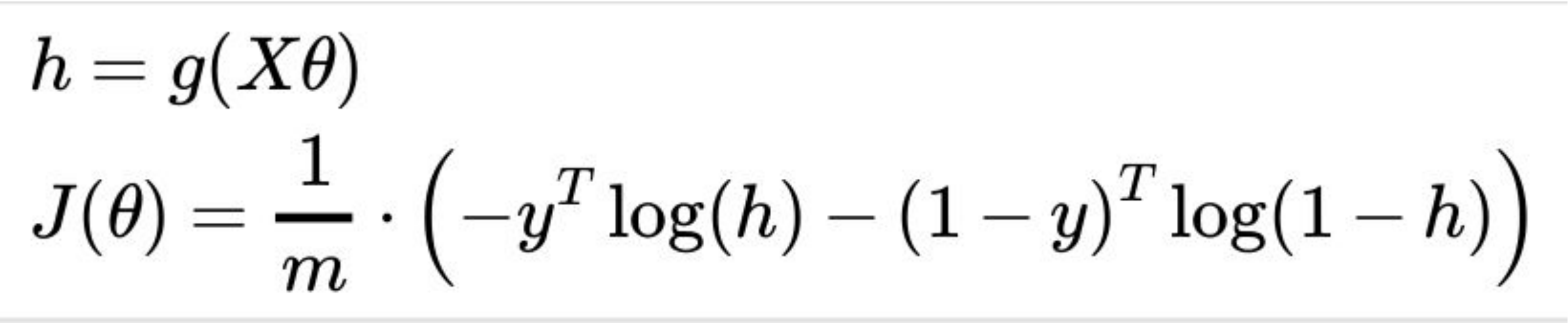
Simplified Cost Function and Gradient Descent

We can write the cost function’s conditional cases in a more compact form

$$\mathrm{Cost}(h_\theta(x), y) = -y \ \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x))$$

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

Vectorized implementation

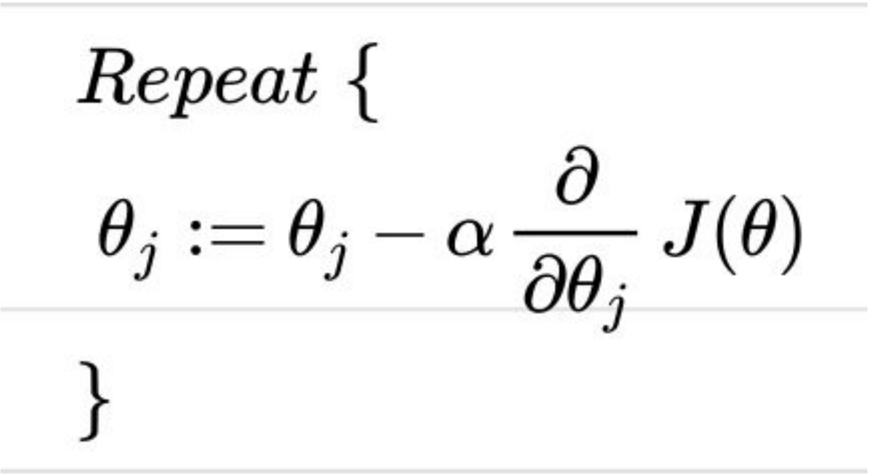


$$h = g(X\theta)$$

$$J(\theta) = \frac{1}{m} \cdot \left(-y^T \log(h) - (1-y)^T \log(1-h)\right)$$

Gradient Descent

The general form of gradient descent is:

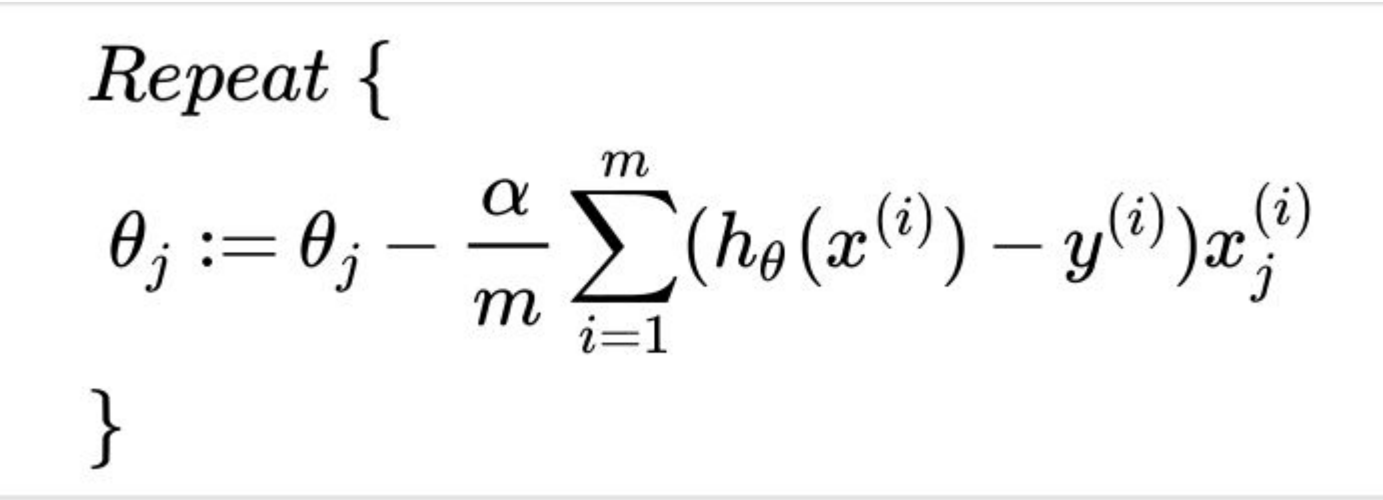


$$Repeat\ \{$$

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

$$\}$$

Working out the partial derivative, we get,



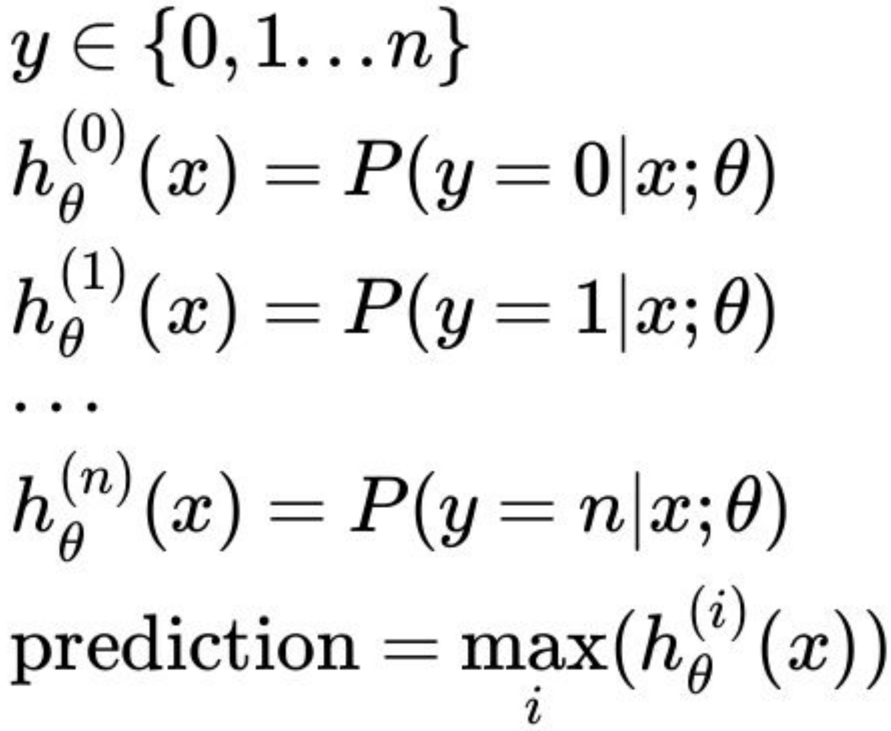
$$
\begin{aligned}
&Repeat\ \{\\
&\quad \theta_j := \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\\
&\}
\end{aligned}
$$

And a vectorized implementation gives,

$$\theta := \theta - \frac{\alpha}{m} X^T \left(g(X\theta) - \vec{y} \right)$$

Multiclass Classification: One-vs-all

y ∈ {0,1, … , n}



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

$$h_\theta^{(0)}(x) = P(y = 0 | x; \theta)$$

$$h_\theta^{(1)}(x) = P(y = 1 | x; \theta)$$

$$\cdots$$

$$h_\theta^{(n)}(x) = P(y = n | x; \theta)$$

$$\text{prediction} = \max_i (h_\theta^{(i)}(x))$$

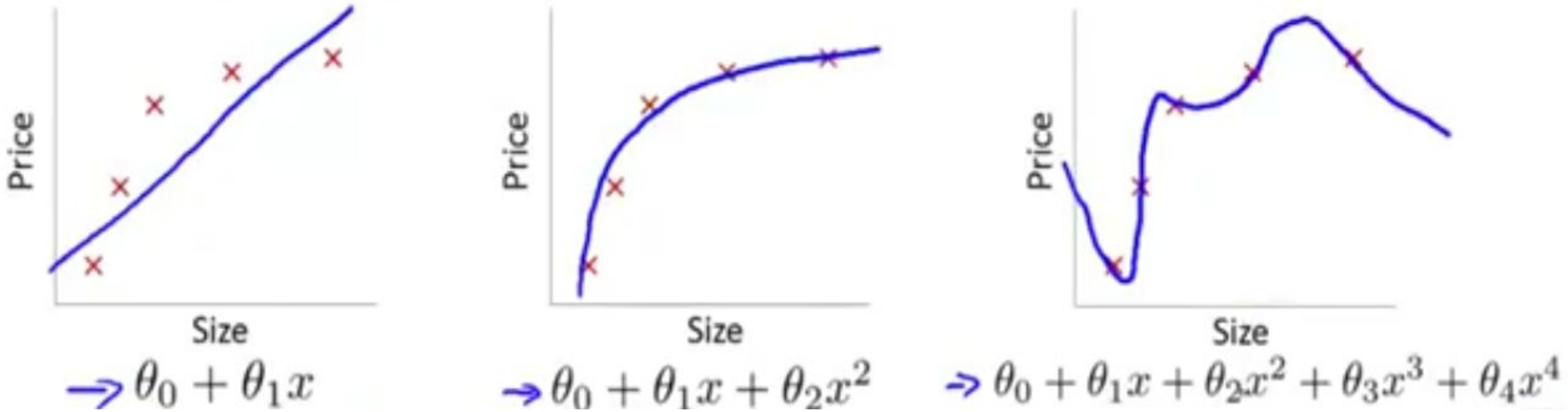
Softmax Regression

Softmax function:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

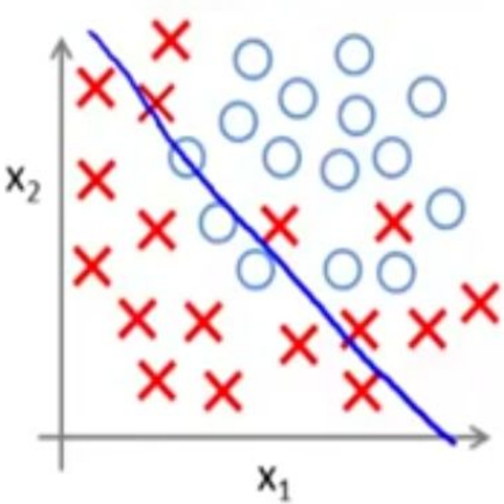
Regularization

Regularization is designed to address the problem of overfitting.



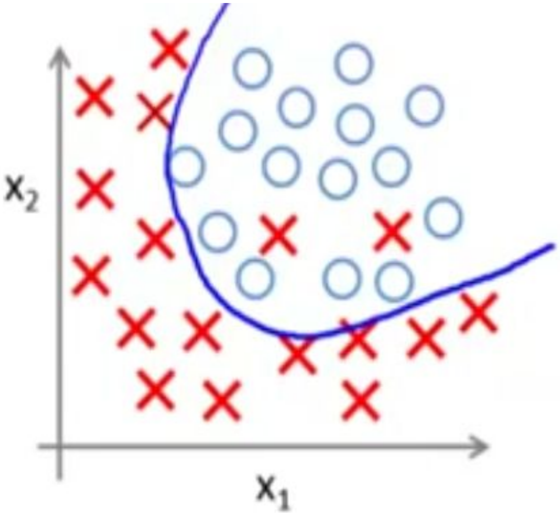
Causes of Overfitting:

Too many features

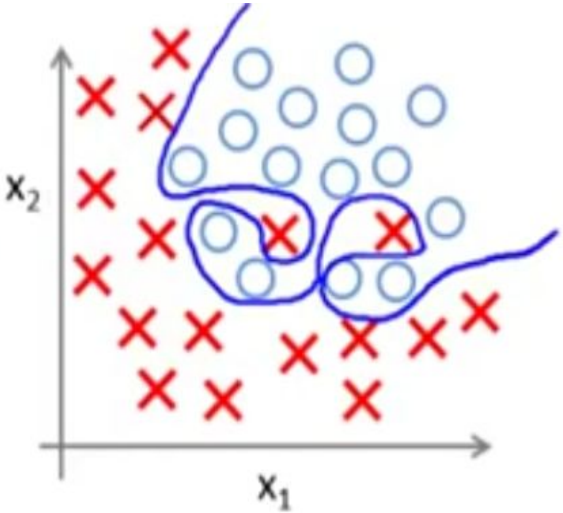


$$h_\theta(x) = g(\underline{\theta_0 + \theta_1 x_1 + \theta_2 x_2})$$

(g = sigmoid function)

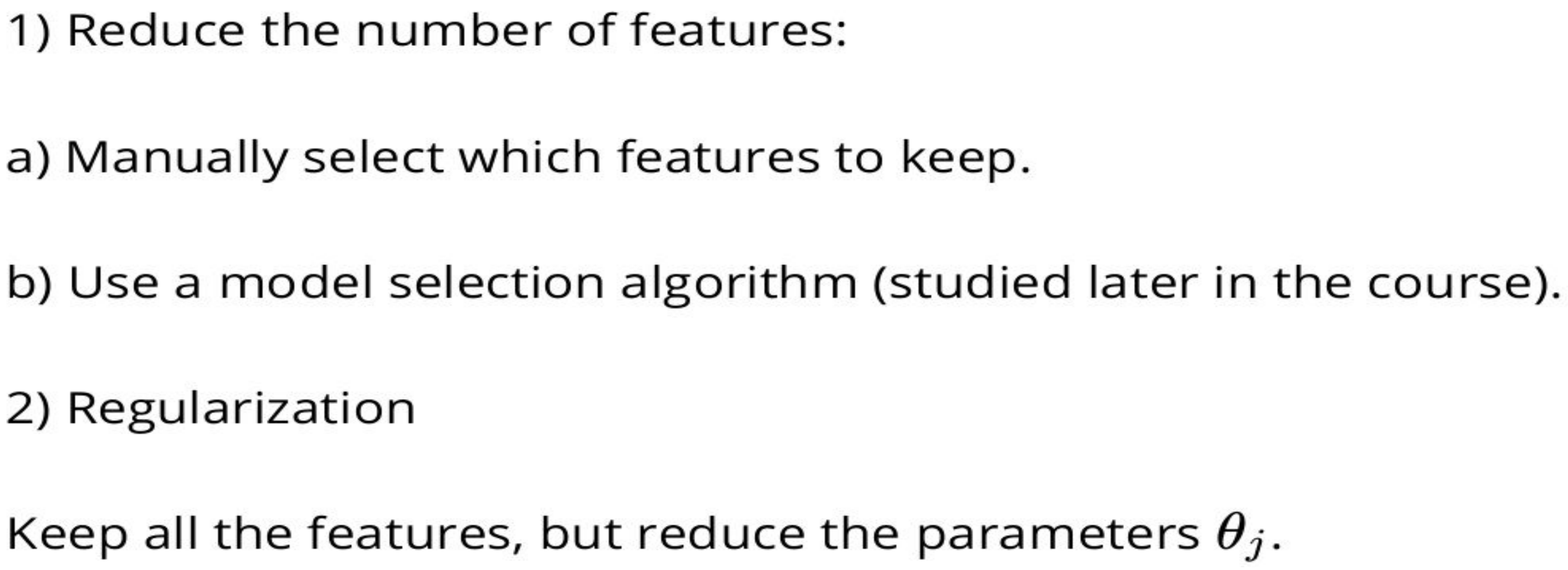


$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ +\theta_3 x_1^2 + \theta_4 x_2^2 \\ +\theta_5 x_1 x_2)$$



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 \\ +\theta_3 \underline{x_1^2 x_2} + \theta_4 \underline{x_1^2 x_2^2} \\ +\theta_5 \underline{x_1^2 x_2^3} + \theta_6 \underline{x_1^3 x_2} + \dots)$$

Addressing overfitting:



1) Reduce the number of features:

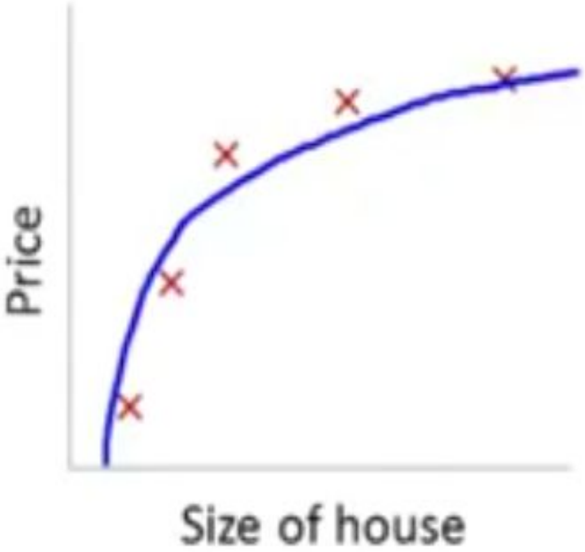
a) Manually select which features to keep.

b) Use a model selection algorithm (studied later in the course).

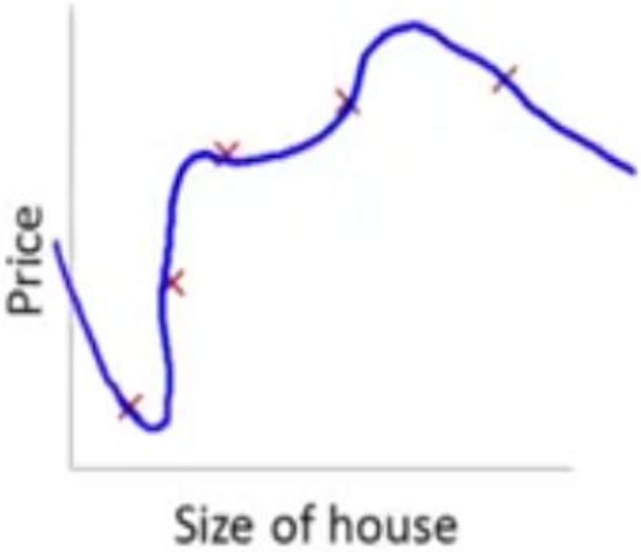
2) Regularization

Keep all the features, but reduce the parameters θ_j.

Cost Function

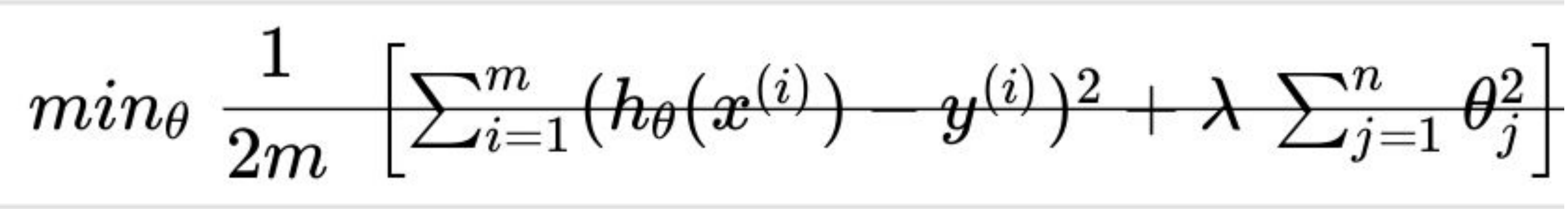


$$\theta_0 + \theta_1 x + \theta_2 x^2$$



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

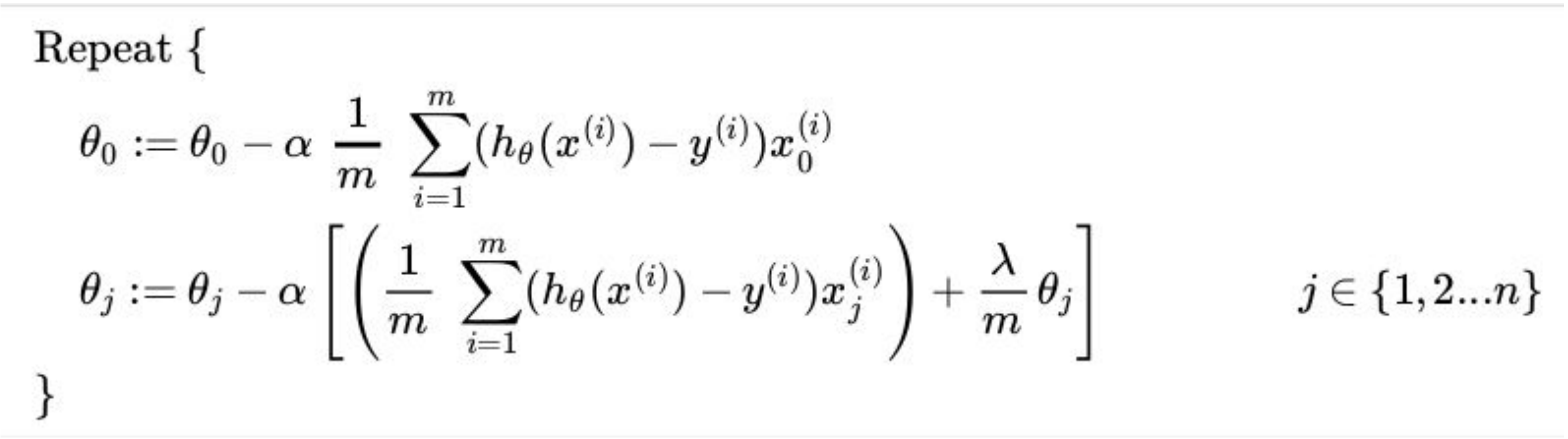
Optimization objective:



$$min_{\theta}\ \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_{\theta}(x^{(i)})-y^{(i)}\right)^{2}+\lambda\sum_{j=1}^{n}\theta_{j}^{2}\right]$$

Regularized Linear Regression

Gradient Descent:



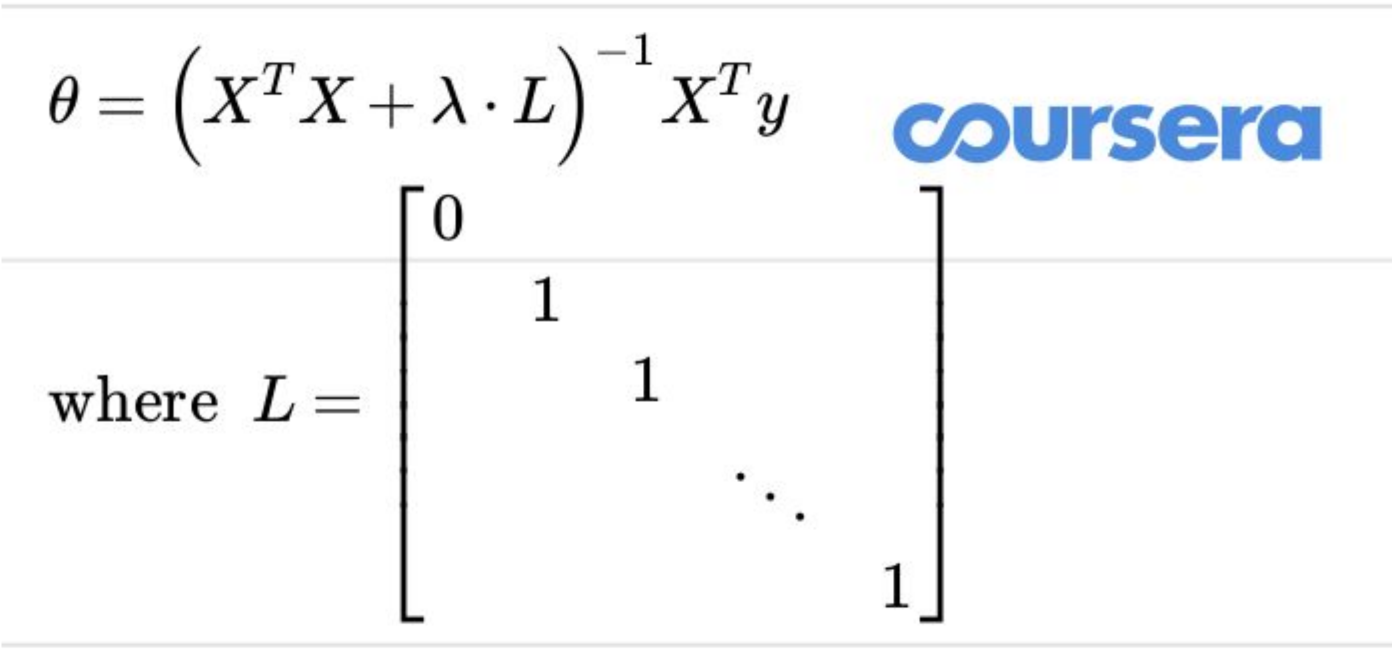
Repeat {

$$\theta_0 := \theta_0 - \alpha \ \frac{1}{m} \ \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \left[\left(\frac{1}{m} \ \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \right) + \frac{\lambda}{m} \theta_j \right] \qquad j \in \{1, 2...n\}$$

}

Normal Equation:



$$\theta = \left(X^T X + \lambda \cdot L\right)^{-1} X^T y$$

$$\text{where } L = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$$

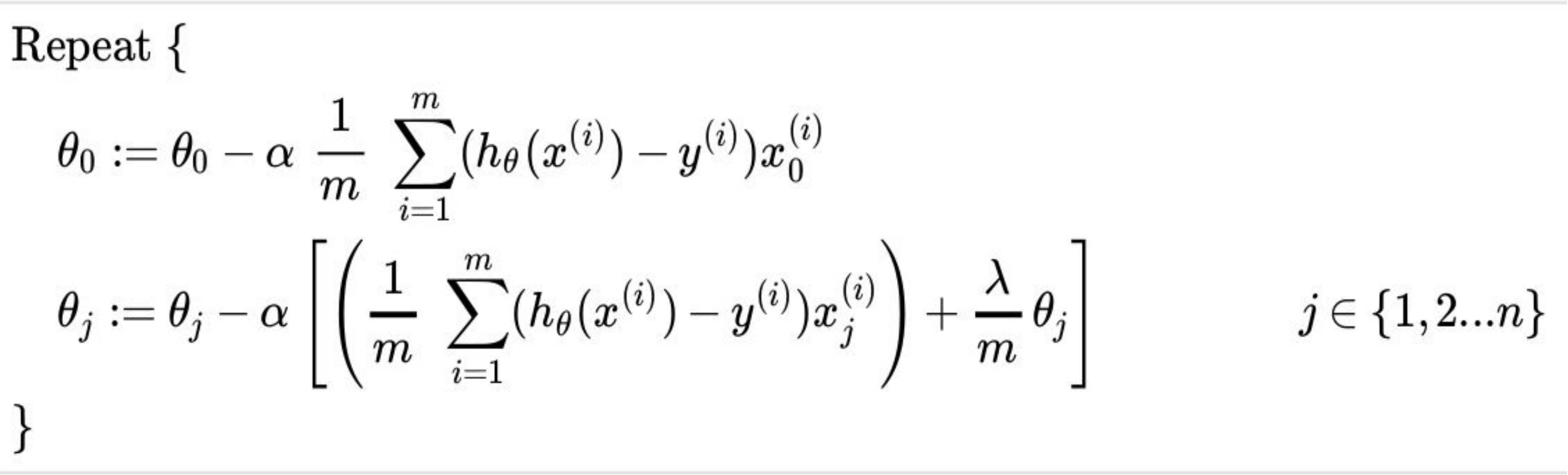
Regularized Logistic Regression

Cost function:

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

The cost function is regularized by adding a term to the end

Gradient Descent:



Repeat $\{$

$$\theta_0 := \theta_0 - \alpha\ \frac{1}{m}\ \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha\left[\left(\frac{1}{m}\ \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j\right] \qquad\qquad j \in \{1, 2...n\}$$

$\}$