FEATURE SCALING
&
LOGISTIC REGRESSION

DATA ANALYSIS @ IITPKD

SECTION 1: Feature Scaling & Methods

DEF^{N}

[wiki:]

Feature scaling is a method used to **standardize** the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step.

MOTIVATION

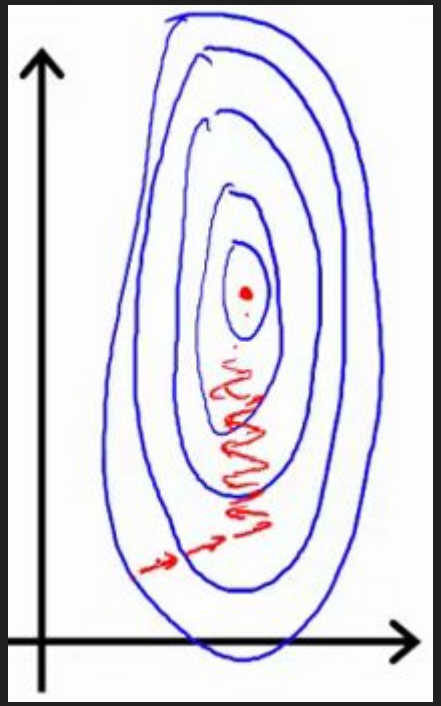
[wiki:]

Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization. For example, the majority of classifiers calculate the distance between two points by the **Euclidean distance**. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.

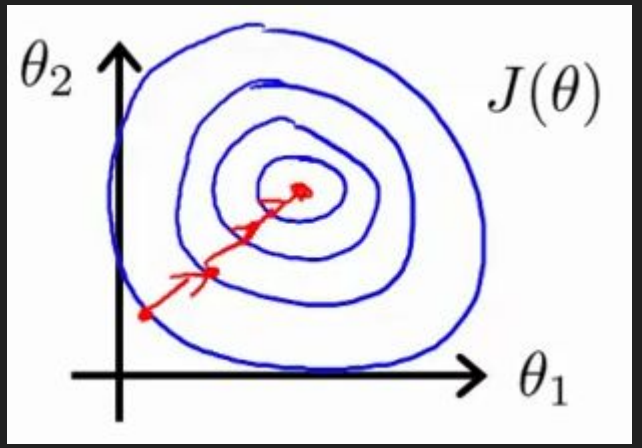
Another reason why feature scaling is applied is that gradient descent converges much faster with feature scaling than without it.

GRADIENT DESCENT

WITHOUT FEATURE SCALING



WITH FEATURE SCALING



[Img. Src: Machine Learning by Andrew Ng(Coursera)]

GRADIENT DESCENT

With feature scaling gradient descent tends to converge faster.

Without feature scaling the contour of the cost function looks little bit like ellipse therefore the gradient step will some time undergo an oscillatory (kind of) step until it reaches the convergence point.

CONCLUSION:

Always use feature scaling before applying gradient descent.

RESCALING

The simplest method is rescaling the range of features to scale the range in [0, 1].

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Where , x → original value & x' → rescaled value

min(x) → minimum among all values of x

max(x) → maximum among all values of x

MEAN NORMALISATION

Final Range : [-1, 1]

$$x' = \frac{x - \text{mean}(x)}{\max(x) - \min(x)}$$

Where , x → original value & x’ → rescaled value

mean(x) → mean of all values of x

min(x) / max(x) →minimum / maximum among all values of x

STANDARDIZATION

Feature standardization makes the values of each feature in the data have **zero-mean** (when subtracting the mean in the numerator) and **unit-variance**. This is typically done by calculating **standard scores(*z*)**:-

$$z = \frac{x - \mu}{\sigma}$$

μ → mean & **σ** → standard deviation

The general method of calculation is to determine the distribution mean and standard deviation for each feature. Next we subtract the mean from each feature. Then we divide the values (mean is already subtracted) of each feature by its standard deviation.

x' → standardized feature vector
x → original feature vector

$$x' = \frac{x - \bar{x}}{\sigma}$$

SECTION 2: Binary Classification

DEF^N

[wiki:]

Binary or **binomial classification** is the task of classifying the elements of a given set into two groups (predicting which group each one belongs to) on the basis of a classification rule.

E.G.

[wiki:]

Some typical binary classification include:

- Medical testing to determine if a patient has certain disease or not – the classification property is the presence of the disease.
- A "pass or fail" test method or quality control in factories, i.e. deciding if a specification has or has not been met – a Go/no go classification.
- Email is spam or not spam

DEF^N

[wiki:]

In statistics, **logistic regression**, or **logit regression**, or **logit model** is a regression model where the dependent variable (DV) is categorical.

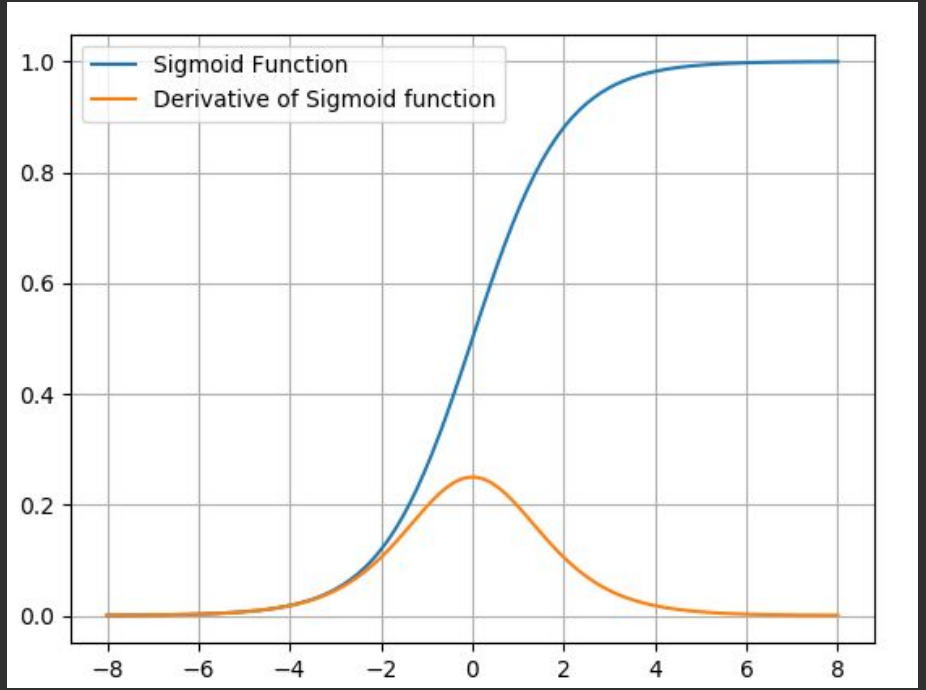
Commonly used Hypothesis Function: “Sigmoid Function” [nxt. slide]

Logistic (Sigmoid) Function

A sigmoid function is a mathematical function having a characteristic "S"-shaped curve or sigmoid curve.

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}.$$

Another Symbol: σ(z)
Derivative : σ(z) (1 - σ(z))



BINARY CLASSIFICATION & LOGISTIC REGRESSION

$\mathbf{h_\theta(z)} = \sigma(z) = \sigma(\theta^T x)$, $z = \theta^T x$

Where, $\theta^T x = \theta_0 + \theta_1 x_1 + \ldots\ldots + \theta_n x_n$

$\therefore$, $\mathbf{h_\theta(z)} = \sigma\left(\theta_0 x_0 + \theta_1 x_1 + \ldots\ldots + \theta_n x_n \right)$, (Remember $x_0 = 1$)

Note that: $0 \leq \sigma(z) \leq 1$

$\therefore$, $\mathbf{0 \leq h_\theta(z) \leq 1}$

Note: we will denote sigmoid function interchangeably by *σ(z) or σ or g(z).* ***Symbol*** *‘σ’ shouldn’t be confused with std. Deviation ‘σ’. In the situation when both of them appear together we will make it clear which one is sigmoid or std. Deviation. Furthermore you can also judge from the context as when we are discussing about function then it is Sigmoid while in the context like Feature scaling it is std. Deviation. It’s a bit unfortunate that we don’t have different symbol for them.*

BINARY CLASSIFICATION & LOGISTIC REGRESSION

In binary Classification we have two classes for eg.,

- Pass or failed
- Spam or not spam
- Cat or dog
- Whether cancer is malignant or benign

We will denote them by { 0 , 1 }.

∴ our output ‘$y^{(i)}$’ can only take values : 0 or 1

BINARY CLASSIFICATION & LOGISTIC REGRESSION

In this situation $\mathbf{h_\theta(z)}$ will gives the probability that our output is 1.

$\mathbf{h_\theta(z) = P(y=1 \mid x,\theta) = 1 - P(y=0 \mid x,\theta)}$

Whereas , $(1-\mathbf{h_\theta(z)})$ gives the probability that our output is 0.

Our probability that our prediction is 0 is just the complement of our probability that it is 1 (e.g. if probability that it is 1 is 70%, then the probability that it is 0 is 30%).

PREDICTION

When $h_\theta(z) \geq 0.5 \Rightarrow y = 1$

$h_\theta(z) < 0.5 \Rightarrow y = 0$

$h_\theta(z) = \sigma(\theta^T x) \geq 0.5 \Rightarrow \theta^T x \geq 0 \Rightarrow y = 1$

$h_\theta(z) = \sigma(\theta^T x) < 0.5 \Rightarrow \theta^T x < 0 \Rightarrow y = 0$

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

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

DECISION BOUNDARY

The decision boundary is the line that separates the area where y = 0 and where y = 1.

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

$$\theta^T x < 0 \quad \Rightarrow \quad y = 0$$

The input to the sigmoid function $\sigma(z)$ (in this context: $\theta^T x$) doesn't need to be linear, and could be a function that describes a circle.

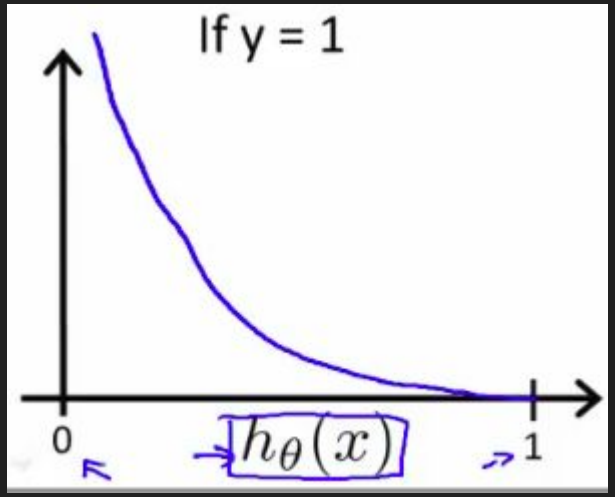
COST FUNCTION FOR BINARY CLASSIFICATION:

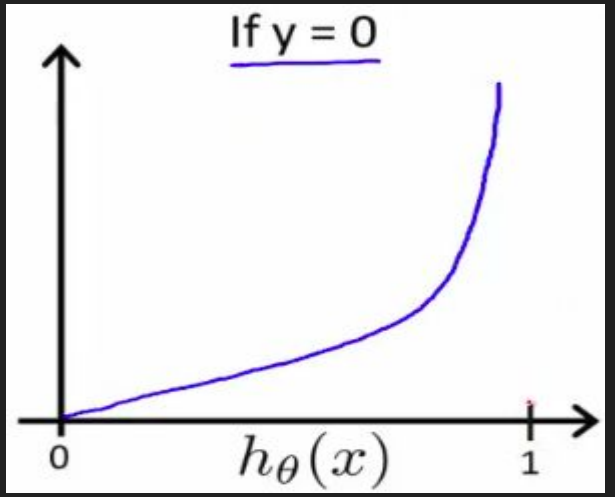
$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

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

$$\text{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \qquad \text{if y} = 0$$

COST FUNCTION FOR BINARY CLASSIFICATION:





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

COST FUNCTION FOR BINARY CLASSIFICATION:

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

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

[Note: Before Proceeding to next slide try to find a vectorized implementation for the above formula.]

COST FUNCTION FOR BINARY CLASSIFICATION:

Vectorized Implementation :

Here, ***X*** = design matrix or feature matrix

y = column vector containing : $y^{(1)}$, $y^{(2)}$, ……. , $y^{(m)}$

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

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

Note : Now you know the Cost Function for binary classification you can try finding out the gradient descent algorithm for this.

Thanks!

Kaushal Kishore (111601008)
Amit Vikram Singh (111601001)
Sai Suchith Mahajan
(121601016)