





Machine Learning for Cyber Security (CS-602) L#07

Logistic Regression

By
Dr Sunita Dhavale

Syllabus

- Data Analytics Foundations: R programming, Python Basics -Expressions and Variables, String Operations, Lists and Tuples, Sets, Dictionaries Conditions and Branching, Loops, Functions, Objects and Classes, Reading/Writing files, Handling data with Pandas, Scikit Library, Numpy Library, Matplotlib, scikit programming for data analysis, setting up lab environment, study of standard datasets. Introduction to Machine Learning- Applications of Machine Learning, Supervised, unsupervised classification and regression analysis
- Python libraries suitable for Machine Learning Feature Extraction. Data pre-processing, feature analysis etc., Dimensionality Reduction & Feature Selection Methods, Linear Discriminant Analysis and Principal Component Analysis, tackle data class imbalance problem

Syllabus

- Supervised and regression analysis, Regression, Linear Regression, Non-linear Regression, Model evaluation methods, Classification, K-Nearest Neighbor, Naïve Bayes, Decision Trees, Logistic Regression, Support Vector Machines, Artificial Neural Networks, Model Evaluation. Ensemble Learning, Convolutional Neural Networks, Spectral Embedding, Manifold detection and Anomaly Detection
- Unsupervised classification K-Means Clustering, Hierarchical Clustering, Density-Based Clustering, Recommender Systems-Content-based recommender systems, Collaborative Filtering, machine learning techniques for standard dataset, ML applications, Case studies on Cyber Security problems that can be solved using Machine learning like Malware Analysis, Intrusion Detection, Spam detection, Phishing detection, Financial Fraud detection, Denial of Service Detection.

Text/Reference Books

1. Building Machine Learning Systems with Python – Willi Richert, Luis Pedro Coelho
 2. Alessandro Parisi, Hands-On Artificial Intelligence for Cybersecurity: Implement smart AI systems for preventing cyber attacks and detecting threats and network anomalies
Publication date :Aug 2, 2019, Packt, ISBN-13, 9781789804027
 3. Machine Learning: An Algorithmic Perspective – Stephen Marsland
 4. Sunita Vikrant Dhavale, “Advanced Image-based Spam Detection and Filtering Techniques”, IGI Global, 2017
 5. Soma Halder , Sinan Ozdemir, Hands-On Machine Learning for Cybersecurity: Safeguard your system by making your machines intelligent using the Python ecosystem, By
Publication date : Dec 31, 2018, Packt, ISBN-13 :9781788992282
-
1. Stuart Russell, Peter Norvig (2009), “Artificial Intelligence – A Modern Approach”, Pearson Elaine Rich & Kevin Knight (1999), “Artificial Intelligence”, TMH, 2nd Edition
 2. NP Padhy (2010), “Artificial Intelligence & Intelligent System”, Oxford
 3. ZM Zurada (1992), “Introduction to Artificial Neural Systems”, West Publishing Company
 4. Research paper for study (if any) – White papers on multimedia from IEEE/ACM/Elsevier/Spinger/ Nvidia sources.

Lab assignments

1	Python Programming part-1
2	Python Programming part-2
3	Study and Implement Linear Regression Algorithm for any standard dataset like in cyber security domain
4	Study and Implement KMeans Algorithm for any standard dataset in cyber security domain
5	Study and Implement KNN for any standard dataset in cyber security domain
6	Study and Implement ANN for any standard dataset in cyber security domain
7	Study and Implement PCA for any standard dataset in cyber security domain
8	Case Study: Use of ML along with Fuzzy Logic/GA to solve real world Problem in cyber security domain
9	Mini assignment: Apply ML along with PSO/ACO to solve any real world problem in cyber security domain
10	ML Practice Test – 1 Quiz

Defence Institute of Advanced Technology

School of Computer Engineering & Mathematical Sciences

SEMESTER-I TIME TABLE (AUTUMN 2024)[§]

PROGRAMMES: (I) CS [M.TECH IN CYBER SECURITY] (II) AI [M.TECH CSE (ARTIFICIAL INTELLIGENCE)]

BATCH: 2024-2026

Lecture Day	L1 0900-1000	L2 1000-1100	L3 1100-1200	L4 1200-1300		L4 1400-1500	L4 1500-1600	L4 1600-1700	L4 1700-1800
Monday	CE-602 (AI) CS-602 (CS)	CE-604 (AI) CS-603 (CS)	CE-601 (AI) CS-604 (CS)	CE-601 (AI) LAB CS-603 (CS)	Lunch Break 1300-1400	LAB CE-601 (AI) LAB CS-602 (CS)		AM607	
Tuesday	CE-603 (AI) LAB CS-603 (CS)	CE-602 (AI) CS-602 (CS)	CE-601 (AI) CS-605 (CS)	CE-604 (AI) CS-604 (CS)		PGC 601		AM607	
Wednesday	CS-605 (CS)	CE-603 (AI) CS-602 (CS)	CE-602 (AI) CS-603 (CS)	CE-604 (AI) CS-604 (CS)		CE-605(AI) LAB CS-605 (CS)	LAB CS-605 (CS)	AM607	
Thursday	LAB CE-604 (AI) CS-603 (CS)	LAB CE-604 (AI) CS-605 (CS)	LAB CE-602 (AI) CS-601 (CS)	CE-603 (AI) CS-601 (CS)		PGC 601		AM607	
Friday	LAB CE-603 (AI) LAB CS-601 (CS)		LAB CE-602 (AI) CS-601 (CS)	LAB CS-604 (CS)		CE-605(AI) LAB CS-604 (CS)	CE-605(AI)	LAB CE-605(AI)	

COURSE CODE & COURSE NAME		FACULTY
Programme: CS [M.Tech in Cyber Security] Classroom: Arjun	Programme: AI [M.Tech CSE (Artificial Intelligence)] Classroom: Kaveri	
CS-601 Data Security & Privacy	CE-601 Responsible Artificial Intelligence;	MUN: Dr. Manisha J. Nene
CS-602 ML for Cyber Security	CE-604 Practical Machine Learning;	SVD: Dr. Sunita V. Dhavale
CS-605 Network and Cloud Security	CE-602 Intelligent Algorithms	CRS: Prof. CRS Kumar
CS-604 Advanced System Security	-----	DVV: Dr. Deepti V. Vidyarthi
CS-603 Applied Cryptography	-----	AM: Dr. Arun Mishra
-----	CE-603 Deep Neural Network;	US: Dr. Upasna Singh
-----	CE-605 Mathematics for ML;	Unit-2: Dr Upasna, Unit 4: Dr Sunita, Unit3:MIM, Unit 1: Faculty To be Nominated
AM-607 Mathematics for Engineers	AM-607 Mathematics for Engineers	OO/DS/DP: Dr Odellu O., Dr Dasari S., Dr. Debasis P.
PGC-601 Research Methodology	PGC-601 Research Methodology	Common Subject for All

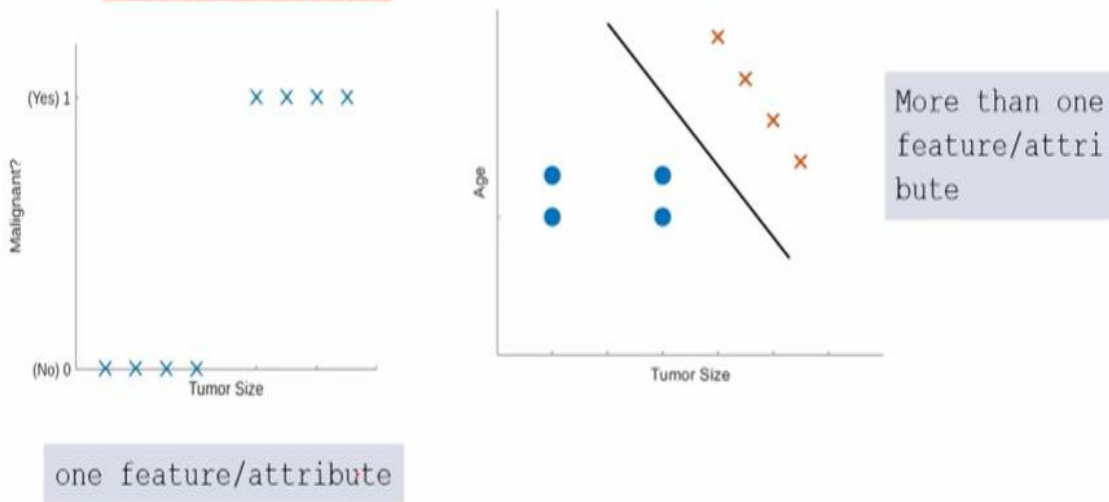
§ TENTATIVE T.T. SUBJECT TO CHANGE

Program Coordinator,
M.Tech (CS & AI), Batch 2024-26

Director, SoCE&MS

Binary Classification Problem – Tumor Malign or not

Classification Problem



How to classify using linear classifier:

If only one feature

X =tumor size

If two features $X(x_1, x_2)$ i.e. tumor size and Age

The different **hypothesis function of linear predictors** are compositions of a function $g: \mathcal{R} \rightarrow \mathcal{Y}$ over the linear function $l_W: \mathcal{R}^n \rightarrow \mathcal{R}$, which is parameterized by w_0 (called the bias), w_1, \dots, w_n and $l_W(X) = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$.

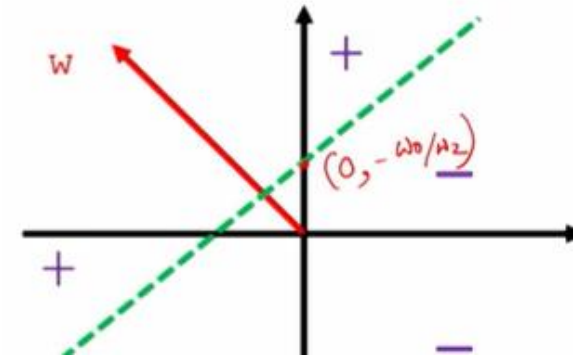
Each function $l_W(X)$ takes as input a vector X and returns as output the scalar.

For example, in binary classification, we can choose g to be the sign function, and for regression problems, where $\mathcal{Y} = \mathcal{R}$, g is simply the identity function.

Half space Hypothesis for Binary Classification problem

The first hypothesis we consider is the Halfspace, designed for binary classification problems, namely, $\mathcal{Y} = \{-1, +1\}$.

- The halfspace is defined as $h_W(X) = g_{\text{sign}} \cdot l_W(X)$, where each half space hypothesis is dependent on w_0 (called the bias), and w_1, \dots, w_n .
- While taking a vector X the hypothesis returns the label $g_{\text{sign}}(w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n)$.



In **Logistic Regression** we will learn a family of functions $h_W: \mathbb{R}^n \rightarrow [0, 1]$. We can interpret $h_W(X)$ as the probability that the label of X is 1.

Logistic Regression

- Logistic regression - models probability and can be used for classification.
- Uses Logistic /Sigmoid function-> S-shaped function that smoothly maps $(-\infty, \infty)$ to $[0, 1]$, but also can be differentiated with ease.
- $f(Z)$ depicts the probability of the value of y being 1, given the data.
- for large negative value, sigmoid returns approximate zero (or very near to zero).
- For large positive value sigmoid returns nearly one value.
- Can be used for binary classification
- doesn't require linear relationship between dependent and independent variables

$$h_W(X) = g_{sig} \cdot l_W(X),$$

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

Logistic Regression

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

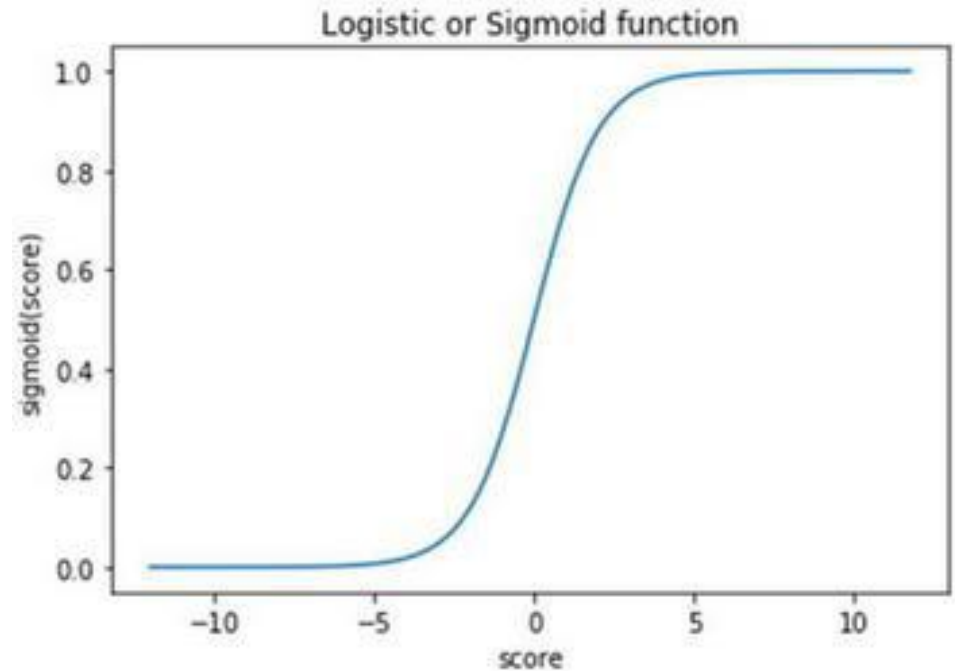
$$\lim_{z \rightarrow -\infty} g(z) \rightarrow 0$$

$$\lim_{z \rightarrow \infty} g(z) \rightarrow 1$$

$$0 \leq g(z) \leq 1 \Rightarrow 0 \leq h_W(X) \leq 1$$

$$h_W(X) = \frac{1}{1 + e^{-W^T X}},$$

$W \rightarrow$ parameter for estimation



$$\frac{dg}{dz} = g'(z) = \frac{e^{-z}}{(1 + e^{-z})^2} = \frac{1}{\underbrace{(1 + e^{-z})}_{g(z)}} \frac{e^{-z}}{\underbrace{(1 + e^{-z})}_{(1-g(z))}}$$

$$\therefore g'(z) = g(z)(1 - g(z))$$

$$\text{prob}(y = \text{positive} \mid x, w) = \frac{1}{\left(1 + e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_d x_d}\right)}$$

In Logistic Regression

hypothesis function is the composition of a Sigmoid function over the linear function

Predictions of Hypothesis Function of LR are similar as Half space Hypothesis when $W^T X$ is very large

$h_W(X)$ is the estimated probability that $y = 1$ on an input X .

Example: If for a new input X , $h_W(X) = 0.7$, tell the patient that there is 70% chance of tumor being malignant.

Mathematically, $h_W(X) = g(W^T X) = P(y = 1|X; W)$, “Probability that $y = 1$ ”, given X with some parameter W .

If $h_W(X) = 0.5$, means approximately 0.5 probability that it is in class 1.

$$P(y = 0|X; W) = 1 - P(y = 1|X; W)$$

Note that when $W^T X$ is very large then $h_W(X)$ is close to 1,

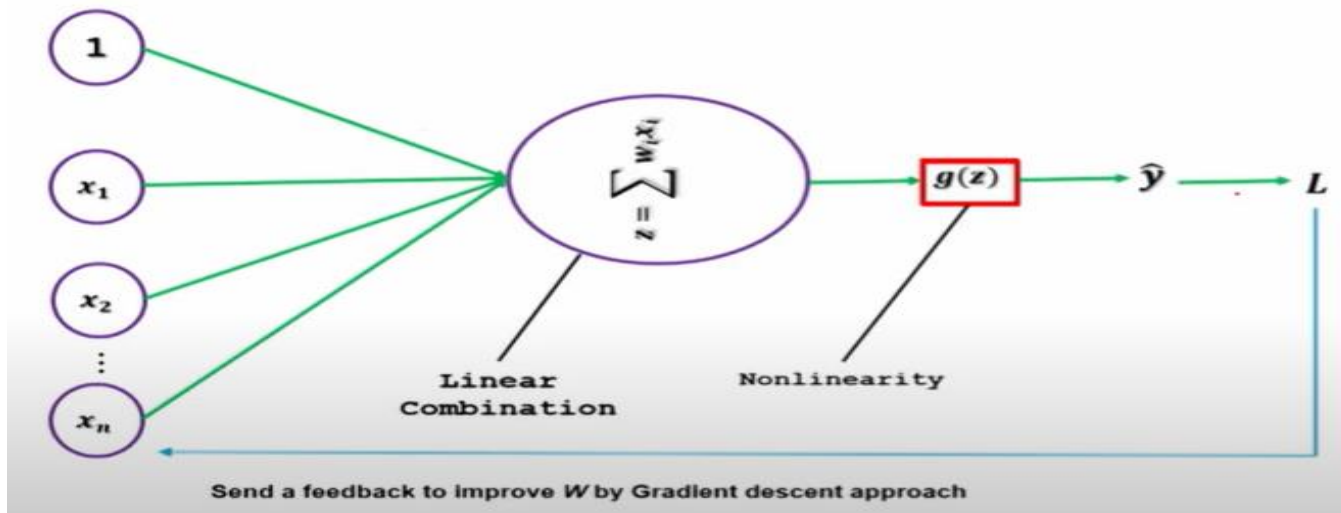
whereas if $W^T X$ is very small then $h_W(X)$ is close to 0.

Recall that the prediction of the half space corresponding to a vector W is $\text{sign}(w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n)$.

However, when $W^T X$ is close to 0 we have that $h_W(X) \approx 1/2$

In contrast, the halfspace hypothesis always outputs a deterministic prediction of either 1 or -1, even if $W^T X$ is very close to 0.

Cross entropy Loss - Desirable properties



$$\hat{y} = \frac{1}{1+e^{-w^T X}} \in (0, 1]$$

if the number is proper fraction, i.e.: it lies between 0 and 1, then logarithm is negative.

loss function for logistic regression – **log loss/Cross entropy Loss**

$$Cost(y, \hat{y}) = L = - \left\{ y \ln \frac{1}{1+e^{-w^T X}} + (1-y) \ln \left(\frac{1}{1+e^{w^T X}} \right) \right\}$$

$$(1-y) = 0 \text{ or } 1 \text{ and } \ln(1-\hat{y}) \rightarrow -ve \Rightarrow (1-y) \ln(1-\hat{y}) \rightarrow -ve$$

Desirable properties:

- $L = 0$ if $y = \hat{y}$
- L should be very high for misclassification
- Consistently $L \geq 0$

$$\ln \hat{y} \rightarrow -ve, y \text{ is either } 0 \text{ or } 1 \Rightarrow y \ln \hat{y} \rightarrow -ve$$

$$\text{So, } L \text{ is } +ve \geq 0$$

Cross entropy Loss - Desirable properties

$$\operatorname{argmin}_{W \in \mathbb{R}^n} - \sum_{i=1}^n \{y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)\}$$

- If you classify correctly, cost function will be approximately 0, i.e., $L = 0$ if $y = \hat{y}$.

For example, if $y = 0$ and \hat{y} is close to 0 then $y \ln \hat{y} \rightarrow 0$ and $(1 - y) \ln(1 - \hat{y}) \rightarrow \text{close to } 0 \Rightarrow L \approx 0$

L should be very high for misclassification.

Let $y = 0$ and $\hat{y} \approx 1$, $y \ln \hat{y} = 0$ as $y = 0$

$(1 - y) = 1$ as $y = 0$ and $\hat{y} \approx 1 \Rightarrow \ln(1 - \hat{y}) \rightarrow -\infty \Rightarrow L \rightarrow \infty$

Also, if $y = 1$ and $\hat{y} \approx 0$, $y \ln \hat{y} \rightarrow -\infty$

$(1 - y) = 0$ as $y = 1$ and $(1 - y) \ln(1 - \hat{y}) = 0 \Rightarrow L \rightarrow \infty$

Examples	\bar{X}	Ground Truth Y either 0 or 1	\hat{y} Prediction	$L^{(i)} = h_W(x)$ $= g(W^T X)$
x_1	$x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(N)}$	y_1	\hat{y}_1	L_1
x_2	$x_2^{(1)}, x_2^{(2)}, \dots, x_2^{(N)}$	y_2	\hat{y}_2	L_1
\vdots	\vdots	\vdots	\vdots	\vdots
x_n	$x_n^{(1)}, x_n^{(2)}, \dots, x_n^{(N)}$	y_n	\hat{y}_n	L_n

Gradient computation of cross entropy loss function – Chain Rule

$L = \sum_{i=1}^n L_i$, summation of all the binary entropy losses from each individual data point.

$$\frac{\partial L}{\partial w_j} = \sum_{i=1}^n \frac{\partial L_i}{\partial w_j}$$

$$\frac{\partial L_i}{\partial w_j} = \frac{\partial L^{(i)}}{\partial \hat{y}^{(i)}} \frac{\partial \hat{y}^{(i)}}{\partial z} \frac{\partial z}{\partial w_j}$$

Let for simplicity we are not writing i

$$L = -\{y \ln \hat{y} + (1 - y) \ln(1 - \hat{y})\}$$

y is fixed and \hat{y} is the hypothesis function which is estimated

$$\frac{\partial L}{\partial \hat{y}} = -\left\{ \frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})} \right\} \text{-----} -A$$

$$\frac{\partial \hat{y}}{\partial z} = \frac{\partial g(z)}{\partial z} = g(z)(1 - g(z)) = \hat{y}(1 - \hat{y}) \text{-----} -B$$

$$\frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} = -\left\{ \frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})} \right\} \hat{y}(1 - \hat{y}) = -\{y - \hat{y}\}$$

By chain rule,

$$\frac{\partial L}{\partial z} = -\{y - \hat{y}\} = -error$$

Next, $\frac{\partial z}{\partial w_j} = x_j$ where $z = w^T x$

So, $\frac{\partial L_i}{\partial w_j} = -\{y_i - \hat{y}_i\} x_i^{(j)}$

$$\Rightarrow \frac{\partial L}{\partial w_j} = \sum_{i=1}^n -\{y_i - \hat{y}_i\} x_i^{(j)}$$

Decision Boundary is that line where both cases are equiprobable

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

From the figure,

$$\left. \begin{array}{l} g(z) \geq 0.5 \text{ when } z \geq 0 \\ g(z) < 0.5 \text{ when } z < 0 \end{array} \right\} \text{--- -A}$$

Using the sigmoid function, the conditional probability of y being 1 is computed as: $h_W(X) = g(W^T X) = \frac{1}{1+e^{-W^T X}} = P(y = 1/x; W)$.

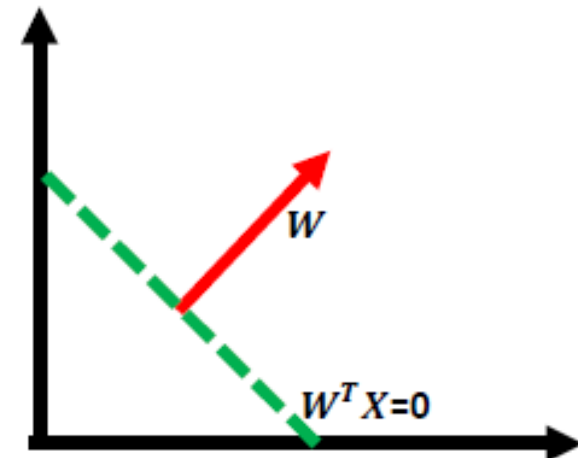
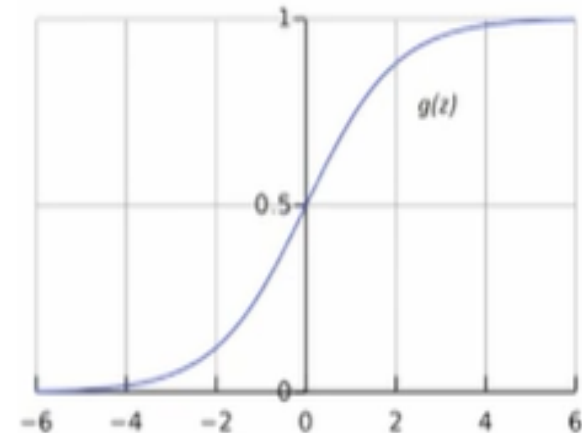
$\left\{ \begin{array}{l} \text{Predict "y = 1" if } h_W(X) \geq 0.5 \\ \text{Predict "y = 0" if } h_W(X) < 0.5 \end{array} \right.$

Observation A suggests that, $h_W(X) = g(W^T X) \geq 0.5$ when $W^T X \geq 0$

$$h_W(X) = g(W^T X) < 0.5 \text{ when } W^T X < 0$$

Thus, $\left\{ \begin{array}{l} \text{Predict "y = 1" if } W^T X \geq 0 \\ \text{Predict "y = 0" if } W^T X < 0 \end{array} \right\} \text{--- -B}$

At decision boundary, $W^T X = 0$ implies $P(y = 1/x; W) = P(y = 0/x; W) = 0.5$.



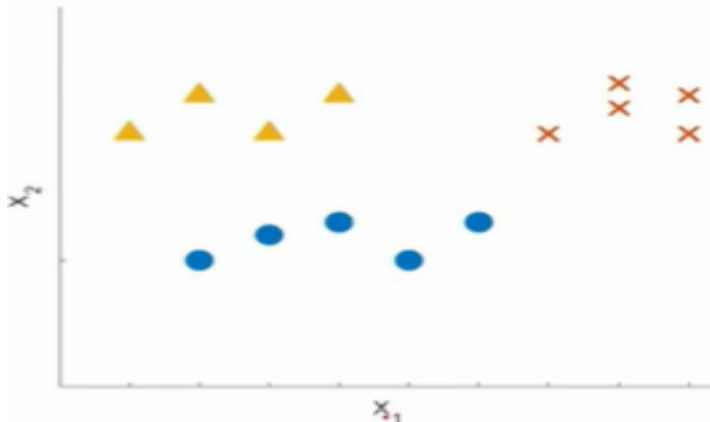
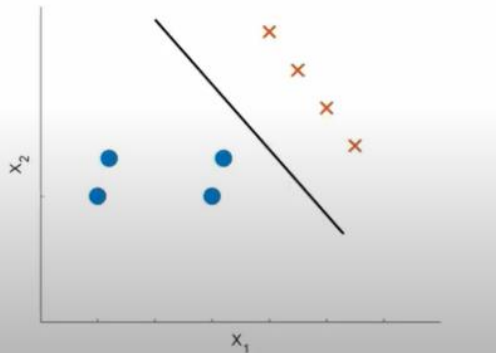
Logistic regression for multi-class classification task

Example:

Medical Diagnosis	Stage I	Stage II	Stage III
	$y = 1$	$y = 2$	$y = 3$

Weather	Sunny	Cloudy	Rain
	$y = 1$	$y = 2$	$y = 3$

Binary:-






N classes, N binary classifier models

We have used 3 different symbols to represent 3 classes.

How to get learning algorithm to work for it?

Training set:-

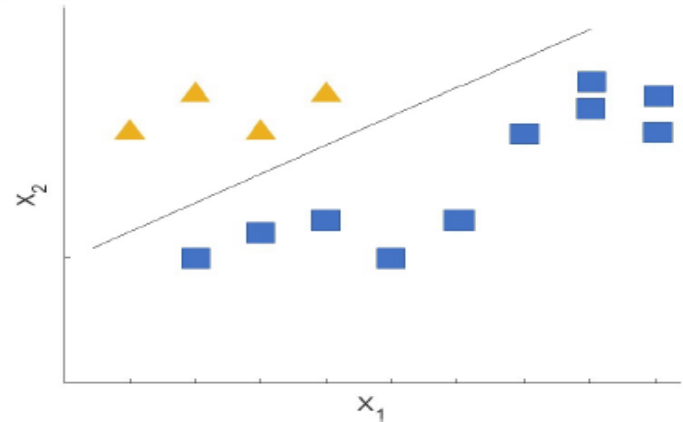
Class 1	
Class 2	
Class 3	

This training set can be divided into 3 separate binary classification problems.

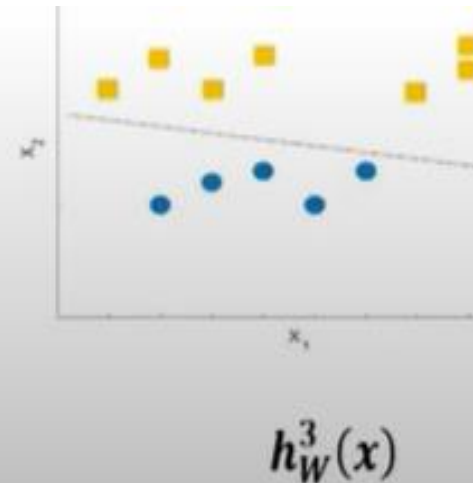
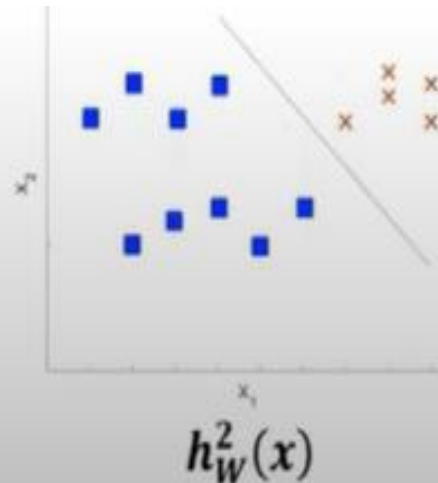
Logistic regression for multi-class classification task

Train a Logistic regression classifier $h_W^i(X)$ for each class i to predict the probability at $y = i$.

Create a new training set by considering class 2 and class 3 as negative class and class 1 as positive class. Then, apply logistic regression to find the decision boundary and design the classifier $h_W^1(x) = P(y = 1|x; W)$.



Similarly, for class 2, we will design $h_W^2(X)$ and for class 3, we will get $h_W^3(X)$.



we fit 3 classifiers $h_W^i(x) = P(y = i|x; W)$, for each class i .

To deal with multiclass is one-versus-all classification

- Build a separate model for each class.
- N number of data points and K classes.
- Build K different logistic regression model with the data.
- In each model, we take one class, and all other classes as one class to build the same two class model.
- Now, when we get any record to classify in one of the K classes, we predict each class probability by using K different model.
- The class having maximum probability is reported as predicted class.
- One important observation is the sum of each class probability is not needed to be 1.
- For example, By model-1 $\text{Prob}(\text{dog} | x) = 0.4$, By model-2 $\text{Prob}(\text{cat} | x) = 0.7$, By model-3 $\text{Prob}(\text{mouse} | x) = 0.2$
- Prediction = Class ($\text{Max}(\text{Prob}(\text{dog} | x), \text{Prob}(\text{cat} | x), \text{Prob}(\text{mouse} | x))$) = cat

Logistic regression

- For logistic regression, the loss function is not convex.
- There is a possibility of many classifiers.
- To avoid over fitting and under fitting, we should include all significant variables.
- It requires large sample sizes
- The independent variables should not be correlated with each other i.e. no multi collinearity.
- If the values of dependent variable is ordinal, then it is called as **Ordinal logistic regression**
- If dependent variable is multi class then it is known as **Multinomial Logistic regression**.

Thank you

- ???